



debian

Ghidul de referință Debian

Osamu Aoki

Copyright © 2013-2024 Osamu Aoki

Acest Ghid de referință Debian (versiunea 2.136) () are scopul de a oferi o imagine de ansamblu asupra sistemului Debian, ca ghid de utilizare post-instalare. Acesta acoperă multe aspecte ale administrării sistemului prin exemple de comenzi shell pentru utilizatorii care nu sunt dezvoltatori.

Cuprins

1	Tutoriale GNU/Linux	1
1.1	Noțiuni de bază despre consolă	1
1.1.1	Promptul shell	1
1.1.2	Promptul shell în GUI	2
1.1.3	Contul root	2
1.1.4	Promptul shell-ului root	3
1.1.5	Instrumente grafice de administrare a sistemului	3
1.1.6	Console virtuale	3
1.1.7	Cum să ieșiți din promptul liniei de comandă	4
1.1.8	Cum să opriți sistemul	4
1.1.9	Recuperarea unei console funcționale	4
1.1.10	Sugestii de pachete suplimentare pentru începători	4
1.1.11	Un cont de utilizator suplimentar	5
1.1.12	Configurarea «sudo»	5
1.1.13	Ora de joacă	6
1.2	Sistem de fișiere de tip Unix	6
1.2.1	Noțiuni de bază despre fișierele Unix	7
1.2.2	Elementele interne ale sistemului de fișiere	8
1.2.3	Permițiuni ale sistemului de fișiere	8
1.2.4	Controlul permișiunilor pentru fișierele nou create: umask	11
1.2.5	Permițiuni pentru grupuri de utilizatori (grup)	11
1.2.6	Marcaje de timp	13
1.2.7	Legături	14
1.2.8	Conducte cu nume (FIFO)	15
1.2.9	Socluri	15
1.2.10	Fișiere de dispozitive	16
1.2.11	Fișiere de dispozitive speciale	16
1.2.12	procfs și sysfs	17
1.2.13	tmpfs	17
1.3	Midnight Commander (MC)	18

1.3.1	Personalizarea MC	18
1.3.2	Lansarea MC	18
1.3.3	Gestionarul de fișiere din MC	18
1.3.4	Trucuri din linia de comandă în MC	19
1.3.5	Editorul intern din MC	19
1.3.6	Vizorul intern din MC	20
1.3.7	Funcții de pornire automată ale MC	20
1.3.8	Sistemul de fișiere virtual al MC	20
1.4	Mediul de lucru de bază de tip Unix	20
1.4.1	Shell-ul de autentificare	20
1.4.2	Personalizarea bash	21
1.4.3	Combinatii speciale de taste	22
1.4.4	Operații cu mouse-ul	22
1.4.5	Paginatorul	23
1.4.6	Editorul de text	23
1.4.7	Definirea unui editor de text implicit	24
1.4.8	Utilizarea editorului vim	24
1.4.9	Înregistrarea activităților shell-ului	25
1.4.10	Comenzi Unix de bază	25
1.5	Comanda simplă de shell	26
1.5.1	Executarea comenzilor și variabilele de mediu	28
1.5.2	Variabila „\$LANG”	28
1.5.3	Variabila „\$PATH”	29
1.5.4	Variabila „\$HOME”	29
1.5.5	Opțiuni în linia de comandă	30
1.5.6	Facilitatea glob a shell-ului	30
1.5.7	Valoarea returnată de comandă	31
1.5.8	Secvențe tipice de comenzi și redirecționarea shell-ului	31
1.5.9	Alias de comandă	33
1.6	Procesarea textului în stilul Unix	33
1.6.1	Instrumente pentru text în Unix	34
1.6.2	Expresii regulate	36
1.6.3	Expresii de înlocuire	36
1.6.4	Înlocuire globală cu expresii regulate	37
1.6.5	Extragerea datelor din tabelul fișierului text	38
1.6.6	Fragmente de script pentru comenzi de direcționare	39

2	Gestionarea pachetelor Debian	41
2.1	Cerințe preliminare pentru gestionarea pachetelor Debian	41
2.1.1	Sistemul de gestionare a pachetelor Debian	41
2.1.2	Configurația pachetului	42
2.1.3	Precauții de bază	42
2.1.4	O viață cu îmbunătățiri veșnice	43
2.1.5	Noțiuni de bază despre arhiva Debian	44
2.1.6	Debian este software 100% liber	48
2.1.7	Dependențele pachetelor	49
2.1.8	Fluxul evenimentelor din gestionarea pachetelor	50
2.1.9	Primul răspuns la problemele legate de gestionarea pachetelor	51
2.1.10	Cum să selectați pachetele Debian	52
2.1.11	Cum să faceți față cerințelor contradictorii	52
2.2	Operații de bază de gestionare a pachetelor	53
2.2.1	apt vs. apt-get / apt-cache vs. aptitude	53
2.2.2	Operații de bază de gestionare a pachetelor din linia de comandă	54
2.2.3	Utilizarea interactivă a «aptitude»	54
2.2.4	Combinatii de taste pentru aptitude	56
2.2.5	Vizualizarea de pachete în aptitude	57
2.2.6	Opțiuni pentru metoda de căutare cu aptitude	58
2.2.7	Formula expresiei regulate pentru aptitudine	58
2.2.8	Rezolvarea dependențelor de către aptitude	60
2.2.9	Jurnale de activitate ale pachetelor	60
2.3	Exemple de operații cu aptitudine	60
2.3.1	Căutarea de pachete interesante	60
2.3.2	Listarea pachetelor care se potrivesc cu expresia regulată în numele pachetelor	61
2.3.3	Navigarea prin rezultatul potrivirilor expresiei regulate	61
2.3.4	Stergerea definitivă a pachetelor eliminate	61
2.3.5	Reorganizarea stării instalării automate/manuale	61
2.3.6	Actualizare la nivel de sistem	62
2.4	Operații avansate de gestionare a pachetelor	63
2.4.1	Operații avansate de gestionare a pachetelor din linia de comandă	63
2.4.2	Verificarea fișierelor pachetului instalat	65
2.4.3	Protejarea împotriva problemelor legate de pachete	65
2.4.4	Căutarea în metadatele pachetului	65
2.5	Detalii interne privind gestionarea pachetelor Debian	65
2.5.1	Metadate de arhivă	66
2.5.2	Fișierul „Release” de nivel superior și autenticitatea	66
2.5.3	Fișiere „Release” la nivel de arhivă	67

2.5.4	Preluarea metadatelor pentru pachet	68
2.5.5	Starea pachetului pentru APT	68
2.5.6	Starea pachetului pentru aptitude	68
2.5.7	Copiile locale ale pachetelor descărcate	69
2.5.8	Numele fișierelor pachetelor Debian	69
2.5.9	Comanda «dpkg»	70
2.5.10	Comanda «update-alternatives»	70
2.5.11	Comanda «dpkg-statoverride»	71
2.5.12	Comanda «dpkg-divert»	71
2.6	Recuperarea dintr-un sistem defect	71
2.6.1	Instalare eșuată din cauza dependențelor lipsă	72
2.6.2	Erori de stocare în cache ale datelor pachetului	72
2.6.3	Incompatibilitate cu configurația veche a utilizatorului	72
2.6.4	Pachete diferite cu fișiere suprapuse	72
2.6.5	Remediarea scriptului pachetului defect	73
2.6.6	Recuperarea cu comanda dpkg	73
2.6.7	Recuperarea datelor privind selecția pachetelor	74
2.7	Sfaturi pentru gestionarea pachetelor	74
2.7.1	Cine a încărcat pachetul?	74
2.7.2	Limitarea lățimii de bandă pentru descărcare pentru APT	75
2.7.3	Descărcarea și actualizarea automată a pachetelor	75
2.7.4	Actualizări și retro-adaptări (versiuni de software migrate din ramura principală de dezvoltare și adaptate pentru a funcționa cu această versiune)	75
2.7.5	Arhive de pachete externe	76
2.7.6	Pachete din surse mixte de arhive fără apt-pinning	76
2.7.7	Ajustarea versiunii candidate cu apt-pinning	77
2.7.8	Blocarea pachetelor instalate de „Recommends”	79
2.7.9	Urmărirea suitei <code>testing</code> cu unele pachete din <code>unstable</code>	79
2.7.10	Urmărirea suitei <code>unstable</code> cu unele pachete din <code>experimental</code>	80
2.7.11	Retrogradarea de urgență	81
2.7.12	Pachetul <code>equivs</code>	82
2.7.13	Adaptarea unui pachet la sistemul stabil	82
2.7.14	Server proxy pentru APT	83
2.7.15	Mai multe informații despre gestionarea pachetelor	83

3	Inițializarea sistemului	84
3.1	O prezentare generală a procesului de inițializare	84
3.1.1	Etapa 1: UEFI	85
3.1.2	Etapa 2: încărcătorul de pornire	85
3.1.3	Etapa 3: sistemul mini-Debian	87
3.1.4	Etapa 4: sistemul Debian normal	87
3.2	Systemd	88
3.2.1	Init systemd	88
3.2.2	Autentificarea cu systemd	89
3.3	Mesajele nucleului	90
3.4	Mesajele sistemului	90
3.5	Gestionarea sistemului	90
3.6	Alte monitoare de sistem	92
3.7	Configurația sistemului	92
3.7.1	Numele gazdei	92
3.7.2	Sistemul de fișiere	93
3.7.3	Inițializarea interfeței de rețea	93
3.7.4	Inițializarea sistemului cloud	93
3.7.5	Exemplu de personalizare pentru ajustarea serviciului sshd	93
3.8	Sistemul udev	94
3.9	Inițializarea modulelor de nucleu	94
4	Autentificare și controale de acces	96
4.1	Autentificare normală Unix	96
4.2	Gestionarea informațiilor privind contul și parola	98
4.3	Parolă bună	98
4.4	Crearea unei parole criptate	99
4.5	PAM și NSS	99
4.5.1	Configuration files accessed by PAM and NSS	100
4.5.2	The modern centralized system management	101
4.5.3	"Why GNU su does not support the wheel group"	101
4.5.4	Stricter password rule	101
4.6	Security of authentication	102
4.6.1	Secure password on the Internet	102
4.6.2	Shell securizat	102
4.6.3	Extra security measures for the Internet	103
4.6.4	Securing the root password	103
4.7	Other access controls	104
4.7.1	Liste de control al accesului (ACL)	104
4.7.2	sudo	105
4.7.3	PolicyKit	105
4.7.4	Restricting access to some server services	105
4.7.5	Linux security features	106

5	Configurarea rețelei	107
5.1	The basic network infrastructure	107
5.1.1	Rezoluția numelui de gazdă	107
5.1.2	Numele interfeței de rețea	109
5.1.3	Intervalul de adrese pentru rețeaua locală (LAN)	110
5.1.4	The network device support	110
5.2	The modern network configuration for desktop	110
5.2.1	GUI network configuration tools	111
5.3	The modern network configuration without GUI	111
5.4	The modern network configuration for cloud	112
5.4.1	The modern network configuration for cloud with DHCP	112
5.4.2	The modern network configuration for cloud with static IP	112
5.4.3	The modern network configuration for cloud with Network Manager	112
5.5	The low level network configuration	113
5.5.1	Comenzi «iproute2»	113
5.5.2	Safe low level network operations	113
5.6	Optimizarea rețelei	114
5.6.1	Finding optimal MTU	114
5.6.2	WAN TCP optimization	115
5.7	Infrastructura netfilter	115
6	Network applications	117
6.1	Navigatoare Web	117
6.1.1	Spoofing the User-Agent string	118
6.1.2	Browser extension	118
6.2	Sistemul de poștă electronică	118
6.2.1	Email basics	118
6.2.2	Modern mail service limitation	119
6.2.3	Historic mail service expectation	120
6.2.4	Agentul de transport al poștei electronice („Mail transport agent”: MTA)	120
6.2.4.1	The configuration of exim4	120
6.2.4.2	The configuration of postfix with SASL	122
6.2.4.3	The mail address configuration	123
6.2.4.4	Basic MTA operations	124
6.3	The remote access server and utilities (SSH)	124
6.3.1	Basics of SSH	125
6.3.2	User name on the remote host	125
6.3.3	Connecting without remote passwords	126
6.3.4	Dealing with alien SSH clients	126

6.3.5	Setting up ssh-agent	126
6.3.6	Sending a mail from a remote host	127
6.3.7	Port forwarding for SMTP/POP3 tunneling	127
6.3.8	How to shutdown the remote system on SSH	127
6.3.9	Soluționarea problemelor SSH	127
6.4	The print server and utilities	128
6.5	Other network application servers	128
6.6	Other network application clients	129
6.7	The diagnosis of the system daemons	129
7	GUI System	131
7.1	GUI desktop environment	131
7.2	GUI communication protocol	132
7.3	GUI infrastructure	133
7.4	Aplicații cu interfață grafică	133
7.5	User directories	134
7.6	Tipuri de litere	134
7.6.1	Basic fonts	134
7.6.2	Font rasterization	137
7.7	Cutia cu nisip (sandbox)	137
7.8	Mediu de birou la distanță	138
7.9	X server connection	138
7.9.1	X server local connection	138
7.9.2	X server remote connection	140
7.9.3	X server chroot connection	140
7.10	Clipboard	140
8	I18N și L10N	142
8.1	The locale	142
8.1.1	Rationale for UTF-8 locale	142
8.1.2	The reconfiguration of the locale	143
8.1.3	Codificarea numelor de fișiere	144
8.1.4	Localized messages and translated documentation	144
8.1.5	Effects of the locale	144
8.2	Introducere de la tastatură	145
8.2.1	The keyboard input for Linux console and X Window	145
8.2.2	The keyboard input for Wayland	145
8.2.3	The input method support with IBus	145
8.2.4	Un exemplu pentru japoneză	146
8.3	The display output	147
8.4	East Asian Ambiguous Character Width Characters	147

9	System tips	148
9.1	The console tips	148
9.1.1	Recording the shell activities cleanly	148
9.1.2	Programul «screen»	149
9.1.3	Navigating around directories	150
9.1.4	Readline wrapper	150
9.1.5	Scanning the source code tree	150
9.2	Personalizarea vim	151
9.2.1	Personalizarea vim cu caracteristicile interne	151
9.2.2	Personalizarea vim cu pachete externe	153
9.3	Data recording and presentation	154
9.3.1	Demonul de jurnalizare	154
9.3.2	Log analyzer	154
9.3.3	Customized display of text data	155
9.3.4	Customized display of time and date	155
9.3.5	Colorized shell echo	156
9.3.6	Comenzi colorate	156
9.3.7	Recording the editor activities for complex repeats	156
9.3.8	Recording the graphics image of an X application	157
9.3.9	Recording changes in configuration files	157
9.4	Monitoring, controlling, and starting program activities	157
9.4.1	Timing a process	158
9.4.2	Prioritatea de planificare	158
9.4.3	Comanda «ps»	159
9.4.4	Comanda «top»	159
9.4.5	Listing files opened by a process	159
9.4.6	Tracing program activities	159
9.4.7	Identification of processes using files or sockets	159
9.4.8	Repeating a command with a constant interval	160
9.4.9	Repeating a command looping over files	160
9.4.10	Starting a program from GUI	160
9.4.11	Customizing program to be started	162
9.4.12	Omorârea unui proces	162
9.4.13	Scheduling tasks once	162
9.4.14	Scheduling tasks regularly	164
9.4.15	Scheduling tasks on event	164
9.4.16	Alt-SysRq key	164
9.5	System maintenance tips	165
9.5.1	Who is on the system?	165

9.5.2	Warning everyone	165
9.5.3	Hardware identification	166
9.5.4	Hardware configuration	166
9.5.5	System and hardware time	166
9.5.6	The terminal configuration	167
9.5.7	Infrastructura de sunet	167
9.5.8	Disabling the screen saver	168
9.5.9	Disabling beep sounds	168
9.5.10	Utilizare memorie	169
9.5.11	System security and integrity check	169
9.6	Data storage tips	170
9.6.1	Disk space usage	170
9.6.2	Disk partition configuration	170
9.6.3	Accessing partition using UUID	171
9.6.4	LVM2	171
9.6.5	Filesystem configuration	172
9.6.6	Filesystem creation and integrity check	172
9.6.7	Optimization of filesystem by mount options	173
9.6.8	Optimization of filesystem via superblock	173
9.6.9	Optimizarea discului dur	173
9.6.10	Optimization of solid state drive	174
9.6.11	Using SMART to predict hard disk failure	174
9.6.12	Specify temporary storage directory via \$TMPDIR	174
9.6.13	Expansion of usable storage space via LVM	174
9.6.14	Expansion of usable storage space by mounting another partition	175
9.6.15	Expansion of usable storage space by bind-mounting another directory	175
9.6.16	Expansion of usable storage space by overlay-mounting another directory	175
9.6.17	Expansion of usable storage space using symlink	175
9.7	The disk image	176
9.7.1	Making the disk image file	176
9.7.2	Writing directly to the disk	176
9.7.3	Mounting the disk image file	177
9.7.4	Cleaning a disk image file	178
9.7.5	Making the empty disk image file	178
9.7.6	Making the ISO9660 image file	179
9.7.7	Writing directly to the CD/DVD-R/RW	180
9.7.8	Mounting the ISO9660 image file	180
9.8	The binary data	180
9.8.1	Viewing and editing binary data	180

9.8.2	Manipulating files without mounting disk	181
9.8.3	Data redundancy	181
9.8.4	Data file recovery and forensic analysis	181
9.8.5	Splitting a large file into small files	182
9.8.6	Clearing file contents	182
9.8.7	Dummy files	182
9.8.8	Erasing an entire hard disk	183
9.8.9	Erasing unused area of an hard disk	183
9.8.10	Undeleting deleted but still open files	184
9.8.11	Searching all hardlinks	184
9.8.12	Invisible disk space consumption	184
9.9	Data encryption tips	185
9.9.1	Removable disk encryption with dm-crypt/LUKS	185
9.9.2	Mounting encrypted disk with dm-crypt/LUKS	186
9.10	Nucleul	186
9.10.1	Parametrii nucleului	186
9.10.2	Kernel headers	186
9.10.3	Compiling the kernel and related modules	187
9.10.4	Compiling the kernel source: Debian Kernel Team recommendation	187
9.10.5	Hardware drivers and firmware	188
9.11	Virtualized system	189
9.11.1	Virtualization and emulation tools	189
9.11.2	Virtualization work flow	189
9.11.3	Mounting the virtual disk image file	191
9.11.4	Chroot system	191
9.11.5	Multiple desktop systems	192
10	Data management	194
10.1	Sharing, copying, and archiving	194
10.1.1	Archive and compression tools	195
10.1.2	Copy and synchronization tools	195
10.1.3	Idioms for the archive	195
10.1.4	Idioms for the copy	197
10.1.5	Idioms for the selection of files	198
10.1.6	Archive media	199
10.1.7	Removable storage device	200
10.1.8	Filesystem choice for sharing data	201
10.1.9	Sharing data via network	202
10.2	Backup and recovery	202

10.2.1 Backup and recovery policy	203
10.2.2 Backup utility suites	204
10.2.3 Backup tips	205
10.2.3.1 GUI backup	205
10.2.3.2 Mount event triggered backup	206
10.2.3.3 Timer event triggered backup	206
10.3 Data security infrastructure	207
10.3.1 Key management for GnuPG	207
10.3.2 Using GnuPG on files	208
10.3.3 Using GnuPG with Mutt	208
10.3.4 Using GnuPG with Vim	210
10.3.5 Suma MD5	210
10.3.6 Password keyring	210
10.4 Source code merge tools	210
10.4.1 Extracting differences for source files	212
10.4.2 Merging updates for source files	212
10.4.3 Interactive merge	212
10.5 Git	212
10.5.1 Configuration of Git client	212
10.5.2 Basic Git commands	213
10.5.3 Git tips	214
10.5.4 Git references	214
10.5.5 Other version control systems	216
11 Data conversion	217
11.1 Text data conversion tools	217
11.1.1 Converting a text file with iconv	217
11.1.2 Checking file to be UTF-8 with iconv	219
11.1.3 Converting file names with iconv	219
11.1.4 EOL conversion	220
11.1.5 TAB conversion	220
11.1.6 Editors with auto-conversion	221
11.1.7 Plain text extraction	221
11.1.8 Highlighting and formatting plain text data	221
11.2 XML data	221
11.2.1 Basic hints for XML	223
11.2.2 XML processing	224
11.2.3 The XML data extraction	224
11.2.4 The XML data lint	224

11.3 Type setting	225
11.3.1 roff typesetting	225
11.3.2 TeX/LaTeX	226
11.3.3 Pretty print a manual page	226
11.3.4 Creating a manual page	226
11.4 Printable data	227
11.4.1 Ghostscript	227
11.4.2 Merge two PS or PDF files	227
11.4.3 Printable data utilities	228
11.4.4 Imprimarea cu CUPS	228
11.5 The mail data conversion	229
11.5.1 Mail data basics	229
11.6 Graphic data tools	230
11.6.1 Graphic data tools (metapackage)	230
11.6.2 Graphic data tools (GUI)	230
11.6.3 Graphic data tools (CLI)	230
11.7 Miscellaneous data conversion	233
12 Programare	234
12.1 The shell script	234
12.1.1 POSIX shell compatibility	235
12.1.2 Shell parameters	235
12.1.3 Conditionale shell	237
12.1.4 Bucle shell	237
12.1.5 Shell environment variables	238
12.1.6 The shell command-line processing sequence	238
12.1.7 Utility programs for shell script	239
12.2 Scripting in interpreted languages	240
12.2.1 Debugging interpreted language codes	240
12.2.2 GUI program with the shell script	241
12.2.3 Custom actions for GUI filer	241
12.2.4 Perl short script madness	242
12.3 Coding in compiled languages	242
12.3.1 C	242
12.3.2 Simple C program (gcc)	243
12.3.3 Flex — a better Lex	244
12.3.4 Bison — a better Yacc	244
12.4 Static code analysis tools	245
12.5 Depanare	246

12.5.1 Basic gdb execution	246
12.5.2 Debugging the Debian package	247
12.5.3 Obtaining backtrace	248
12.5.4 Advanced gdb commands	248
12.5.5 Check dependency on libraries	248
12.5.6 Dynamic call tracing tools	249
12.5.7 Debugging X Errors	249
12.5.8 Memory leak detection tools	249
12.5.9 Disassemble binary	249
12.6 Instrumentele de construcție	250
12.6.1 Make	250
12.6.2 Autotools	251
12.6.2.1 Compile and install a program	251
12.6.2.2 Uninstall program	251
12.6.3 Meson	251
12.7 Web	252
12.8 The source code translation	252
12.9 Making Debian package	253
A Appendix	254
A.1 The Debian maze	254
A.2 Copyright history	254
A.3 Formatul documentului	255

Listă de tabele

1.1	Lista pachetelor de programe interesante în modul text	5
1.2	Lista pachetelor de documentație utilă	5
1.3	Lista directoarelor cheie și descrierea utilizării acestora	8
1.4	Lista descriptivă a primului caracter din ieșirea comenzii «ls -l»	9
1.5	Modul numeric pentru permisiunile de fișiere în comenzile chmod(1)	10
1.6	Exemple de valori pentru umask	11
1.7	Lista grupurilor notabile furnizate de sistem pentru accesul la fișiere	12
1.8	Lista grupurilor notabile furnizate de sistem pentru executarea anumitor comenzi	13
1.9	Lista tipurilor de marcaje de timp	13
1.10	Lista fișierelor speciale de dispozitive	17
1.11	Tastele de comenzi rapide ale MC	19
1.12	Reacția la apăsarea tastei «Enter» în MC	20
1.13	Lista programelor shell	21
1.14	Lista tastelor de comenzi rapide pentru bash	22
1.15	Lista operațiilor mouse-ului și acțiunile tastelor asociate în Debian	23
1.16	Lista combinațiilor de taste de bază din Vim	25
1.17	Lista comenzilor Unix de bază	27
1.18	Cele 3 părți ale valorii configurației regionale	28
1.19	Lista recomandărilor privind configurația regională	29
1.20	Lista valorilor variabilei „\$HOME”	29
1.21	Modele globale shell	30
1.22	Coduri de ieșire ale comenzii	31
1.23	Expresii idiomatice ale comenzii shell	32
1.24	Descriptori de fișiere predefiniți	33
1.25	Metacaractere pentru BRE și ERE	35
1.26	Expresia de înlocuire	36
1.27	Lista fragmentelor de script pentru comenzi de direcționare	39
2.1	Lista instrumentelor de gestionare a pachetelor Debian	42
2.2	Lista siturilor de arhivă Debian	46

2.3	Lista secțiunilor de arhivă Debian	46
2.4	Relația dintre versiune și numele în cod	47
2.5	Lista siturilor web importante pentru rezolvarea problemelor legate de un pachet specific	52
2.6	Operații de bază de gestionare a pachetelor din linia de comandă folosind <code>apt(8)</code> , <code>aptitude(8)</code> și <code>apt-get(8)</code> / <code>apt-cache(8)</code>	55
2.7	Opțiuni de comandă demne de menționat pentru <code>aptitude(8)</code>	55
2.8	Lista combinațiilor de taste pentru « <code>aptitude</code> »	56
2.9	Lista vizualizărilor pentru <code>aptitude</code>	57
2.10	Clasificarea vizualizărilor standard ale pachetelor	58
2.11	Lista formulelor de expresii regulate pentru <code>aptitude</code>	59
2.12	Fișierele jurnal pentru activitățile pachetului	60
2.13	Lista operațiilor avansate de gestionare a pachetelor	64
2.14	Conținutul metadatelor arhivei Debian	66
2.15	Structura numelor pachetelor Debian	69
2.16	Caracterele utilizabile pentru fiecare componentă din numele pachetelor Debian	69
2.17	Fișierele importante create de <code>dpkg</code>	70
2.18	Lista valorilor notabile (celor mai importante) ale priorității Pin pentru tehnica apt-pinning .	78
2.19	Lista instrumentelor proxy special pentru arhiva Debian	83
3.1	Lista încărcătorilor de pornire	85
3.2	Semnificația intrării din meniul din partea de sus a <code>/boot/grub/grub.cfg</code>	86
3.3	Lista instrumentelor de pornire pentru sistemul Debian	88
3.4	Lista nivelurilor de eroare ale nucleului	90
3.5	Lista fragmentelor tipice de comandă <code>journalctl</code>	90
3.6	Lista fragmentelor tipice de comenzi <code>systemctl</code>	91
3.7	Lista altor fragmente de comenzi de monitorizare sub <code>systemd</code>	92
4.1	3 fișiere de configurare importante pentru <code>pam_unix(8)</code>	96
4.2	Al doilea conținut al intrării „ <code>/etc/passwd</code> ”	97
4.3	Lista comenzilor pentru gestionarea informațiilor contului	98
4.4	Lista instrumentelor pentru generarea parolei	99
4.5	List of notable PAM and NSS systems	100
4.6	Lista fișierelor de configurare accesate de PAM și NSS	100
4.7	List of insecure and secure services and ports	102
4.8	List of tools to provide extra security measures	103
5.1	List of network configuration tools	108
5.2	List of network address ranges	110
5.3	Translation table from obsolete <code>net-tools</code> commands to new <code>iproute2</code> commands	113
5.4	List of low level network commands	113

5.5	Instrumente de optimizare a rețelei	114
5.6	Basic guide lines of the optimal MTU value	115
5.7	Lista instrumentelor de paravan de protecție	116
6.1	List of web browsers	117
6.2	List of mail user agent (MUA)	119
6.3	List of basic mail transport agent related packages	121
6.4	List of important postfix manual pages	122
6.5	List of mail address related configuration files	123
6.6	List of basic MTA operation	124
6.7	List of remote access server and utilities	125
6.8	List of SSH configuration files	125
6.9	List of SSH client startup examples	126
6.10	List of free SSH clients for other platforms	126
6.11	List of print servers and utilities	128
6.12	List of other network application servers	129
6.13	List of network application clients	130
6.14	List of popular RFCs	130
7.1	List of desktop environment	131
7.2	List of notable GUI infrastructure packages	133
7.3	List of notable GUI applications	135
7.4	List of notable TrueType and OpenType fonts	136
7.5	List of notable font environment and related packages	137
7.6	List of notable sandbox environment and related packages	138
7.7	List of notable remote access server	139
7.8	List of connection methods to the X server	139
7.9	List of programs related to manipulating character clipboard	141
8.1	List of IBus and its engine packages	146
9.1	List of programs to support console activities	148
9.2	List of key bindings for screen	150
9.3	Informații despre inițializarea vim	154
9.4	List of system log analyzers	155
9.5	Display examples of time and date for the "ls -l" command with the time style value	155
9.6	List of graphics image manipulation tools	157
9.7	List of packages which can record configuration history	157
9.8	List of tools for monitoring and controlling program activities	158
9.9	List of nice values for the scheduling priority	158

9.10 List of ps command styles	159
9.11 List of frequently used signals for kill command	163
9.12 List of notable SAK command keys	165
9.13 List of hardware identification tools	166
9.14 List of hardware configuration tools	166
9.15 List of sound packages	168
9.16 List of commands for disabling the screen saver	168
9.17 List of memory sizes reported	169
9.18 List of tools for system security and integrity check	170
9.19 List of disk partition management packages	171
9.20 List of filesystem management packages	172
9.21 List of packages which view and edit binary data	181
9.22 List of packages to manipulate files without mounting disk	181
9.23 List of tools to add data redundancy to files	181
9.24 List of packages for data file recovery and forensic analysis	182
9.25 List of data encryption utilities	185
9.26 List of key packages to be installed for the kernel recompilation on the Debian system	187
9.27 Lista instrumentelor de virtualizare	190
10.1 List of archive and compression tools	196
10.2 List of copy and synchronization tools	197
10.3 List of filesystem choices for removable storage devices with typical usage scenarios	201
10.4 List of the network service to chose with the typical usage scenario	202
10.5 List of backup suite utilities	204
10.6 List of data security infrastructure tools	207
10.7 List of GNU Privacy Guard commands for the key management	208
10.8 List of the meaning of the trust code	208
10.9 List of GNU Privacy Guard commands on files	209
10.10 List of source code merge tools	211
10.11 List of git related packages and commands	213
10.12 Main Git commands	214
10.13 Git tips	215
10.14 List of other version control system tools	216
11.1 List of text data conversion tools	217
11.2 List of encoding values and their usage	218
11.3 List of EOL styles for different platforms	220
11.4 List of TAB conversion commands from <code>bsdmainutils</code> and <code>coreutils</code> packages	220
11.5 List of tools to extract plain text data	222

11.6 List of tools to highlight plain text data	222
11.7 List of predefined entities for XML	223
11.8 List of XML tools	224
11.9 List of DSSSL tools	224
11.10List of XML data extraction tools	225
11.11List of XML pretty print tools	225
11.12List of type setting tools	225
11.13List of packages to help creating the manpage	227
11.14List of Ghostscript PostScript interpreters	227
11.15List of printable data utilities	228
11.16List of packages to help mail data conversion	229
11.17List of graphics data tools (metapackage)	230
11.18List of graphics data tools (GUI)	231
11.19List of graphics data tools (CLI)	232
11.20List of miscellaneous data conversion tools	232
12.1 List of typical bashisms	235
12.2 List of shell parameters	236
12.3 List of shell parameter expansions	236
12.4 List of key shell parameter substitutions	236
12.5 List of file comparison operators in the conditional expression	237
12.6 List of string comparison operators in the conditional expression	238
12.7 List of packages containing small utility programs for shell scripts	239
12.8 List of interpreter related packages	240
12.9 List of dialog programs	241
12.10List of compiler related packages	243
12.11List of Yacc-compatible LALR parser generators	244
12.12List of tools for static code analysis	246
12.13List of debug packages	246
12.14List of advanced gdb commands	249
12.15List of memory leak detection tools	249
12.16List of build tool packages	250
12.17List of make automatic variables	250
12.18List of make variable expansions	250
12.19List of source code translation tools	252

Rezumat

Această carte este gratuită; o puteți redistribui și/sau modifica în conformitate cu termenii Licenței Publice Generale GNU, în orice versiune compatibilă cu Ghidul Debian pentru Software Liber (DFSG).

Prefață

Acest [Ghid de referință Debian \(versiunea 2.136\)](#) () are scopul de a oferi o imagine de ansamblu asupra administrării sistemului Debian, ca ghid de utilizare post-instalare.

Cititorul vizat este cineva care dorește să învețe scripturi shell, dar care nu este pregătit să citească toate sursele C pentru a înțelege cum funcționează sistemul [GNU/Linux](#).

Pentru instrucțiuni de instalare, consultați:

- [Ghidul de instalare Debian GNU/Linux pentru versiunea stabilă actuală](#)
- [Ghidul de instalare Debian GNU/Linux pentru versiunea de testare actuală](#)

Declinare de responsabilitate

Toate garanțiile sunt declinate. Toate mărcile comerciale sunt proprietatea deținătorilor respectivi ai mărcilor comerciale.

Sistemul Debian în sine este un obiectiv în continuă schimbare. Acest lucru face ca documentația sa să fie dificil de actualizat și corectat. Deși versiunea actuală de testare `testing` a sistemului Debian a fost utilizată ca bază pentru redactarea acestui document, este posibil ca unele informații să fie deja depășite în momentul în care citiți aceste rânduri.

Vă rugăm să tratați acest document ca referință secundară. Acest document nu înlocuiește niciun ghid oficial. Autorul și colaboratorii nu își asumă responsabilitatea pentru consecințele erorilor, omisiunilor sau ambiguităților din acest document.

Ce este Debian

[Proiectul Debian](#) este o asociație de persoane care s-au unit pentru a crea un sistem de operare liber. Distribuția sa se caracterizează prin următoarele.

- Angajamentul față de libertatea software-ului: [Contractul social Debian și Ghidul Debian pentru software-ul liber \(DFSG\)](#)
 - Distribuirea rezultatului muncii voluntare neremunerate în Internet: <https://www.debian.org>
 - Un număr mare de pachete software precompilate de înaltă calitate
 - Accent pe stabilitate și securitate, cu acces facil la actualizările de securitate
 - Accent pe actualizarea fără probleme la cele mai recente pachete software din arhivele `testing`
 - Oferă suport pentru un număr mare de arhitecturi hardware
-

Componentele software-ului liber din Debian provin din [GNU](#), [Linux](#), [BSD](#), [X](#), [ISC](#), [Apache](#), [Ghostscript](#), [Common Unix Printing System](#), [Samba](#), [GNOME](#), [KDE](#), [Mozilla](#), [LibreOffice](#), [Vim](#), [TeX](#), [LaTeX](#), [DocBook](#), [Perl](#), [Python](#), [Tcl](#), [Java](#), [Ruby](#), [PHP](#), [Berkeley DB](#), [MariaDB](#), [PostgreSQL](#), [SQLite](#), [Exim](#), [Postfix](#), [Mutt](#), [FreeBSD](#), [OpenBSD](#), [Plan 9](#) și din multe alte proiecte independente de software liber. Debian integrează această diversitate de software liber într-un singur sistem.

Despre acest document

Reguli

La întocmirea acestui document s-au respectat următoarele reguli.

- Furnizează o imagine de ansamblu și omite cazurile speciale. (**Imaginea de ansamblu**)
- Menținerea acestuia într-o formă scurtă și simplă. (aplicarea principiului **KISS**)
- Aplicarea principiului „Nu reinventa roata!”. (Folosește trimiteri către **referințele existente**)
- Pune accentul pe instrumentele non-GUI și console. (Folosește **exemple shell**)
- Să fie obiectiv. (Folosește [popcon](#) etc.)

Indicație

S-a încercat să se elucideze aspectele ierarhice și nivelurile inferioare ale sistemului.

Cerințe prealabile



Avertisment

Se așteaptă ca dumneavoastră să depuneți eforturi susținute pentru a găsi răspunsuri pe cont propriu, dincolo de această documentație. Acest document oferă doar puncte de plecare eficiente.

Trebuie să căutați soluții pe cont propriu, din surse primare.

- Situl Debian la <https://www.debian.org> pentru informații generale
- Documentația din directorul „`/usr/share/doc/nume-pachet`”
- **Pagina de manual (manpage)** în stil Unix: «`dpkg -L nume-pachet | grep '/man/man.*/'`»
- **Pagina de informații info page)** în stil Unix: «`dpkg -L nume-pachet | grep '/info/'`»
- Raportul de eroare: <https://bugs.debian.org/nume-pachet>
- Wiki-ul Debian la <https://wiki.debian.org/> pentru subiecte specifice și în mișcare
- Specificația UNIX unică (Single UNIX Specification) de la Open Group [The UNIX System Home Page](#)
- Enciclopedia liberă Wikipedia la <https://www.wikipedia.org/>
- [Manualul administratorului Debian](#)
- Ghidurile HOWTO de la [Proiectul de documentație Linux \(TLPD\)](#)

Notă

Pentru documentație detaliată, poate fi necesar să instalați pachetul de documentație corespunzător, al cărui nume este numele pachetului cu sufixul „-doc”.

Convenții

Acest document oferă informații prin intermediul următorului stil de prezentare simplificat, cu exemple de comenzi shell bash(1).

```
# command-in-root-account
$ command-in-user-account
```

Aceste prompturi shell disting contul utilizat și corespund variabilelor de mediu definite astfel: „PS1= '\\$'” și „PS2= ' '”. Aceste valori sunt alese pentru a facilita citirea acestui document și nu sunt tipice pentru sistemul instalat efectiv.

Toate exemplele de comenzi sunt rulate în limba engleză „LANG=en_US.UTF8”. Vă rugăm să nu vă așteptați ca șirurile de caractere de substituție, cum ar fi *command-in-root-account* (comandă-în-contul-root) și *command-in-user-account* (comandă-în-contul-utilizatorului), să fie traduse în exemplele de comenzi. Aceasta este o alegere intenționată pentru a menține toate exemplele traduse la zi.

Notă

Consultați semnificația variabilelor de mediu „\$PS1” și „\$PS2” în bash(1).

Acțiunea necesară din partea administratorului de sistem este scrisă într-o propoziție imperativă, de exemplu „Apăsăți tasta Enter după ce introduceți fiecare șir de comenzi în shell”.

Coloana **descriere** și coloanele similare din tabel pot conține o **fraza nominală** care urmează [convenția de descriere scurtă a pachetului](#) care omite articolele introductive precum „un” și „o”. Alternativ, ele pot conține o frază infinitivă ca **fraza nominală** fără „la” la început, urmând convenția de descriere scurtă a comenzilor din paginile man. Acestea pot părea ciudate pentru unii, dar sunt alegeri stilistice intenționate ale mele pentru a păstra această documentație cât mai simplă posibil. Aceste **fraze nominale** nu încep cu majusculă și nu se termină cu punct, conform convenției de descriere scurtă.

Notă

Numele proprii, inclusiv numele comenzilor, își păstrează majusculele/minusculele indiferent de locația lor.

Un **fragment de comandă** citat într-un paragraf de text este indicat prin fontul de mașină de scris între ghilimele franceze, cum ar fi «*aptitude safe-upgrade*».

Un **text de date** dintr-un fișier de configurare citat într-un paragraf de text este indicat prin fontul de mașină de scris între ghilimele, cum ar fi „*deb-src*”.

O **comandă** este menționată prin numele său în fontul de mașină de scris, urmată opțional de numărul secțiunii din pagina de manual între paranteze, cum ar fi bash(1). Vă recomandăm să obțineți informații tastând următoarele.

```
$ man 1 bash
```

O **pagină de manual** este indicată prin numele său scris cu fontul de mașină de scris, urmat de numărul secțiunii paginii de manual între paranteze, cum ar fi *sources.list*(5). Vă recomandăm să obțineți informații tastând următoarele.

```
$ man 5 sources.list
```

O **pagină de informații** este indicată prin fragmentul de comandă scris cu fontul de mașină de scris între ghilimele, cum ar fi „*info make*”. Vă încurajăm să obțineți informații tastând următoarele.

```
$ info make
```

Un **nume de fișier** este indicat prin fontul de mașină de scris între ghilimele, cum ar fi „/etc/passwd”. Pentru fișierele de configurare, vă recomandăm să obțineți informații tastând următoarele.

```
$ sensible-pager "/etc/passwd"
```

Un **nume de director** este indicat prin fontul de mașină de scris între ghilimele, cum ar fi „/etc/apt/”. Vă încurajăm să explorați conținutul acestuia tastând următoarele.

```
$ mc "/etc/apt/"
```

Un **nume de pachet** este menționat prin numele său în fontul de mașină de scris, cum ar fi vim. Vă încurajăm să obțineți informații tastând următoarele.

```
$ dpkg -L vim
$ apt-cache show vim
$ aptitude show vim
```

O **documentație** poate indica locația sa prin numele fișierului în fontul de mașină de scris între ghilimele, cum ar fi „/usr/share/doc/base-passwd/users-and-groups.txt.gz” și „/usr/share/doc/base-passwd/users-and-” sau prin **adresa URL**, cum ar fi <https://www.debian.org>. Vă încurajăm să citiți documentația tastând următoarele.

```
$ zcat "/usr/share/doc/base-passwd/users-and-groups.txt.gz" | sensible-pager
$ sensible-browser "/usr/share/doc/base-passwd/users-and-groups.html"
$ sensible-browser "https://www.debian.org"
```

O **variabilă de mediu** este indicată prin numele său precedat de „\$” în fontul de mașină de scris între ghilimele, cum ar fi „\$TERM”. Vă recomandăm să obțineți valoarea sa curentă tastând următoarea comandă.

```
$ echo "$TERM"
```

Statistici de utilizare (popcon)

Datele [popcon](#) sunt prezentate ca măsură obiectivă a popularității fiecărui pachet. Acestea au fost descărcate la data de 2026-01-02 13:42:08 UTC și conțin totalul de 269003 de raportări referitoare la 215470 de pachete binare și 27 de arhitecturi.

Notă

Vă rugăm să rețineți că arhiva amd64 unstable conține în prezent numai pachete 74980. Datele popcon conțin rapoarte de la multe instalări de sistem vechi.

Numărul popcon precedat de „V:” pentru „voturi” se calculează astfel: „1000 * (numărul de trimiteri popcon pentru pachetul executat recent pe PC)/(numărul total de trimiteri popcon)”.

Numărul popcon precedat de „I:” pentru „instalări” se calculează astfel: „1000 * (numărul de rapoarte popcon pentru pachetul instalat în PC)/(numărul total de rapoarte popcon)”.

Notă

Cifrele popcon nu trebuie considerate ca fiind măsuri absolute ale importanței pachetelor. Există mulți factori care pot denatura statisticile. De exemplu, unele sisteme care participă la popcon pot avea directoare montate, cum ar fi „/usr/bin” cu opțiunea „noatime” pentru îmbunătățirea performanței sistemului și au dezactivat efectiv „votul” din astfel de sisteme.

Dimensiunea pachetului

Datele privind dimensiunea pachetului sunt, de asemenea, prezentate ca măsură obiectivă pentru fiecare pachet. Aceasta se bazează pe „Installed-Size:” raportată de comanda «`apt-cache show`» sau «`aptitude show`» (în prezent pe arhitectura amd64 pentru versiunea unstable). Dimensiunea raportată este exprimată în Kio ([Kibi-octet](#) = unitate pentru 1024 octeți).

Notă

Un pachet cu o dimensiune numerică mică poate indica faptul că pachetul din versiunea unstable este un pachet fictiv care instalează alte pachete cu conținut semnificativ prin dependență. Pachetul fictiv permite o tranziție lină sau divizarea pachetului.

Notă

O dimensiune a pachetului urmată de „(*)” indică faptul că pachetul din versiunea unstable lipsește și că se utilizează în schimb dimensiunea pachetului pentru versiunea experimental.

Rapoarte de erori privind acest document

Vă rugăm să raportați erorile din pachetul `debian-reference` folosind `reportbug(1)` dacă găsiți probleme în acest document. Vă rugăm să includeți sugestii de corectare folosind «`diff -u`» în versiunea text simplu sau în sursă.

Sfaturi pentru utilizatorii noi

Iată câteva sfaturi pentru utilizatorii noi:

- Faceți o copie de rezervă a datelor
 - A se vedea Secțiune [10.2](#).
 - Protejați-vă parola și cheile de securitate
 - [KISS \(keep it simple stupid\)](#) - păstrează lucrurile simple, prostule
 - Nu complicați excesiv sistemul
 - Citiți fișierele jurnal
 - **PRIMA** eroare este cea care contează
 - [RTFM: „Read the fucking manual”](#) - (Citește naibii manualul!)
 - Caută în Internet înainte de a pune întrebări
 - Nu fi root când nu este necesar să fii
 - Nu te juca cu sistemul de gestionare a pachetelor
 - Nu tasta nimic din ceea ce nu înțelegi
 - Nu schimba permisiunile fișierului (înainte de revizuirea completă a securității)
 - Nu părăsi shell-ul root până nu **TESTEZI** modificările
 - Aveți întotdeauna la îndemână un suport de pornire alternativ (memorie USB, CD, ...)
-

Câteva citate pentru utilizatorii noi

Iată câteva citate interesante din lista de discuții Debian care pot ajuta la lămurirea noilor utilizatori.

- „Acesta este Unix. Îți oferă suficientă funie pentru a te spânzura.” --- Miquel van Smoorenburg <miquels at cistron.nl>
- „Unix este prietenos cu utilizatorul... Doar că este selectiv în ceea ce privește prietenii pe care și-i alege.” --- Tollef Fog Heen <tollef at add.no>

Wikipedia are un articol intitulat „[Unix philosophy](#)” (Filosofia Unix) care conține citate interesante.

Capitolul 1

Tutoriale GNU/Linux

Cred că învățarea unui sistem informatic este similară cu învățarea unei limbi străine. Deși manualele și documentația sunt utile, trebuie să exersați singur. Pentru a vă ajuta să începeți fără probleme, voi detalia câteva puncte de bază.

Construcția robustă a [Debian GNU/Linux](#) provine din sistemul de operare [Unix](#), adică un sistem de operare [multi-utilizator](#) și [multi-sarcină](#). Trebuie să învățați să profitați de forța acestor caracteristici și similitudini între Unix și GNU/Linux.

Nu vă feriti de textele orientate spre Unix și nu vă bazați exclusiv pe textele GNU/Linux, deoarece astfel veți pierde multe informații utile.

Notă

Dacă utilizați de ceva timp un sistem [de tip Unix](#) cu instrumente de linie de comandă, probabil că știți deja tot ce explic aici. Vă rugăm să utilizați acest articol ca o verificare a realității și o reîmprospătare a cunoștințelor.

1.1 Noțiuni de bază despre consolă

1.1.1 Promptul shell

La pornirea sistemului, vi se afișează ecranul de autentificare bazat pe caractere dacă nu ați instalat niciun mediu [GUI](#) precum [GNOME](#) sau [KDE](#). Să presupunem că numele gazdei dvs. este foo, promptul de autentificare arată astfel.

Dacă ați instalat un mediu [GUI](#), puteți accesa în continuare promptul de autentificare bazat pe caractere apăsând «Ctrl-Alt-F3» și puteți reveni la mediul GUI apăsând «Ctrl-Alt-F2» (pentru mai multe informații, a se vedea Secțiune [1.1.6](#) mai jos).

```
foo login:
```

La solicitarea de autentificare, introduceți numele de utilizator, de exemplu penguin, și apăsați tasta «Enter», apoi introduceți parola și apăsați din nou tasta «Enter».

Notă

Conform tradiției Unix, numele de utilizator și parola sistemului Debian sunt sensibile la majuscule și minuscule. Numele de utilizator este ales de obicei numai din litere mici. Primul cont de utilizator este creat de obicei în timpul instalării. Conturi de utilizator suplimentare pot fi create cu `adduser(8)` de către root.

Sistemul pornește cu mesajul de salut stocat în „/etc/motd” („Message Of The Day” - Mesajul zilei) și afișează o linie de comandă.

```
Debian GNU/Linux 12 foo tty3

foo login: penguin
Password:

Linux foo 6.5.0-0.deb12.4-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.5.10-1~bpo12+1 (2023-11-23) ↵
x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Last login: Wed Dec 20 09:39:00 JST 2023 on tty3
foo:~$
```

Acum vă aflați în [shell](#). Shell-ul interpretează comenzile dvs.

1.1.2 Promptul shell în GUI

Dacă ați instalat un mediu [GUI](#) în timpul instalării, la pornirea sistemului vi se va afișa ecranul grafic de autentificare. Introduceți numele de utilizator și parola pentru a vă autentifica în contul de utilizator fără privilegii. Utilizați tasta «Tab» pentru a naviga între numele de utilizator și parolă sau utilizați butonul principal al mouse-ului.

Puteți accesa promptul shell în mediul GUI pornind un program `x-terminal-emulator` precum `gnome-terminal(1)`, `rxvt(1)` sau `xterm(1)`. În mediul grafic de birou GNOME, apăsați tasta «SUPER» (tasta «Windows») și tastați „terminal” în câmpul de căutare.

În cazul altor sisteme grafice de birou (cum ar fi `fluxbox`), este posibil să nu existe un punct de pornire evident pentru meniu. Dacă se întâmplă acest lucru, încercați să faceți clic (dreapta) pe fundalul imaginii de birou și așteptați să apară un meniu.

1.1.3 Contul root

Contul root este denumit și [super-utilizator](#) sau utilizator privilegiat. Din acest cont, puteți efectua următoarele sarcini de administrare a sistemului.

- Citește, scrie și elimină orice fișiere din sistem, indiferent de permisiunile acestor fișiere.
- Stabilește drepturile de proprietate și permisiunile pentru orice fișier din sistem
- Stabilește parola pentru orice utilizator fără privilegii din sistem
- Se conectează la orice cont fără a introduce parola

Această putere nelimitată a contului root necesită să fiți atent și responsabil atunci când îl utilizați.



Avertisment

Nu divulgați niciodată parola de root altor persoane.

Notă

Permisunile unui fișier (inclusiv dispozitive hardware precum CD-ROM etc., care sunt doar un alt fișier pentru sistemul Debian) pot face ca acesta să fie inutilizabil sau inaccesibil pentru utilizatorii care nu sunt root. Deși utilizarea contului root este o modalitate rapidă de a testa acest tip de situație, rezolvarea acestora trebuie făcută prin stabilirea corespunzătoare a permisiunilor fișierului și a apartenenței utilizatorului la grup (a se vedea Secțiune 1.2.3).

1.1.4 Promptul shell-ului root

Iată câteva metode de bază pentru a obține promptul shell root utilizând parola root.

- Tastați root la promptul de autentificare bazat pe caractere.
- Tastați „su - l” din orice prompt al shell-ului utilizatorului.
 - Aceasta nu păstrează mediul utilizatorului curent.
- Tastați „su” din orice prompt al shell-ului utilizatorului.
 - Aceasta păstrează o parte din mediul utilizatorului curent.

1.1.5 Instrumente grafice de administrare a sistemului

Când meniul mediului grafic de birou nu pornește automat instrumentele de administrare a sistemului GUI cu privilegiile corespunzătoare, le puteți porni din promptul shell-ului root al emulatorului de terminal, cum ar fi `gnome-terminal(1)`, `rxvt(1)` sau `xterm(1)`. Consultați Secțiune 1.1.4 și Secțiune 7.9.

**Avertisment**

Nu porniți niciodată administratorul de afișare/sesiune cu interfață grafică sub contul root introducând root la promptul administratorului de afișare, cum ar fi `gdm3(1)`.

Nu rulați niciodată programe cu interfață grafică la distanță care nu sunt de încredere sub X Window atunci când sunt afișate informații critice, deoarece acestea pot intercepta ecranul X.

1.1.6 Console virtuale

În sistemul Debian implicit, există șase console de caractere comutabile [de tip VT100](#) disponibile pentru a porni shell-ul de comandă direct pe gazda Linux. Cu excepția cazului în care vă aflați într-un mediu cu interfață grafică, puteți comuta între consolele virtuale apăsând simultan tasta Alt-stânga și una dintre tastele F1 — F6. Fiecare consolă de caractere permite conectarea independentă la cont și oferă un mediu multiutilizator. Acest mediu multiutilizator este o caracteristică excelentă a Unix și foarte captivantă.

Dacă vă aflați în mediul cu interfață grafică, puteți accesa consola de caractere 3 apăsând tastele Ctrl-Alt-F3, adică apăsând simultan tasta Ctrl-stânga, tasta Alt-stânga și tasta F3. Puteți reveni la mediul cu interfață grafică, care rulează în mod normal pe consola virtuală 2, apăsând Alt-F2.

Alternativ, puteți trece la o altă consolă virtuală, de exemplu la consola 3, din linia de comandă.

```
# chvt 3
```

1.1.7 Cum să ieșiți din promptul liniei de comandă

Tastați `Ctrl-D`, adică apăsați simultan tasta `Ctrl` - stânga și tasta `d` în linia de comandă pentru a închide activitatea shell-ului. Dacă vă aflați în consola de caractere, veți reveni la linia de autentificare. Chiar dacă aceste caractere de control sunt denumite «control D» cu majusculă, nu este necesar să apăsați tasta «Shift». Expresia prescurtată, `^D`, este utilizată de asemenea pentru `Ctrl-D`. Alternativ, puteți tasta „exit”.

Dacă vă aflați în `x-terminal-emulator(1)`, puteți închide fereastra `x-terminal-emulator` cu această comandă.

1.1.8 Cum să opriți sistemul

La fel ca orice alt sistem de operare modern în care operațiile cu fișiere implică [stocarea datelor în memoria cache](#) pentru o performanță îmbunătățită, sistemul Debian necesită o procedură de oprire corespunzătoare înainte ca alimentarea să poată fi oprită în siguranță. Acest lucru are scopul de a menține integritatea fișierelor, forțând toate modificările din memorie să fie scrise pe disc. Dacă este disponibilă funcția de control al alimentării software, procedura de oprire oprește automat alimentarea sistemului. (În caz contrar, poate fi necesar să apăsați butonul de alimentare timp de câteva secunde după procedura de oprire.)

Puteți opri sistemul în modul multiutilizator normal din linia de comandă.

```
# shutdown -h now
```

Puteți opri sistemul în modul utilizator unic din linia de comandă.

```
# poweroff -i -f
```

Consultați Secțiune [6.3.8](#).

1.1.9 Recuperarea unei console funcționale

Când ecranul devine nebun după ce faceți unele lucruri ciudate, cum ar fi „*cat vreun-fișier-binar*”, tastați „reset” la promptul de comandă. Este posibil să nu puteți vedea comanda introdusă pe măsură ce o tastați. De asemenea, puteți tasta „*c lear*” pentru a curăța ecranul.

1.1.10 Sugestii de pachete suplimentare pentru începători

Deși chiar și instalarea minimă a sistemului Debian fără niciun mediu grafic de birou oferă funcționalitatea de bază Unix, este o idee bună să instalați câteva pachete suplimentare pentru terminalul caracter bazat pe linia de comandă și curses, cum ar fi `mc` și `vim` cu `apt-get(8)` pentru începători, pentru a începe cu următoarele.

```
# apt-get update
...
# apt-get install mc vim sudo aptitude
...
```

Dacă aveți deja aceste pachete instalate, nu se instalează pachete noi.

Ar fi o idee bună să citiți câteva documente informative.

Puteți instala unele dintre aceste pachete urmând pașii de mai jos.

```
# apt-get install package_name
```


pachet	popcon(popularitate)	descriere
mc	V:42, I:189	1590
sudo	V:738, I:863	6768
vim	V:85, I:353	4077
vim-tiny	V:55, I:977	1873
emacs-nox	V:3, I:13	46536
w3m	V:11, I:151	2853
gpm	V:9, I:10	545

Tabela 1.1: Lista pachetelor de programe interesante în modul text

pachet	popcon(popularitate)	descriere	
doc-debian	I:879	187	Documentația proiectului Debian (Întrebări frecvente despre Debian) și alte documente
debian-policy	I:8	5061	Manualul de politici Debian și documentele conexe
developers-reference	V:0, I:3	2602	Ghiduri și informații pentru dezvoltatorii Debian
debmake-doc	I:0	12503	Ghid pentru responsabili cu întreținerea pachetelor Debian
debian-history	I:0	6251	Istoria proiectului Debian
debian-faq	I:878	798	Întrebări frecvente despre Debian (FAQ)

Tabela 1.2: Lista pachetelor de documentație utilă

1.1.11 Un cont de utilizator suplimentar

Dacă nu doriți să utilizați contul principal de utilizator pentru următoarele activități de instruire, puteți crea un cont de utilizator pentru instruire, de exemplu `fish`, urmând pașii de mai jos.

```
# adduser fish
```

Răspundeți la toate întrebările.

Aceasta creează un cont nou numit `fish`. După ce ați exersat, puteți șterge acest cont de utilizator și directorul său principal urmând pașii de mai jos.

```
# deluser --remove-home fish
```

În sistemele non-Debian și Debian specializate, activitățile de mai sus trebuie să utilizeze în schimb instrumentele de nivel inferior `useradd(8)` și `userdel(8)`.

1.1.12 Configurarea «sudo»

Pentru stațiile de lucru tipice cu un singur utilizator, cum ar fi sistemul Debian cu mediu grafic de birou pe laptop, este obișnuit să se implementeze o configurație simplă a `sudo(8)` după cum urmează, pentru a permite utilizatorului fără privilegii, de exemplu `penguin`, să obțină privilegii administrative doar cu parola sa de utilizator, fără a fi nevoie de parola root.

```
# echo "penguin ALL=(ALL) ALL" >> /etc/sudoers
```

Alternativ, este de asemenea obișnuit să se procedeze după cum urmează pentru a permite utilizatorului fără privilegii, de exemplu `penguin`, să obțină privilegii administrative fără nicio parolă.

```
# echo "penguin ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers
```

Această truc ar trebui folosit numai pentru stația de lucru cu un singur utilizator pe care o administrați și unde sunteți singurul utilizator.



Avertisment

Nu configurați conturi de utilizatori obișnuiți pe stații de lucru multiutilizator de acest tip, deoarece acest lucru ar afecta grav securitatea sistemului.



Atenție

Parola și contul penguin din exemplul de mai sus necesită aceeași protecție ca și parola root și contul root. În acest context, privilegiul administrativ aparține unei persoane autorizate să îndeplinească sarcini de administrare a sistemului pe stația de lucru. Nu acordați niciodată un astfel de privilegiu unui responsabil din departamentul administrativ al companiei dvs. sau șefului dvs., cu excepția cazului în care aceștia sunt autorizați și capabili să îndeplinească astfel de sarcini.

Notă

Pentru a acorda privilegii de acces la dispozitive și fișiere limitate, ar trebui să luați în considerare utilizarea **grupului** pentru a acorda acces limitat, în loc să utilizați privilegiul root prin intermediul sudo(8).

Cu o configurare mai atentă și mai grijulie, sudo(8) poate acorda privilegii administrative limitate altor utilizatori dintr-un sistem partajat, fără a partaja parola root. Acest lucru poate ajuta la asigurarea responsabilității în cazul gazdelor cu mai mulți administratori, astfel încât să puteți ști cine a făcut ce. Pe de altă parte, este posibil să nu doriți ca altcineva să aibă astfel de privilegii.

1.1.13 Ora de joacă

Acum sunteți gata să utilizați sistemul Debian fără riscuri, atâta timp cât utilizați contul de utilizator fără privilegii.

Acest lucru se datorează faptului că sistemul Debian, chiar și după instalarea implicită, este configurat cu permisiuni de fișiere adecvate, care împiedică utilizatorii fără privilegii să deterioreze sistemul. Desigur, pot exista încă unele vulnerabilități care pot fi exploatate, dar cei care sunt preocupați de aceste probleme nu ar trebui să citească această secțiune, ci ar trebui să citească [Manualul de securizare Debian](#).

Vom învăța sistemul Debian ca un sistem [de tip Unix](#) cu ajutorul următoarelor:

- Secțiune [1.2](#) (conceptul de bază)
- Secțiune [1.3](#) (metoda de supraviețuire)
- Secțiune [1.4](#) (metoda de bază)
- Secțiune [1.5](#) (mecanismul shell)
- Secțiune [1.6](#) (metoda de procesare a textului)

1.2 Sistem de fișiere de tip Unix

În GNU/Linux și alte sisteme de operare [de tip Unix](#), [fișierele](#) sunt organizate în [directoare](#). Toate fișierele și directoarele sunt aranjate într-o singură structură arborescentă cu rădăcina în „/”. Se numește structură arborescentă deoarece, dacă desenezi sistemul de fișiere, acesta arată ca un arbore, dar cu rădăcina în sus.

Aceste fișiere și directoare pot fi distribuite pe mai multe dispozitive. mount(8) servește la atașarea sistemului de fișiere găsit pe un dispozitiv la arborele de fișiere mare. În schimb, umount(8) îl detașează din nou. Pe nucleele

Linux recente, `mount(8)` cu anumite opțiuni poate lega o parte a arborelui de fișiere în altă parte sau poate monta sistemul de fișiere ca fiind partajat, privat, secundar sau nelegat. Opțiunile de montare acceptate pentru fiecare sistem de fișiere sunt disponibile în `„/usr/share/doc/linux-doc-*/Documentation/filesystems/”`.

Directoarele din sistemele Unix sunt denumite **dosare** în alte sisteme. De asemenea, rețineți că nu există conceptul de **unitate** precum „A:” pe niciun sistem Unix. Există un singur sistem de fișiere și totul este inclus. Acesta este un avantaj enorm în comparație cu Windows.

1.2.1 Noțiuni de bază despre fișierele Unix

Iată câteva noțiuni de bază despre fișierele Unix.

- Numele fișierelor sunt **sensibile la majuscule și minuscule**. Adică, „FIȘIERULMEU” și „FișierulMEU” sunt fișiere diferite.
- **Directorul rădăcină** înseamnă rădăcina sistemului de fișiere, denumită simplu „/”. Nu confundați acest lucru cu directorul de bază al utilizatorului rădăcină: „/root”.
- Fiecare director are un nume care poate conține orice litere sau simboluri **cu excepția „/”**. Directorul rădăcină este o excepție; numele său este „/” (pronunțat „bară oblică” sau „directorul rădăcină”) și nu poate fi redenumit.
- Fiecare fișier sau director este desemnat printr-un **nume de fișier complet calificat**, **nume de fișier absolut** sau **rută**, indicând secvența de directoare care trebuie parcurse pentru a ajunge la acesta. Cei trei termeni sunt sinonimi.
- Toate **numele de fișiere complet calificate** încep cu directorul „/” și există o „/” între fiecare director sau fișier din numele fișierului. Primul „/” este directorul de nivel superior, iar celelalte „/” separă subdirectoarele succesive, până când ajungem la ultima intrare, care este numele fișierului propriu-zis. Cuvintele folosite aici pot fi confuze. Luați ca exemplu următorul **nume de fișier complet calificat**: „/usr/share/keytables/us.map.gz”. Cu toate acestea, oamenii se referă la numele său de bază „us.map.gz” ca fiind numele fișierului.
- Directorul rădăcină are o serie de ramuri, cum ar fi „/etc/” și „/usr/”. Aceste subdirectoare se ramifică la rândul lor în și mai multe subdirectoare, cum ar fi „/etc/systemd/” și „/usr/local/”. Ansamblul este denumit **arborele de directoare**. Puteți considera un nume de fișier absolut ca o rută de la baza arborelui („/”) până la capătul unei ramuri (un fișier). De asemenea, veți auzi oamenii vorbind despre arborele de directoare ca și cum ar fi un **arborele genealogic** care cuprinde toți descendenții direcți ai unei singure entități numite directorul rădăcină („/”): astfel, subdirectoarele au **părinți**, iar o rută arată ascendența completă a unui fișier. Există, de asemenea, rute relative care încep în alt loc decât directorul rădăcină. Trebuie să rețineți că directorul „./” se referă la directorul părinte. Această terminologie se aplică și altor structuri de tip director, cum ar fi structurile de date ierarhice.
- Nu există nicio componentă specială a numelui rutei de director care să corespundă unui dispozitiv fizic, cum ar fi discul dur. Acest lucru diferă de [RT-11](#), [CP/M](#), [OpenVMS](#), [MS-DOS](#), [AmigaOS](#) și [Microsoft Windows](#), unde ruta conține un nume de dispozitiv, cum ar fi „C:\”. (Cu toate acestea, există intrări de director care se referă la dispozitive fizice ca parte a sistemului de fișiere normal. Consultați Secțiune [1.2.2](#).)

Notă

Deși **puteți** utiliza aproape orice litere sau simboluri într-un nume de fișier, în practică nu este recomandat să faceți acest lucru. Este mai bine să evitați orice caractere care au adesea semnificații speciale în linia de comandă, inclusiv spații, tabulatoare, linii noi și alte caractere speciale: { } () [] ' ` " \ / > < | ; ! # & ^ * % @ \$. Dacă doriți să separați cuvintele dintr-un nume, alegeți bune sunt punctul, cratima și linia de subliniere. De asemenea, puteți scrie fiecare cuvânt cu majusculă la început, „CaAcesta”. Utilizatorii experimentați de Linux tind să evite spațiile în numele fișierelor.

Notă

Cuvântul „root” poate însemna fie „utilizatorul root”, fie „directorul root”. Contextul utilizării lor ar trebui să clarifice acest lucru.

Notă

Cuvântul **rută (path)** este folosit nu doar pentru **nume de fișier complet calificat**, așa cum s-a menționat mai sus, ci și pentru **ruta de căutare a comenzii**. Semnificația intenționată este de obicei clară din context.

Cele mai bune practici detaliate pentru ierarhia fișierelor sunt descrise în Filesystem Hierarchy Standard -- Standardul de ierarhie a sistemului de fișiere („/usr/share/doc/debian-policy/fhs/fhs-2.3.txt.gz” și hier(7)). Ar trebui să rețineți următoarele aspecte ca punct de plecare.

director	utilizarea directorului
/	directorul rădăcină
/etc/	fișierele de configurare la nivel de sistem
/var/log/	fișierele de jurnal ale sistemului
/home/	toate directoarele personale (acasă) pentru toți utilizatorii fără privilegii

Tabela 1.3: Lista directoarelor cheie și descrierea utilizării acestora

1.2.2 Elementele interne ale sistemului de fișiere

Urmând **tradiția Unix**, sistemul Debian GNU/Linux oferă **sistemul de fișiere** în care se află datele fizice de pe discurile dure și alte dispozitive de stocare, iar interacțiunea cu dispozitivele hardware, cum ar fi ecranele consolei și consolele seriale la distanță, este reprezentată într-o manieră unificată sub „/dev/”.

Fiecare fișier, director, conductă cu nume (o modalitate prin care două programe pot partaja date) sau dispozitiv fizic dintr-un sistem Debian GNU/Linux are o structură de date numită **nod-i** care descrie atributele asociate, cum ar fi utilizatorul care îl deține (proprietarul), grupul căruia îi aparține, ora ultimei accesări etc. Ideea de a reprezenta aproape totul în sistemul de fișiere a fost o inovație Unix, iar nucleele Linux moderne au dezvoltat această idee și mai mult. Acum, chiar și informațiile despre procesele care rulează în calculator pot fi găsite în sistemul de fișiere.

Această reprezentare abstractă și unificată a entităților fizice și a proceselor interne este foarte puternică, deoarece ne permite să folosim aceeași comandă pentru același tip de operație pe multe dispozitive total diferite. Este chiar posibil să se modifice modul de funcționare al nucleului prin scrierea de date în fișiere speciale care sunt legate de procesele în execuție.

Indicație

Dacă trebuie să identificați corespondența dintre arborele de fișiere și entitatea fizică, executați `mount(8)` fără argumente.

1.2.3 Permisuni ale sistemului de fișiere

Permisunile sistemului de fișiere ale unui sistem precum **Unix** sunt definite pentru trei categorii de utilizatori afectați.

- **Utilizatorul** care deține fișierul (**u**)
- Alți utilizatori din **grupul** căruia aparține fișierul (**g**)
- Toți **ceilalți** utilizatori (**o**), denumiți și „lumea” și „toată lumea”

Pentru fișier, fiecare permisiune corespunzătoare permite următoarele acțiuni.

- Permisunea de **citare - (read) (r)** permite proprietarului să examineze conținutul fișierului.

- Permisiunea de **scriere - (write) (w)** permite proprietarului să modifice fișierul.
- Permisiunea de **executare - (execute) (x)** permite proprietarului să execute fișierul ca o comandă.

Pentru director, fiecare permisiune corespunzătoare permite următoarele acțiuni.

- Permisiunea de **citire - (read) (r)** permite proprietarului să afișeze conținutul directorului.
- Permisiunea de **scriere - (write) (w)** permite proprietarului să adauge sau să elimine fișiere din director.
- Permisiunea de **executare - (execute) (x)** permite proprietarului să acceseze fișierele din director.

Aici, permisiunea de **executare - (execute)** asupra unui director înseamnă nu numai să permită citirea fișierelor din acel director, ci și să permită vizualizarea atributelor acestora, cum ar fi dimensiunea și data modificării.

Instrumentul `ls(1)` este utilizat pentru a afișa informații privind permisiunile (și altele) pentru fișiere și directoare. Când este invocat cu opțiunea „-l”, afișează următoarele informații în ordinea indicată.

- **Tipul fișierului** (primul caracter)
- **Permisiunea** de acces la fișier (nouă caractere, compuse din câte trei caractere pentru utilizator, grup și altele, în această ordine)
- **Numărul de legături dure** către fișier
- Numele **utilizatorului** care deține fișierul
- Numele **grupului** căruia îi aparține fișierul
- **Dimensiunea** fișierului în caractere (octeți)
- **Data și ora** fișierului (mtime)
- **Numele** fișierului

caracter	semnificație
-	fișier normal
d	director
l	legătură simbolică
c	nod de dispozitiv de caractere
b	nod de dispozitiv de blocuri
p	conductă cu nume
s	soclu

Tabela 1.4: Lista descriptivă a primului caracter din ieșirea comenzii «`ls -l`»

`chown(1)` este utilizat din contul root pentru a schimba proprietarul fișierului. `chgrp(1)` este utilizat din contul proprietarului fișierului sau din contul root pentru a schimba grupul fișierului. `chmod(1)` este utilizat din contul proprietarului fișierului sau din contul root pentru a schimba permisiunile de acces la fișiere și directoare. Sintaxa de bază pentru manipularea unui fișier `foo` este următoarea.

```
# chown newowner foo
# chgrp newgroup foo
# chmod [ugoa][+ -=][rwxXst][, ...] foo
```

De exemplu, puteți crea o structură de directoare care să aparțină utilizatorului `foo` și să fie partajată de grupul `bar` folosind următoarea comandă.

```
# cd /some/location/
# chown -R foo:bar .
# chmod -R ug+rwX,o=rX .
```

Există încă trei biți de permisiune specială.

- Bit-ul **set user ID** (**s** sau **S** în loc de **x** al utilizatorului)
- Bit-ul **set group ID** (**s** sau **S** în loc de **x** al grupului)
- Bit-ul **sticky** (**t** sau **T** în loc de **x** al celorlalți)

Aici, rezultatul comenzii „ls -l” pentru acești biți este **cu majuscule** dacă biții de execuție ascunși de aceste rezultate sunt **neactivați**.

Activarea bitului **set user ID** pe un fișier executabil permite utilizatorului să execute fișierul executabil cu ID-ul propri-
etarului fișierului (de exemplu **root**). În mod similar, activarea bitului **set group ID** pentru un fișier executabil permite
utilizatorului să execute fișierul executabil cu ID-ul de grup al fișierului (de exemplu, **root**). Deoarece activările acestor
biți pot genera riscuri de securitate, activarea lor necesită o atenție sporită.

Activarea bitului **set group ID** pentru un director activează schema de creare a fișierelor **tip BSD** în care toate fișierele
create în director aparțin **grupului** directorului.

Activarea bitului **sticky** pe un director împiedică ștergerea unui fișier din director de către un utilizator care nu este
proprietarul fișierului. Pentru a securiza conținutul unui fișier din directoare care pot fi scrise de toată lumea, cum ar fi
„/tmp” sau din directoare care pot fi scrise de grup, nu trebuie doar să redefiniți permisiunea de **scriere** pentru fișier,
ci și să activați bitul **sticky** pe director. În caz contrar, fișierul poate fi șters și poate fi creat un fișier nou cu același
nume de către orice utilizator care are acces de scriere la director.

Iată câteva exemple interesante de permisiuni pentru fișiere.

```
$ ls -l /etc/passwd /etc/shadow /dev/ppp /usr/sbin/exim4
crw-----T 1 root root    108, 0 Oct 16 20:57 /dev/ppp
-rw-r--r-- 1 root root      2761 Aug 30 10:38 /etc/passwd
-rw-r----- 1 root shadow  1695 Aug 30 10:38 /etc/shadow
-rwsr-xr-x 1 root root    973824 Sep 23 20:04 /usr/sbin/exim4
$ ls -ld /tmp /var/tmp /usr/local /var/mail /usr/src
drwxrwxrwt 14 root root   20480 Oct 16 21:25 /tmp
drwxrwsr-x 10 root staff   4096 Sep 29 22:50 /usr/local
drwxr-xr-x 10 root root     4096 Oct 11 00:28 /usr/src
drwxrwsr-x  2 root mail     4096 Oct 15 21:40 /var/mail
drwxrwxrwt  3 root root     4096 Oct 16 21:20 /var/tmp
```

Există un mod numeric alternativ pentru a descrie permisiunile fișierelor cu **chmod(1)**. Acest mod numeric utilizează
numere octale (în bază=8) cu 3 până la 4 cifre.

cifra	semnificație
prima cifră (opțională)	suma dintre bitul set user ID (=4), bitul set group ID (=2) și bitul sticky (=1)
a doua cifră	suma permisiunilor citire (=4), scriere (=2) și executare (=1) pentru utilizator
a treia cifră	la fel ca a doua cifră, dar pentru grup
a patra cifră	la fel ca a doua cifră, dar pentru ceilalți

Tabela 1.5: Modul numeric pentru permisiunile de fișiere în comenzile **chmod(1)**

Sună complicat, dar de fapt este destul de simplu. Dacă vă uitați la primele câteva coloane (2-10) din „ls -l” și le
citiți ca o reprezentare binară (bază=2) a permisiunilor de fișier („-” fiind „0” și „rwx” fiind „1”), ultimele 3 cifre ale valorii
numerice ale modului ar trebui să aibă sens ca o reprezentare octală (bază=8) a permisiunilor de fișier.

De exemplu, încercați următoarele

```
$ touch foo bar
$ chmod u=rw,go=r foo
$ chmod 644 bar
$ ls -l foo bar
-rw-r--r-- 1 penguin penguin 0 Oct 16 21:39 bar
-rw-r--r-- 1 penguin penguin 0 Oct 16 21:35 foo
```

Indicație

Dacă trebuie să accesați informațiile afișate de „ls -l” într-un script shell, trebuie să utilizați comenzi relevante, cum ar fi `test(1)`, `stat(1)` și `readlink(1)`. Se pot utiliza și comenzi încorporate în shell, cum ar fi „[” sau „test”.

1.2.4 Controlul permisiunilor pentru fișierele nou create: umask

Permisiunile aplicate unui fișier sau director nou creat sunt restricționate de comanda încorporată în shell `umask`. Consultați `dash(1)`, `bash(1)` și `builtins(7)`.

```
(file permissions) = (requested file permissions) & ~(umask value)
```

umask	permisiunile fișierelor create	permisiunile directoarelor create	utilizare
0022	-rw-r--r--	-rwxr-xr-x	poate fi scris numai de către utilizator
0002	-rw-rw-r--	-rwxrwxr-x	poate fi scris de grup

Tabela 1.6: Exemple de valori pentru **umask**

Sistemul Debian utilizează în mod implicit o schemă de grupuri private de utilizatori („user private group”: UPG). Un UPG este creat de fiecare dată când un nou utilizator este adăugat în sistem. Un UPG are același nume ca utilizatorul pentru care a fost creat, iar acel utilizator este singurul membru al UPG. Schema UPG face ca stabilirea `umask` la 0002 să fie sigură, deoarece fiecare utilizator are propriul grup privat; (în unele variante Unix, este destul de obișnuit să se configureze toți utilizatorii normali aparținând unui singur grup **users** și este o idee bună să se stabilească `umask` la 0022 pentru securitate în astfel de cazuri).

Indicație

Activați UPG introducând „`umask 002`” în fișierul `~/ .bashrc`.

1.2.5 Permisiuni pentru grupuri de utilizatori (grup)

**Avertisment**

Asigurați-vă că salvați modificările nesalvate înainte de a reporni sau de a efectua acțiuni similare.

Puteți adăuga un utilizator penguin la un grup bird în doi pași:

- Modificați configurația grupului utilizând una dintre următoarele opțiuni:
 - Executați «`sudo usermod -aG bird penguin`».
 - Executați «`sudo adduser penguin bird`» (numai pe sistemele Debian tipice)
 - Executați «`sudo vigr`» pentru `/etc/group` și «`sudo vigr -s`» pentru `/etc/gshadow` pentru a adăuga penguin în linia pentru bird.
- Aplicați configurația utilizând una dintre următoarele opțiuni:
 - Reporniți sistemul și conectați-vă (cea mai bună opțiune)
 - Executați «`kill -TERM -1`» și efectuați câteva acțiuni de remediere, cum ar fi «`systemctl restart NetworkManager`».

- Deconectați-vă prin meniul interfeței grafice și conectați-vă.

Puteți elimina un utilizator penguin dintr-un grup bird în doi pași:

- Modificați configurația grupului utilizând una dintre următoarele opțiuni:
 - Executați «`sudo usermod -rG bird penguin`».
 - Executați «`sudo deluser penguin bird`» (numai pe sistemele Debian tipice)
 - Executați «`sudo vigr`» pentru `/etc/group` și «`sudo vigr -s`» pentru `/etc/gshadow` pentru a elimina penguin din linia pentru bird.
- Aplicați configurația utilizând una dintre următoarele opțiuni:
 - Reporniți sistemul și conectați-vă (cea mai bună opțiune)
 - Executați «`kill -TERM -1`» și efectuați câteva acțiuni de remediere, cum ar fi «`systemctl restart NetworkManager`»
 - Deconectarea prin meniul interfeței grafice nu este o opțiune pentru mediul grafic de birou Gnome.

Orice încercare de repornire la cald este o înlocuire fragilă a repornirii la rece reale în cadrul sistemului de birou modern.

Notă

Alternativ, puteți adăuga dinamic utilizatori la grupuri în timpul procesului de autentificare, adăugând linia „`auth optional pam_group.so`” la „`/etc/pam.d/common-auth`” și configurând „`/etc/security/group.conf`”; (consultați Cap. 4).

Dispozitivele fizice sunt doar un alt tip de fișier în sistemul Debian. Dacă aveți probleme cu accesarea dispozitivelor precum CD-ROM-ul și memoria USB dintr-un cont de utilizator, ar trebui să adăugați utilizatorul respectiv ca membru al grupului relevant.

Unele grupuri importante (notabile) furnizate de sistem permit membrilor lor să acceseze anumite fișiere și dispozitive fără privilegii root.

grupul	descrierea fișierelor și dispozitivelor accesibile
<code>dialout</code>	acces complet și direct la porturile seriale („ <code>/dev/ttyS[0-3]</code> ”)
<code>dip</code>	acces limitat la porturile seriale pentru conexiunea Dialup IP la mașini de încredere
<code>cdrom</code>	unități CD-ROM, DVD+/-RW
<code>audio</code>	dispozitiv audio
<code>video</code>	dispozitiv video
<code>scanner</code>	scaner(e)
<code>adm</code>	jurnalele de monitorizare a sistemului
<code>staff</code>	unele directoare pentru lucrări administrative minore: „ <code>/usr/local</code> ”, „ <code>/home</code> ”

Tabela 1.7: Lista grupurilor notabile furnizate de sistem pentru accesul la fișiere

Indicație

Trebuie să aparțineți grupului `dialout` pentru a reconfigura modemul, a forma orice număr etc. Dar dacă root creează fișiere de configurare predefinite pentru partenerii (mașinile) de încredere în „`/etc/ppp/peers/`”, trebuie doar să aparțineți grupului `dip` pentru a crea o conexiune **Dialup IP** la acele mașini de încredere folosind comenzile `pppd(8)`, `pon(1)` și `poff(1)`.

Unele grupuri notabile furnizate de sistem permit membrilor lor să execute anumite comenzi fără privilegii root.

Pentru lista completă a utilizatorilor și grupurilor furnizate de sistem, consultați versiunea recentă a documentului „Utilizatori și grupuri” din „`/usr/share/doc/base-passwd/users-and-groups.html`” furnizat de pachetul `base-passwd`.

Consultați `passwd(5)`, `group(5)`, `shadow(5)`, `newgrp(1)`, `vipw(8)`, `vigr(8)` și `pam_group(8)` pentru comenzile de gestionare a utilizatorilor și grupurilor din sistem.

grupul	comenzi disponibile
sudo	execută orice comandă cu privilegii de superutilizator
lpadmin	execută comenzi pentru a adăuga, modifica și elimina imprimante din bazele de date ale imprimantelor

Tabela 1.8: Lista grupurilor notabile furnizate de sistem pentru executarea anumitor comenzi

1.2.6 Marcaje de timp

Există trei tipuri de marcaje de timp pentru un fișier GNU/Linux.

tipul	semnificația (definiția istorică Unix)
mtime	momentul modificării fișierului (ls -l)
ctime	momentul modificării stării fișierului (ls -lc)
atime	ultima dată când a fost accesat fișierul (ls -lu)

Tabela 1.9: Lista tipurilor de marcaje de timp

Notă

ctime nu este ora creării fișierului.

Notă

Valoarea reală a **atime** pe sistemul GNU/Linux poate fi diferită de cea din definiția istorică Unix.

- Suprascrierea unui fișier modifică toate atributele **mtime**, **ctime** și **atime** ale fișierului.
- Schimbarea proprietarului sau a permisiunilor unui fișier modifică atributele **ctime** și **atime** ale fișierului.
- Citirea unui fișier modifică atributul **atime** al fișierului în sistemul Unix clasic.
- Citirea unui fișier modifică atributul **atime** al fișierului pe sistemul GNU/Linux dacă sistemul său de fișiere este montat cu „strictatime”.
- Citirea unui fișier pentru prima dată sau după o zi modifică atributul **atime** al fișierului pe sistemul GNU/Linux dacă sistemul său de fișiere este montat cu „relatime”; comportamentul implicit începând cu Linux 2.6.30).
- Citirea unui fișier nu modifică atributul **atime** al fișierului pe sistemul GNU/Linux dacă sistemul său de fișiere este montat cu „noatime”.

Notă

Opțiunile de montare „noatime” și „relatime” sunt introduse pentru a îmbunătăți performanța de citire a sistemului de fișiere în condiții normale de utilizare. Operația simplă de citire a fișierelor în cadrul opțiunii „strictatime” însoțește operația de scriere care consumă mult timp pentru a actualiza atributul **atime**. Dar atributul **atime** este rar utilizat, cu excepția fișierului mbox(5). Consultați mount(8).

Utilizați comanda touch(1) pentru a modifica marcajele temporale ale fișierelor existente.

Pentru marcajele temporale, comanda ls afișează șiruri traduse în limbi diferite de engleză („fr_FR.UTF-8”).

```
$ LANG=C ls -l foo
-rw-rw-r-- 1 penguin penguin 0 Oct 16 21:35 foo
$ LANG=en_US.UTF-8 ls -l foo
-rw-rw-r-- 1 penguin penguin 0 Oct 16 21:35 foo
$ LANG=fr_FR.UTF-8 ls -l foo
-rw-rw-r-- 1 penguin penguin 0 oct. 16 21:35 foo
```

Indicație

Consultați Secțiune [9.3.4](#) pentru a personaliza ieșirea comenzii «ls -l».

1.2.7 Legături

Există două metode de asociere a unui fișier „foo” cu un nume de fișier diferit „bar”.

- [Legătură dură](#)
 - Nume duplicat pentru un fișier existent
 - «ln foo bar»
- [Legătură simbolică sau symlink](#)
 - Fișier special care indică un alt fișier după nume
 - «ln -s foo bar»

A se vedea exemplul următor pentru modificările numărului de legături și diferențele subtile în rezultatul comenzii «rm».

```
$ umask 002
$ echo "Original Content" > foo
$ ls -li foo
1449840 -rw-rw-r-- 1 penguin penguin 17 Oct 16 21:42 foo
$ ln foo bar      # hard link
$ ln -s foo baz   # symlink
$ ls -li foo bar baz
1449840 -rw-rw-r-- 2 penguin penguin 17 Oct 16 21:42 bar
1450180 lrwxrwxrwx 1 penguin penguin  3 Oct 16 21:47 baz -> foo
1449840 -rw-rw-r-- 2 penguin penguin 17 Oct 16 21:42 foo
$ rm foo
$ echo "New Content" > foo
$ ls -li foo bar baz
1449840 -rw-rw-r-- 1 penguin penguin 17 Oct 16 21:42 bar
1450180 lrwxrwxrwx 1 penguin penguin  3 Oct 16 21:47 baz -> foo
1450183 -rw-rw-r-- 1 penguin penguin 12 Oct 16 21:48 foo
$ cat bar
Original Content
$ cat baz
New Content
```

Legătura dură poate fi creată în cadrul aceluiași sistem de fișiere și are același număr de nod-i, pe care îl afișează opțiunea „-i” cu ls(1) .

Legătura simbolică are întotdeauna permisiuni nominale de acces la fișiere de tipul „rwxrwxrwx”, așa cum se arată în exemplul de mai sus, permisiunile efective de acces fiind dictate de permisiunile fișierului către care face trimitere.

**Atenție**

În general, este o idee bună să nu creați legături simbolice sau legături dure complicate, cu excepția cazului în care aveți un motiv foarte bun. Acest lucru poate provoca probleme în care combinația logică a legăturilor simbolice duce la bucle în sistemul de fișiere.

Notă

În general, este preferabil să utilizați legături simbolice în locul legăturilor dure, cu excepția cazului în care aveți un motiv întemeiat pentru a utiliza o legătură dură.

Directorul „.” conține legături către directorul în care apare, astfel încât numărul de legături ale oricărui director nou începe de la 2. Directorul „. .” conține legături către directorul părinte, astfel încât numărul de legături ale directorului crește odată cu adăugarea de noi subdirectoare.

Dacă tocmai treceți de la Windows la Linux, veți observa rapid cât de bine concepută este legarea fișierelor în Unix, în comparație cu echivalentul Windows al „scurtăturilor”. Deoarece este implementată în sistemul de fișiere, aplicațiile nu pot vedea nicio diferență între un fișier legat și original. În cazul legăturilor dure, nu există nicio diferență.

1.2.8 Conducute cu nume (FIFO)

O [conductă cu nume](#) este un fișier care funcționează ca o conductă. Introduceți ceva în fișier și acesta iese la celălalt capăt. De aceea se numește FIFO, sau First-In-First-Out (primul intrat, primul ieșit): primul lucru pe care îl introduceți în conductă este primul lucru care iese la celălalt capăt.

Dacă scrieți într-o conductă cu nume, procesul care scrie în conductă nu se termină până când informațiile scrise nu sunt citite din conductă. Dacă citiți dintr-o conductă cu nume, procesul de citire așteaptă până când nu mai este nimic de citit înainte de a se termina. Dimensiunea conductei este întotdeauna zero --- nu stochează date, ci doar leagă două procese, similar funcționalității oferite de sintaxa shell „|”. Cu toate acestea, deoarece această conductă are un nume, cele două procese nu trebuie să se afle pe aceeași linie de comandă și nici măcar să fie rulate de același utilizator. Conducutele au reprezentat o inovație foarte influentă a sistemului Unix.

De exemplu, încercați următoarele

```
$ cd; mkfifo mypipe
$ echo "hello" >mypipe & # put into background
[1] 8022
$ ls -l mypipe
prw-rw-r-- 1 penguin penguin 0 Oct 16 21:49 mypipe
$ cat mypipe
hello
[1]+  Done                  echo "hello" >mypipe
$ ls mypipe
mypipe
$ rm mypipe
```

1.2.9 Socluri

Soclrile sunt utilizate pe scară largă de toate comunicațiile pe Internet, bazele de date și sistemul de operare în sine. Este similar cu conducta cu nume (FIFO) și permite proceselor să schimbe informații chiar și între calculatoare diferite. Pentru soclu, aceste procese nu trebuie să ruleze în același timp și nici să ruleze ca procese secundare ale aceluiași proces ancestral. Acesta este punctul final pentru [comunicarea între procese \(IPC\)](#). Schimbul de informații poate avea loc prin rețea între diferite gazde. Cele mai comune două sunt [soclul Internet](#) și [soclul de domeniu Unix](#).

Indicație

«netstat -an» oferă o imagine de ansamblu foarte utilă asupra soclurilor deschise pe un anumit sistem.

1.2.10 Fișiere de dispozitive

Fișierele de dispozitive se referă la dispozitivele fizice sau virtuale din sistemul dvs., cum ar fi discul dur, placa video, ecranul sau tastatura. Un exemplu de dispozitiv virtual este consola, reprezentată de „/dev/console”.

Există 2 tipuri de fișiere de dispozitiv.

- **Dispozitiv de caractere**

- accesat de câte un caracter pe rând
- 1 caracter = 1 octet
- De exemplu, tastatura, portul serial, ...

- **Dispozitiv de blocuri**

- accesat în unități mai mari numite blocuri
- 1 bloc > 1 octet
- De exemplu, discul dur, ...

Puteți citi și scrie fișiere de dispozitiv, deși fișierul poate conține date binare care pot fi incompreensibile pentru oameni. Scrierea datelor direct în aceste fișiere este uneori utilă pentru depanarea conexiunilor hardware. De exemplu, puteți descărca un fișier text pe dispozitivul de imprimare „/dev/lp0” sau puteți trimite comenzi modem către portul serial corespunzător „/dev/ttyS0”. Dar, dacă nu se face cu atenție, acest lucru poate provoca o catastrofă majoră. Așadar, fiți precauți.

Notă

Pentru accesul normal la o imprimantă, utilizați lp(1).

Numărul nodului dispozitivului este afișat prin executarea comenzii ls(1), după cum urmează.

```
$ ls -l /dev/sda /dev/sr0 /dev/ttyS0 /dev/zero
brw-rw---T 1 root disk      8,  0 Oct 16 20:57 /dev/sda
brw-rw---T+ 1 root cdrom    11,  0 Oct 16 21:53 /dev/sr0
crw-rw---T 1 root dialout   4, 64 Oct 16 20:57 /dev/ttyS0
crw-rw-rw- 1 root root      1,  5 Oct 16 20:57 /dev/zero
```

- „/dev/sda” are numărul major al dispozitivului 8 și numărul minor al dispozitivului 0. Acesta este accesibil în mod citire/scriere pentru utilizatorii care aparțin grupului disk.
- „/dev/sr0” are numărul major al dispozitivului 11 și numărul minor al dispozitivului 0. Acesta este accesibil în mod citire/scriere pentru utilizatorii care aparțin grupului cdrom.
- „/dev/ttyS0” are numărul major al dispozitivului 4 și numărul minor al dispozitivului 64. Acesta este accesibil în mod citire/scriere pentru utilizatorii care aparțin grupului dialout.
- „/dev/zero” are numărul major al dispozitivului 1 și numărul minor al dispozitivului 5. Acesta poate fi citit/scriș de către oricine.

În sistemul Linux modern, sistemul de fișiere din „/dev/” este completat automat de mecanismul udev(7).

1.2.11 Fișiere de dispozitive speciale

Există câteva fișiere speciale pentru dispozitive.

Acestea sunt utilizate frecvent împreună cu redirectionarea shell-ului (a se vedea Secțiune [1.5.8](#)).

fișierul de dispozitiv	acțiunea	descrierea răspunsului
/dev/null	citire	returnează „caracterul de sfârșit de fișier (EOF)”
/dev/null	scriere	nu returnează nimic (o groapă fără fund pentru stocarea datelor)
/dev/zero	citire	returnează „caracterul \0 (NUL)” (nu este același cu numărul zero ASCII)
/dev/random	citire	returnează caractere aleatorii dintr-un generator de numere aleatorii reale, oferind entropie reală (lent)
/dev/urandom	citire	returnează caractere aleatorii dintr-un generator de numere pseudoaleatorii securizat criptografic
/dev/full	scriere	returnează eroarea de disc plin (ENOSPC)

Tabela 1.10: Lista fișierelor speciale de dispozitive

1.2.12 procfs și sysfs

[procfs](#) și [sysfs](#) montate pe „/proc” și „/sys” sunt pseudo-sisteme de fișiere și expun structurile de date interne ale nucleului către spațiul utilizatorului. Cu alte cuvinte, aceste intrări sunt virtuale, ceea ce înseamnă că acționează ca o fereastră convenabilă către funcționarea sistemului de operare.

Directorul „/proc” conține (printre altele) un subdirector pentru fiecare proces care rulează în sistem, numit după ID-ul procesului (PID). Utilitarele de sistem care accesează informații despre procese, cum ar fi `ps(1)`, obțin informațiile din această structură de directoare.

Directoarele din „/proc/sys/” conțin interfețe pentru modificarea anumitor parametri ai nucleului în timpul rulării; (puteți face același lucru prin comanda specializată `sysctl(8)` sau prin fișierul său de preîncărcare/configurare „/etc/sysctl.conf”).

Oamenii intră adesea în panică când observă un anumit fișier - „/proc/kcore” - care este, în general, foarte mare. Acesta este (mai mult sau mai puțin) o copie a conținutului memoriei calculatorului dvs. Este utilizat pentru depanarea nucleului. Este un fișier virtual care indică memoria calculatorului, așa că nu vă faceți griji în privința dimensiunii sale.

Directoarele din „/sys” conțin structuri de date exportate ale nucleului, atributele acestora și legăturile dintre ele. De asemenea, conține interfețe pentru modificarea anumitor parametri ai nucleului în timpul rulării.

A se vedea „`proc.txt(.gz)`”, „`sysfs.txt(.gz)`” și alte documente conexe din documentația nucleului Linux („/usr/share/doc/linux-doc-*/Documentation/filesystems/*”) furnizate de pachetul `linux-doc-*`.

1.2.13 tmpfs

[tmpfs](#) este un sistem de fișiere temporar care păstrează toate fișierele în [memoria virtuală](#). Datele din `tmpfs` din [cache-ul paginii](#) din memorie pot fi transferate în [spațiul de interschimb \(swap\)](#) de pe disc, după cum este necesar.

Directorul „/run” este montat ca `tmpfs` în procesul de pornire inițială. Acest lucru permite scrierea în el chiar și atunci când directorul „/” este montat ca fiind numai pentru citire. Aceasta este noua locație pentru stocarea fișierelor de stare tranzitorie și înlocuiește mai multe locații descrise în [Filesystem Hierarchy Standard](#) versiunea 2.3:

- „/var/run” → „/run”
- „/var/lock” → „/run/lock”
- „/dev/shm” → „/run/shm”

A se vedea „`tmpfs.txt(.gz)`” în documentația nucleului Linux („/usr/share/doc/linux-doc-*/Documentation/f”) furnizată de pachetul `linux-doc-*`.

1.3 Midnight Commander (MC)

Midnight Commander (MC) este un „briceag elvețian” GNU pentru consola Linux și alte medii terminale. Acesta oferă începătorilor o experiență de consolă bazată pe meniuri, mult mai ușor de învățat decât comenzile standard Unix.

Este posibil să fie necesar să instalați pachetul Midnight Commander, denumit „mc”, urmând pașii de mai jos.

```
$ sudo apt-get install mc
```

Utilizați comanda `mc(1)` pentru a explora sistemul Debian. Acesta este cel mai bun mod de a învăța. Explorați câteva locații interesante folosind tastele săgeții și tasta Enter.

- „/etc” și subdirectoarele acestuia
- „/var/log” și subdirectoarele acestuia
- „/usr/share/doc” și subdirectoarele acestuia
- „/usr/sbin” și „/usr/bin”

1.3.1 Personalizarea MC

Pentru a face ca MC să schimbe directorul de lucru la ieșire și după aceea a executacd către director, sugerez să modificați „~/ .bashrc” pentru a include un script furnizat de pachetul mc.

```
. /usr/lib/mc/mc.sh
```

Vedeți `mc(1)` (sub opțiunea „-P”) pentru a afla motivul. Dacă nu înțelegeți exact despre ce vorbesc aici, puteți face acest lucru mai târziu.

1.3.2 Lansarea MC

MC poate fi lansat astfel.

```
$ mc
```

MC se ocupă de toate operațiile cu fișiere prin intermediul meniului său, necesitând un efort minim din partea utilizatorului. Apăsăți tasta F1 pentru a afișa ecranul de ajutor. Puteți utiliza MC apăsând tastele cursorului și tastele de funcții.

Notă

În unele console, cum ar fi `gnome-terminal(1)`, apăsările tastelor de funcții pot fi interceptate de programul consolei. Puteți dezactiva aceste funcții în meniul „Preferințe” → „General” și „Comenzi rapide” pentru `gnome-terminal`.

Dacă întâmpinați probleme de codificare a caracterelor care afișează caractere neinteligibile, adăugarea „-a” la linia de comandă MC poate ajuta la prevenirea problemelor.

Dacă acest lucru nu rezolvă problemele de afișare cu MC, consultați Secțiune [9.5.6](#).

1.3.3 Gestionarul de fișiere din MC

Implicit sunt două panouri de directoare care conțin liste de fișiere. Un alt mod util este configurarea ferestrei din dreapta pe „informații” pentru a vedea informații despre privilegiile de acces la fișiere etc. În continuare sunt prezentate câteva comenzi esențiale. Cu demonul `gpm(8)` în funcțiune, se poate utiliza mouse-ul și pe consolele de caractere Linux. (Asigurați-vă că apăsați tasta Shift pentru a obține comportamentul normal al funcțiilor de tăiere și lipire în MC.)

tasta	comanda/funcția asociată
F1	meniul de ajutor
F3	vizorul de fișiere intern
F4	editorul intern
F9	activează meniul derulant
F10	ieșire din Midnight Commander
Tab	deplasare între două ferestre
Insert sau Ctrl-T	marchează fișierul pentru o operație cu mai multe fișiere, cum ar fi copierea
Del	șterge fișierul (aveți grijă --- configurați MC în modul de ștergere sigură)
Tastele cursor	auto-explicative

Tabela 1.11: Tastele de comenzi rapide ale MC

1.3.4 Trucuri din linia de comandă în MC

- Comanda `cd` modifică directorul afișat pe ecranul selectat.
- `Ctrl-Enter` sau `Alt-Enter` copiază un nume de fișier în linia de comandă. Utilizați această comandă împreună cu comenzile `cp(1)` și `mv(1)` și cu editarea liniei de comandă.
- `Alt-Tab` afișează opțiunile de extindere a numelor de fișiere din shell.
- Se poate specifica directorul de pornire pentru ambele ferestre ca argumente pentru MC; de exemplu, „`mc /etc /root`”.
- `Esc + n-key` → `Fn` (adică, `Esc + 1` → `F1`, etc.; `Esc + 0` → `F10`)
- Apăsarea tastei `Esc` înaintea tastei are același efect ca și apăsarea simultană a tastelor `Alt` și ; Adică, tastați `Esc + c` pentru `Alt-C`. `Esc` se numește meta-tastă și uneori este notată ca „`M-`”.

1.3.5 Editorul intern din MC

Editorul intern are o schemă interesantă de copiere și lipire. Apăsarea tastei `F3` marchează începutul unei selecții, o a doua apăsare a tastei `F3` marchează sfârșitul selecției și evidențiază selecția. Apoi puteți muta cursorul. Dacă apăsați tasta «`F6`», zona selectată este mutată la locația cursorului. Dacă apăsați tasta «`F5`», zona selectată este copiată și inserată la locația cursorului. `F2` salvează fișierul. `F10` vă scoate din program. Majoritatea tastelor cursorului funcționează intuitiv.

Acest editor poate fi pornit direct pe un fișier folosind una dintre următoarele comenzi.

```
$ mc -e filename_to_edit
```

```
$ mcedit filename_to_edit
```

Acesta nu este un editor cu ferestre multiple, dar se pot utiliza mai multe console Linux pentru a obține același efect. Pentru a copia între ferestre, utilizați tastele `Alt-Fn` pentru a comuta între consolele virtuale și utilizați „`File → Insert file`” (Fișier → Inserați fișier) sau „`File → Copy to file`” (Fișier → Copiați în fișier) pentru a muta o parte dintr-un fișier într-un alt fișier.

Acest editor intern poate fi înlocuit cu orice editor extern la alegere.

De asemenea, multe programe utilizează variabilele de mediu „`$EDITOR`” sau „`$VISUAL`” pentru a decide ce editor să utilizeze. Dacă nu vă simțiți confortabil cu `vim(1)` sau `nano(1)` la început, puteți defini aceste variabile ca „`mcedit`” adăugând următoarele linii în „`~/ .bashrc`”.

```
export EDITOR=mcedit
export VISUAL=mcedit
```

Recomand să le definiți ca „vim”, dacă este posibil.

Dacă nu vă simțiți confortabil cu vim(1), puteți continua să utilizați mcedit(1) pentru majoritatea sarcinilor de întreținere a sistemului.

1.3.6 Vizorul intern din MC

MC este un vizor foarte inteligent. Este un instrument excelent pentru căutarea cuvintelor în documente. Îl folosesc întotdeauna pentru fișierele din directorul „/usr/share/doc”. Este cea mai rapidă modalitate de a răsfoi cantități mari de informații despre Linux. Acest vizor poate fi pornit direct folosind una dintre următoarele comenzi.

```
$ mc -v path/to/filename_to_view
```

```
$ mcview path/to/filename_to_view
```

1.3.7 Funcții de pornire automată ale MC

Apăsati tasta «Enter» pe un fișier, iar programul corespunzător va gestiona conținutul fișierului (consultați Secțiune 9.4.11). Aceasta este o funcție MC foarte convenabilă.

tipul fișierului	reacția la apăsarea tastei «Enter»
fișier executabil	execută comanda
fișier de pagină de manual	directionează conținutul către software-ul de vizualizare
fișier html	directionează conținutul către navigatorul web
fișiere „*.tar.gz” și „*.deb”	răsfoiește conținutul său ca și cum ar fi un subdirector

Tabela 1.12: Reacția la apăsarea tastei «Enter» în MC

Pentru ca aceste funcții de vizualizare și fișiere virtuale să funcționeze, fișierele care pot fi vizualizate nu trebuie definite ca fiind executabile. Modificați-le starea folosind chmod(1) sau prin meniul fișier MC.

1.3.8 Sistemul de fișiere virtual al MC

MC poate fi utilizat pentru a accesa fișiere prin Internet. Accesați meniul apăsând F9, „Enter” și „h” pentru a activa sistemul de fișiere Shell. Introduceți o adresă URL în formatul „sh://[[utilizator@]mașină[:opțiuni] / [director - l care recuperează un director la distanță care apare ca unul local utilizând ssh.

1.4 Mediul de lucru de bază de tip Unix

Deși MC vă permite să faceți aproape totul, este foarte important să învățați cum să utilizați instrumentele liniei de comandă invocate din promptul shell și să vă familiarizați cu mediul de lucru de tip Unix.

1.4.1 Shell-ul de autentificare

Deoarece shell-ul de autentificare poate fi utilizat de unele programe de inițializare a sistemului, este prudent să îl păstrați ca bash(1) și să evitați schimbarea shell-ului de autentificare cu chsh(1).

Dacă doriți să utilizați un prompt interactiv diferit, definiți-l din configurația emulatorului de terminal cu interfață grafică sau porniți-l din ~/.bashrc, de exemplu, introducând „exec /usr/bin/zsh -i -l” sau „exec /usr/bin/fish -i -l”.

pachet	popcon(popularitate)	limbaj	shell POSIX	descriere
bash	V:875, I:999	7273	Da	Bash : GNU Bourne Again SHell (standard de facto)
bash-completion	V:33, I:951	1952	N/D	completare programabilă pentru shell-ul bash
dash	V:910, I:998	207	Da	Debian Almquist Shell , bun pentru scripturi shell
zsh	V:41, I:71	2509	Da	Z shell : shell-ul standard cu numeroase îmbunătățiri
tcsh	V:3, I:15	1366	Nu	TENEX C Shell : o versiune îmbunătățită a Berkeley csh
mksh	V:2, I:8	7713	Da	O versiune a Korn shell
csh	V:1, I:5	348	Nu	OpenBSD C Shell , o versiune a Berkeley csh
sash	V:0, I:5	1335	Da	Shell autonom cu comenzi încorporate (nu este destinat pentru standardul „/usr/bin/sh”)
ksh	I:8	65	Da	Versiunea reală, AT&T a Korn shell
rc	V:0, I:0	182	Nu	Implementarea AT&T Plan 9 rc shell
posh	V:0, I:0	187	Da	Shell obișnuit conform cu politica (derivat din pdksh)

Tabela 1.13: Lista programelor shell

Indicație

Deși shell-urile de tip POSIX au aceeași sintaxă de bază, ele pot diferi în ceea ce privește comportamentul în aspecte fundamentale, precum variabilele shell și expansiunile globale (cu caractere joker). Vă rugăm să consultați documentația acestora pentru detalii.

În acest capitol al tutorialului, shell-ul interactiv înseamnă întotdeauna bash.

1.4.2 Personalizarea bash

Puteți personaliza comportamentul bash(1) prin „~/ .bashrc”.

De exemplu, încercați următoarele.

```
# enable bash-completion
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

# CD upon exiting MC
. /usr/lib/mc/mc.sh

# set CDPATH to a good one
CDPATH=./usr/share/doc::~~/Desktop::~~
export CDPATH

PATH="${PATH+$PATH:}/usr/sbin:/sbin"
# set PATH so it includes user's private bin if it exists
if [ -d ~/bin ] ; then
  PATH="~/bin${PATH+:$PATH}"
```

```
fi
export PATH

EDITOR=vim
export EDITOR
```

Indicație

Puteți găsi mai multe sfaturi de personalizare bash, cum ar fi Secțiune [9.3.6](#), în Cap. [9](#).

Indicație

Pachetul `bash-completion` permite completarea programabilă pentru bash.

1.4.3 Combinații speciale de taste

În mediul [de tip Unix](#) există câteva combinații de taste care au semnificații speciale. Rețineți că pe o consolă Linux normală, numai tastele `Ctrl` și `Alt` din partea stângă funcționează așa cum vă așteptați. Iată câteva combinații de taste importante de reținut.

tasta	descrierea acțiunii efectuate
<code>Ctrl-U</code>	șterge linia din fața cursorului
<code>Ctrl-H</code>	șterge un caracter înaintea cursorului
<code>Ctrl-D</code>	termină introducerea („ieșire din shell” dacă utilizați shell)
<code>Ctrl-C</code>	termină un program în execuție
<code>Ctrl-Z</code>	oprește temporar programul mutându-l în fundal
<code>Ctrl-S</code>	oprește afișarea pe ecran
<code>Ctrl-Q</code>	reactivează ieșirea pe ecran
<code>Ctrl-Alt-Del</code>	repornește/oprește sistemul, a se vedea <code>init tab(5)</code>
Tasta <code>Alt</code> stânga (opțional, Tasta <code>Windows</code>)	meta-tastă pentru Emacs și interfețe similare
Săgeată-sus	pornește căutarea istoricului comenzilor sub bash
<code>Ctrl-R</code>	pornește căutarea incrementală în istoricul comenzilor sub bash
<code>Tab</code>	completează introducerea numelui fișierului în linia de comandă sub bash
<code>Ctrl-V Tab</code>	introduce <code>Tab</code> fără extindere în linia de comandă sub bash

Tabela 1.14: Lista tastelor de comenzi rapide pentru bash

Indicație

Funcția terminalului `Ctrl-S` poate fi dezactivată folosind `stty(1)`.

1.4.4 Operații cu mouse-ul

[Operațiile cu mouse-ul pentru text în sistemul Debian combină două stiluri](#) cu câteva modificări:

- Operații tradiționale ale mouse-ului în stil Unix:
 - utilizează 3 butoane (clic)

- utilizare PRIMARĂ
- utilizată de aplicații X precum `xterm` și aplicații text în consola Linux
- Operații moderne cu mouse-ul în stil de interfață grafică:
 - utilizează 2 butoane (glisare + clic)
 - utilizare PRIMARĂ și CLIPBOARD
 - utilizată în aplicații cu interfață grafică modernă, cum ar fi `gnome-terminal`

acțiunea	răspunsul
Clic stânga și glisare cu mouse-ul	selectează intervalul ca selecție PRIMARĂ
Clic stânga	selectează începutul intervalului pentru selecția PRIMARĂ
Clic dreapta (tradițional)	selectează sfârșitul intervalului pentru selecția PRIMARĂ
Clic dreapta (modern)	menu dependent de context (tăiere/copiere/lipire)
Clic pe butonul din mijloc sau Shift-Ins	inserează selecția PRIMARĂ la poziția cursorului
Ctrl-X	taie selecția PRIMARĂ în CLIPBOARD
Ctrl-C (Shift-Ctrl-C în terminal)	copiază selecția PRIMARĂ în CLIPBOARD
Ctrl-V	lipește conținutul CLIPBOARDULUI la poziția cursorului

Tabela 1.15: Lista operațiilor mouse-ului și acțiunile tastelor asociate în Debian

Aici, selecția PRIMARĂ este intervalul de text evidențiat. În cadrul programului de terminal, se utilizează în schimb Shift-Ctrl-C pentru a evita închiderea unui program în execuție.

Rotița centrală a mouse-ului modern cu rotiță este considerată butonul din mijloc al mouse-ului și poate fi utilizată pentru clic-mijloc. Apăsarea simultană a butoanelor stânga și dreapta ale mouse-ului servește ca clic-mijloc în cazul unui mouse cu 2 butoane.

Pentru a utiliza un mouse în consolele de caractere Linux, trebuie să aveți `gpm(8)` rulând ca demon.

1.4.5 Paginatorul

Comanda `less(1)` este un paginator îmbunătățit (permite navigarea prin conținutul fișierelor). Citește fișierul specificat prin argumentul comenzii sau prin intrarea standard. Apăsați „h” dacă aveți nevoie de ajutor în timp ce navigați cu comanda `less`. Poate face mult mai mult decât `more(1)` și poate fi supraalimentat prin executarea „eval \$(lesspipe)” sau „eval \$(lessfile)” în scriptul de pornire al shell-ului. Pentru mai multe informații, consultați „/usr/share/doc/less/LESSOPEN”. Opțiunea „-R” permite afișarea caracterelor brute și activează secvențele de eludare a culorilor ANSI. Consultați `less(1)`.

Indicație

În comanda `less`, tastați „h” pentru a afișa ecranul de ajutor, tastați „/” sau „?” pentru a căuta un șir de caractere și tastați „-i” pentru a schimba respectarea/ignorarea diferenței dintre majuscule și minuscule.

1.4.6 Editorul de text

Ar trebui să deveniți expert într-una dintre variantele programelor [Vim](#) sau [Emacs](#), care sunt populare în sistemele de tip Unix.

Cred că este bine să vă obișnuiți cu comenzile Vim, deoarece editorul Vi este mereu prezent în lumea Linux/Unix; (de fapt, programele originale `vi` sau noul `nvi` se găsesc peste tot. Am ales Vim pentru începători, deoarece oferă ajutor prin tasta F1, fiind suficient de similar și mai avansat).

Dacă ați ales [Emacs](#) sau [XEmacs](#) ca editor, aceasta este o altă alegere bună, în special pentru programare. Emacs are o mulțime de alte caracteristici, inclusiv funcționarea ca cititor de știri, editor de directoare, program de poștă

electronică etc. Când este utilizat pentru programare sau editarea scripturilor shell, recunoaște în mod inteligent formatul asupra căruia lucrați și încearcă să vă ofere asistență. Unii oameni susțin că singurul program de care au nevoie pe Linux este Emacs. Zece minute de învățare a Emacs acum pot economisi ore mai târziu. Este foarte recomandat să aveți manualul GNU Emacs ca referință atunci când învățați Emacs.

Toate aceste programe vin de obicei cu un program de tutoriat pentru a le învăța prin practică. Porniți Vim tastând „vim” și apăsați tasta F1. Ar trebui să citiți cel puțin primele 35 de linii. Apoi urmați cursul de formare pe internet mutând cursorul la „| tutor |” și apăsând Ctrl-].

Notă

Editorii buni, precum Vim și Emacs, pot gestiona corect textele codificate în UTF-8 și alte codificări exotice. Este o idee bună să utilizați mediul de interfață grafică cu configurația regională pentru text, UTF-8 și să instalați programele și fonturile necesare în acesta. Editorii au opțiuni pentru a configura codificarea fișierelor independent de mediul de interfață grafică. Vă rugăm să consultați documentația acestora referitoare la textul multi-octet.

1.4.7 Definirea unui editor de text implicit

Debian vine cu o serie de editori diferiți. Vă recomandăm să instalați pachetul vim, așa cum s-a menționat mai sus.

Debian oferă acces unificat la editorul implicit al sistemului prin comanda „usr/bin/editor”, astfel încât alte programe (de exemplu, reportbug(1)) să îl poată invoca. Îl puteți modifica după cum urmează.

```
$ sudo update-alternatives --config editor
```

Alegerea „usr/bin/vim.basic” în locul „usr/bin/vim.tiny” este recomandarea mea pentru începători, deoarece oferă suport pentru evidențierea sintaxei.

Indicație

Multe programe utilizează variabilele de mediu „\$EDITOR” sau „\$VISUAL” pentru a decide ce editor să utilizeze (consultați Secțiune 1.3.5 și Secțiune 9.4.11). Pentru consecvență în sistemul Debian, definiți aceste variabile la „usr/bin/editor”; (în trecut, „\$EDITOR” era „ed” și „\$VISUAL” era „vi”).

1.4.8 Utilizarea editorului vim

Versiunea recentă vim(1) pornește în mod implicit cu opțiunea „nocompatible” și intră în modul NORMAL.¹

Vă rugăm să utilizați programul „vimtutor” pentru a învăța vim printr-un curs tutorial interactiv.

Programul vim își modifică comportamentul în funcție de tastele apăstate, pe baza **modului**. Tastarea în tampon se face în principal în modul INSERARE și în modul ÎNLOCUIRE. Deplasarea cursorului se face în principal în modul NORMAL. Selecția interactivă se face în modul VIZUAL. Tastarea „:” în modul NORMAL schimbă **modul** în modul Ex. Modul Ex acceptă comenzi.

Indicație

Vim vine cu pachetul **Netrw**. Netrw permite citirea și scrierea fișierelor, navigarea în directoare prin rețea și navigarea locală! Încercați Netrw cu „vim .” (un punct ca argument) și citiți manualul său la „:help netrw”.

Pentru configurarea avansată a vim, consultați Secțiune 9.2.

¹Chiar și versiunea mai veche vim poate porni în modul „nocompatible” prin opțiunea „-N”.

modul	combinația de taste	acțiunea
NORMAL	:help only	afișează fișierul de ajutor
NORMAL	:e filename.ext	deschide un nou spațiu tampon în memorie pentru a edita filename.ext
NORMAL	:w	suprascrie fișierul original cu conținutul tamponului curent
NORMAL	:w filename.ext	scrie tamponul curent în filename.ext
NORMAL	:q	ieșire din vim
NORMAL	:q!	forțează ieșirea din vim
NORMAL	:only	închide toate celelalte ferestre deschise
NORMAL	:set nocompatible?	verifică dacă vim se află în modul nocompatible
NORMAL	:set nocompatible	stabilește vim în modul de operare nocompatible
NORMAL	i	intră în modul INSERARE
NORMAL	R	intră în modul ÎNLOCUIRE
NORMAL	v	intră în modul VIZUAL
NORMAL	V	intră în modul VIZUAL pe linii
NORMAL	Ctrl-V	intrați în modul VISUAL pe blocuri
cu excepția TERMINAL - JOB	ESC-key	intră în modul NORMAL
NORMAL	:term	intră în modul TERMINAL - JOB
TERMINAL - NORMAL	i	intră în modul TERMINAL - JOB
TERMINAL - JOB	Ctrl-W N (sau Ctrl-\ Ctrl-N)	intră în modul TERMINAL - NORMAL
TERMINAL - JOB	Ctrl-W :	intră în modul Ex din modul TERMINAL - NORMAL

Tabela 1.16: Lista combinațiilor de taste de bază din Vim

1.4.9 Înregistrarea activităților shell-ului

Rezultatul comenzii shell poate să nu mai apară pe ecran și să se piardă definitiv. Este recomandat să înregistrați activitățile shell în fișier pentru a le putea revizui ulterior. Acest tip de înregistrare este esențial atunci când efectuați orice sarcini de administrare a sistemului.

Indicație

Noul Vim (versiunea ≥ 8.2) poate fi utilizat pentru a înregistra activitățile shell-ului în mod curat folosind modul TERMINAL - JOB. A se vedea Secțiune [1.4.8](#).

Metoda de bază pentru înregistrarea activității shell-ului este rularea acestuia sub script(1).

De exemplu, încercați următoarele

```
$ script
Script started, file is typescript
```

Executați orice comenzi shell sub script.

Apăsați Ctrl-D pentru a ieși din script.

```
$ vim typescript
```

Consultați Secțiune [9.1.1](#).

1.4.10 Comenzi Unix de bază

Să învățăm comenzile de bază ale Unix. Aici folosesc „Unix” în sensul său generic. Orice sistem de operare clonă Unix oferă de obicei comenzi echivalente. Sistemul Debian nu face excepție. Nu vă faceți griji dacă unele comenzi

nu funcționează așa cum doriți acum. Dacă `alias` este utilizat în shell, ieșirile corespunzătoare ale comenzii sunt diferite. Aceste exemple nu sunt menite să fie executate în această ordine.

Încercați toate comenzile următoare din contul de utilizator fără privilegii.

Notă

Unix are tradiția de a ascunde numele fișierelor care încep cu „.”. Acestea sunt, de obicei, fișiere care conțin informații de configurare și preferințele utilizatorului.

Pentru comanda `cd`, consultați `builtins(7)`.

Paginatorul implicit al sistemului Debian de bază este `more(1)`, care nu permite derularea înapoi. Prin instalarea pachetului `less` folosind linia de comandă „`apt-get install less`”, `less(1)` devine paginatorul implicit și puteți derula înapoi cu tastele cursorului.

„[” și „]” din expresia regulată a comenzii „`ps aux | grep -e „[e]xim4*”`” de mai sus permit `grep` să evite potrivirea cu sine însuși. „4*” din expresia regulată înseamnă 0 sau mai multe repetări ale caracterului „4”, permițând astfel lui `grep` să potrivească atât cu „`exim`”, cât și cu „`exim4`”. Deși „*” este utilizat în globalizarea numelui de fișier shell și în expresia regulată, semnificațiile lor sunt diferite. Aflați mai multe despre expresia regulată din `grep(1)`.

Vă rugăm să parcurgeți directoarele și să aruncați o privire în sistem folosind comenzile de mai sus ca exercițiu. Dacă aveți întrebări cu privire la oricare dintre comenzile consolei, vă rugăm să citiți pagina manualului.

De exemplu, încercați următoarele

```
$ man man
$ man bash
$ man builtins
$ man grep
$ man ls
```

Stilul paginilor de manual poate fi puțin dificil de asimilat, deoarece acestea sunt destul de concise, în special cele mai vechi, foarte tradiționale. Dar, odată ce vă obișnuiți cu ele, veți aprecia concisitatea lor.

Rețineți că multe comenzi de tip Unix, inclusiv cele din GNU și BSD, afișează informații succinte de ajutor dacă le invocați într-unul din următoarele moduri (sau fără argumente, în unele cazuri).

```
$ commandname --help
$ commandname -h
```

1.5 Comanda simplă de shell

Acum aveți o idee despre cum se utilizează sistemul Debian. Să analizăm în detaliu mecanismul de execuție a comenzilor în sistemul Debian. Aici, am simplificat realitatea pentru începători. Consultați `bash(1)` pentru explicații exacte.

O comandă simplă este o secvență de componente.

1. Atribuire de variabile (opțional)
 2. Numele comenzii
 3. Argumente (opțional)
 4. Redirecționări (opțional: `>`, `>>`, `<`, `<<`, etc.)
 5. Operator de control (opțional: `&&`, `||`, `newline`, `;`, `&`, `()`)
-

comanda	descriere
<code>pwd</code>	afișează numele directorului curent/de lucru
<code>whoami</code>	afișează numele utilizatorului curent
<code>id</code>	afișează identitatea utilizatorului curent (nume, uid, gid și grupurile asociate)
<code>file foo</code>	afișează tipul de fișier pentru fișierul „ <i>foo</i> ”
<code>type -p nume-comandă</code>	afișează locația fișierului comenzii „ <i>nume-comandă</i> ”
<code>which nume-comandă</code>	, ,
<code>type nume-comandă</code>	afișează informații despre comanda „ <i>nume-comandă</i> ”
<code>apropos cuvânt-cheie</code>	găsește comenzi legate de „ <i>cuvânt-cheie</i> ”
<code>man -k cuvânt-cheie</code>	, ,
<code>whatis nume-comandă</code>	afișează o explicație pe o singură linie privind comanda „ <i>nume-comandă</i> ”
<code>man -a nume-comandă</code>	afișează explicația privind comanda „ <i>nume-comandă</i> ” (stil Unix)
<code>info nume-comandă</code>	afișează o explicație destul de lungă privind comanda „ <i>nume-comandă</i> ” (stil GNU)
<code>ls</code>	listează conținutul directorului (fișiere și directoare fără punct)
<code>ls -a</code>	listează conținutul directorului (toate fișierele și directoarele)
<code>ls -A</code>	listează conținutul directorului (aproape toate fișierele și directoarele, adică omite „.” și „.”)
<code>ls -la</code>	listează tot conținutul directorului cu informații detaliate
<code>ls -lai</code>	listează tot conținutul directorului cu numărul de nod-i și informații detaliate
<code>ls -d</code>	listează toate directoarele din directorul curent
<code>tree</code>	afișează conținutul arborelui de fișiere
<code>ls -l foo</code>	afișează starea deschisă a fișierului „ <i>foo</i> ”
<code>ls -l -p pid</code>	listează fișierele deschise de ID-ul procesului: „ <i>pid</i> ”
<code>mkdir foo</code>	crează un nou director „ <i>foo</i> ” în directorul curent
<code>rmdir foo</code>	elimină directorul „ <i>foo</i> ” din directorul curent
<code>cd foo</code>	schimbă directorul în directorul „ <i>foo</i> ” din directorul curent sau din directorul listat în variabila „ <i>\$CDPATH</i> ”
<code>cd /</code>	schimbă directorul la directorul rădăcină
<code>cd</code>	schimbă directorul la directorul personal al utilizatorului curent
<code>cd /foo</code>	schimbă directorul la directorul cu ruta absolută „ <i>/foo</i> ”
<code>cd ..</code>	schimbă directorul la directorul părinte
<code>cd ~foo</code>	schimbă directorul la directorul personal al utilizatorului „ <i>foo</i> ”
<code>cd -</code>	schimbă directorul la directorul anterior
<code></etc/motd pager</code>	afișează conținutul fișierului „ <i>/etc/motd</i> ” folosind paginatorul implicit
<code>touch fișier-test</code>	creați un fișier gol „ <i>fișier-test</i> ”
<code>cp foo bar</code>	copiază un fișier existent „ <i>foo</i> ” într-un fișier nou „ <i>bar</i> ”
<code>rm fișier-test</code>	elimină fișierul „ <i>fișier-test</i> ”
<code>mv foo bar</code>	redenumeste un fișier existent „ <i>foo</i> ” într-un nou nume „ <i>bar</i> ” („ <i>bar</i> ” nu trebuie să existe)
<code>mv foo bar</code>	mută un fișier existent „ <i>foo</i> ” într-o nouă locație „ <i>bar/foo</i> ” (directorul „ <i>bar</i> ” trebuie să existe)
<code>mv foo bar/baz</code>	mută un fișier existent „ <i>foo</i> ” într-o nouă locație cu un nou nume „ <i>bar/baz</i> ” (directorul „ <i>bar</i> ” trebuie să existe, dar directorul „ <i>bar/baz</i> ” nu trebuie să existe)
<code>chmod 600 foo</code>	face ca un fișier existent „ <i>foo</i> ” să fie necitibil și neinscriptibil de către alte persoane (neexecutabil pentru toți)
<code>chmod 644 foo</code>	face ca un fișier existent „ <i>foo</i> ” să fie citibil, dar neinscriptibil de către alte persoane (neexecutabil pentru toți)
<code>chmod 755 foo</code>	faceți ca un fișier existent „ <i>foo</i> ” să fie citibil, dar neinscriptibil de către alte persoane (executabil pentru toți)
<code>find . -name model</code>	găsește numele de fișiere ce coincid folosind shell-ul „ <i>model</i> ” (mai lent)
<code>locate -d . model</code>	găsește numele de fișiere ce coincid folosind shell-ul „ <i>model</i> ” (mai rapid folosind baza de date generată regulat)
<code>grep -e "model" *.html</code>	găsește un „ <i>model</i> ” în toate fișierele care se termină cu „ <i>.html</i> ” din directorul curent și le afișează pe toate
<code>top</code>	afișează informații despre proces pe ecran complet, tastează „q” pentru a ieși
<code>.</code>	afișează informații despre toate procesele care rulează utilizând

1.5.1 Executarea comenzilor și variabilele de mediu

Valorile unor [variabile de mediu](#) modifică comportamentul unor comenzi Unix.

Valorile implicite ale variabilelor de mediu sunt stabilite inițial de sistemul PAM, iar unele dintre ele pot fi reajustate de anumite programe aplicaționale.

- Sistemul PAM, cum ar fi `pam_env`, poate defini variabile de mediu prin `/etc/pam.conf`, `/etc/environment`, și `/etc/default/locale`.
- Administratorul de afișare, cum ar fi `gdm3`, poate redefini variabilele de mediu pentru sesiunea de interfață grafică prin `~/.profile`.
- Inițializarea programului specific utilizatorului poate redefini variabilele de mediu prin `~/.profile`, `~/.bash_profile` și `~/.bashrc`.

1.5.2 Variabila „\$LANG”

Configurația regională implicită este definită în variabila de mediu „\$LANG” și este configurată ca „LANG=xx_YY.UTF-8” de către programul de instalare sau de către configurația ulterioară a interfeței grafice, de exemplu, „Configurări” → „Regiune și limbă” → „Limbă” / „Formate” pentru GNOME.

Notă

Vă recomand să configurați mediul de sistem doar cu variabila „\$LANG” pentru moment și să evitați variabilele „\$LC_*”, cu excepția cazului în care este absolut necesar.

Valoarea completă a configurației regionale atribuită variabilei „\$LANG” este formată din 3 părți: „xx_YY.ZZZZ”.

valoarea configurației regionale	semnificație
xx	codurile de limbă ISO 639 (cu litere mici), cum ar fi „ro”
YY	codurile de țară ISO 3166 (majuscule), cum ar fi „RO”
ZZZZ	codificarea setului de caractere , definită întotdeauna la „UTF-8”

Tabela 1.18: Cele 3 părți ale valorii configurației regionale

Executarea tipică a comenzii utilizează o secvență de linii shell, după cum urmează.

```
$ echo $LANG
en_US.UTF-8
$ date -u
Wed 19 May 2021 03:18:43 PM UTC
$ LANG=fr_FR.UTF-8 date -u
mer. 19 mai 2021 15:19:02 UTC
```

Aici, programul `date(1)` este executat cu valori de configurație regională diferite.

- Pentru prima comandă, „\$LANG” este definită la valoarea implicită a sistemului pentru [configurația regională](#) „en_US.UTF-8”.
- Pentru a doua comandă, „\$LANG” este definită la valoarea [configurației regionale](#) franceze UTF-8 „fr_FR.UTF-8”.

Majoritatea execuțiilor de comenzi nu au, de obicei, o definiție a variabilei de mediu precedentă. Pentru exemplul de mai sus, puteți executa alternativ următoarele.

```
$ LANG=fr_FR.UTF-8
$ date -u
mer. 19 mai 2021 15:19:24 UTC
```


recomandare configurație regională	limba (zona teritorială)
en_US.UTF-8	engleză (SUA)
en_GB.UTF-8	engleză (Marea Britanie)
fr_FR.UTF-8	franceză (Franța)
de_DE.UTF-8	germană (Germania)
it_IT.UTF-8	italiană (Italia)
es_ES.UTF-8	spaniolă (Spania)
ca_ES.UTF-8	catalană (Spania)
sv_SE.UTF-8	suedeză (Suedia)
pt_BR.UTF-8	portugheză (Brazilia)
ru_RU.UTF-8	rusă (Rusia)
zh_CN.UTF-8	chineză (R.P. Chiina)
zh_TW.UTF-8	chineză (Taiwan)
ja_JP.UTF-8	japoneză (Japonia)
ko_KR.UTF-8	coreeană (Republica Coreea)
vi_VN.UTF-8	vietnameză (Vietnam)

Tabela 1.19: Lista recomandărilor privind configurația regională

Indicație

Când raportați o eroare, este recomandat să rulați și să verificați comanda în configurația regională „en_US . UTF - 8” dacă utilizați un mediu non-englez.

Pentru detalii precise privind configurarea parametrilor configurației regionale, consultați Secțiune [8.1](#).

1.5.3 Variabila „\$PATH”

Când introduceți o comandă în shell, shell-ul caută comanda în lista de directoare conținute în variabila de mediu „\$PATH”. Valoarea variabilei de mediu „\$PATH” este denumită și ruta de căutare a shell-ului.

În instalarea implicită Debian, variabila de mediu „\$PATH” a conturilor de utilizator poate să nu includă „/usr/sbin” și „/usr/bin”. De exemplu, comanda `ifconfig` trebuie emisă cu ruta completă „/usr/sbin/ifconfig”; (comanda similară `ip` se află în „/usr/bin”).

Puteți modifica variabila de mediu „\$PATH” a shell-ului Bash prin fișierele „~/ .bash_profile” sau „~/ .bashrc”.

1.5.4 Variabila „\$HOME”

Multe comenzi stochează configurația specifică utilizatorului în directorul personal al acestuia și modifică comportamentul acestora în funcție de conținutul lor. Directorul personal este identificat de variabila de mediu „\$HOME”.

valoarea variabilei „\$HOME”	situația execuției programului
/	programul este rulat de procesul init (demon)
/root	programul rulează din shell-ul root normal
/home/utilizator-normal	programul rulează din shell-ul utilizatorului normal
/home/utilizator-normal	programul rulează din meniul mediului grafic de birou al utilizatorului normal
/home/utilizator-normal	programul rulează ca root cu «sudo program»
/root	programul rulează ca root cu «sudo -H program»

Tabela 1.20: Lista valorilor variabilei „\$HOME”

Indicație

Shell-ul extinde „~/” la directorul personal al utilizatorului curent, adică „\$HOME/”. Shell-ul extinde „~foo/” la directorul personal al lui foo, adică „/home/foo/”.

Consultați Secțiune [12.1.5](#) dacă \$HOME nu este disponibil pentru programul dvs.

1.5.5 Opțiuni în linia de comandă

Unele comenzi acceptă argumente. Argumentele care încep cu „-” sau „--” sunt numite opțiuni și controlează comportamentul comenzii.

```
$ date
Thu 20 May 2021 01:08:08 AM JST
$ date -R
Thu, 20 May 2021 01:08:12 +0900
```

Aici, argumentul liniei de comandă „-R” modifică comportamentul date(1) pentru a genera un șir de caractere compatibil cu [RFC2822](#).

1.5.6 Facilitatea glob a shell-ului

Adesea, doriți ca o comandă să funcționeze cu un grup de fișiere fără a le introduce pe toate. Modelul de extindere a numelor de fișiere utilizând facilitatea **glob** a shell-ului (denumită uneori **wildcards**) facilitează această necesitate.

modele globale shell	descrierea regulii de potrivire
*	numele fișierului (segment) nu începe cu „.”
.*	numele fișierului (segment) începe cu „.”
?	exact un caracter
[...]	exact un caracter cu orice caracter între paranteze
[a-z]	exact un caracter cu orice caracter între „a” și „z”
[^...]	exact un caracter, altul decât orice caracter inclus între paranteze (cu excepția „^”)

Tabela 1.21: Modele globale shell

De exemplu, încercați următoarele

```
$ mkdir junk; cd junk; touch 1.txt 2.txt 3.c 4.h .5.txt ..6.txt
$ echo *.txt
1.txt 2.txt
$ echo *
1.txt 2.txt 3.c 4.h
$ echo *.hc
3.c 4.h
$ echo .*
. . . 5.txt ..6.txt
$ echo .*[^.]*
.5.txt ..6.txt
$ echo [^1-3]*
4.h
$ cd ../; rm -rf junk
```

A se vedea `glob(7)`.

Notă

Spre deosebire de extinderea normală a numelor de fișiere de către shell, modelul shell „*” testat în `find(1)` cu testul „-name” etc., se potrivește cu „.” inițial al numelui fișierului; (noua caracteristică [POSIX](#)).

Notă

BASH poate fi modificat pentru a-i schimba comportamentul glob cu opțiunile sale încorporate shopt, cum ar fi „dotglob”, „noglob”, „nocaseglob”, „nullglob”, „extglob” etc. Consultați `bash(1)`.

1.5.7 Valoarea returnată de comandă

Fiecare comandă returnează starea sa de ieșire (variabila: „\$?”) ca valoare de returnare.

starea de ieșire a comenzii	valoare numerică returnată	valoarea logică returnată
succes	zero, 0	TRUE
eroare	diferită de zero, -1	FALSE

Tabela 1.22: Coduri de ieșire ale comenzii

De exemplu, încercați următoarele.

```
$ [ 1 = 1 ] ; echo $?
0
$ [ 1 = 2 ] ; echo $?
1
```

Notă

Vă rugăm să rețineți că, în contextul logic pentru shell, **succesul** este tratat ca **TRUE** logic, care are valoarea 0 (zero). Acest lucru este oarecum neintuitiv și trebuie reamintit aici.

1.5.8 Secvențe tipice de comenzi și redirecționarea shell-ului

Să încercăm să reținem următoarele expresii ale comenzii shell tastate într-o singură linie ca parte a comenzii shell.

Sistemul Debian este un sistem multi-sarcini. Sarcinile de fundal permit utilizatorilor să ruleze mai multe programe într-un singur shell. Gestionarea proceselor în fundal implică utilizarea comenzilor încorporate ale shell-ului: `jobs`, `fg`, `bg` și `kill`. Vă rugăm să citiți secțiunile din `bash(1)` sub „SEMNALE” și „CONTROLUL SARCINILOR” și `builtins(1)`.

De exemplu, încercați următoarele

```
$ </etc/motd pager
```

```
$ pager </etc/motd
```

```
$ pager /etc/motd
```

```
$ cat /etc/motd | pager
```

Deși toate cele 4 exemple de redirecționări shell afișează același lucru, ultimul exemplu execută o comandă suplimentară `cat` și irosește resurse fără niciun motiv.

Shell-ul vă permite să deschideți fișiere folosind comanda internă `exec` cu un descriptor de fișier arbitrar.

expresia idiomatică a comenzii	descriere
comanda &	execuție în fundal a comenzii în subshell
comanda1 comanda2	redirecționează ieșirea standard a comenzii1 către intrarea standard a comenzii2 (execuție concomitentă)
comanda1 2>&1 comanda2	redirecționează atât ieșirea standard, cât și eroarea standard a comenzii1 către intrarea standard a comenzii2 (execuție concomitentă)
comanda1 ; comanda2	execută comanda1 și comanda2 secvențial
comanda1 && comanda2	execută comanda1; dacă are succes, execută comanda2 secvențial (returnează succes dacă atât comanda1 cât și comanda2 au succes)
comanda1 comanda2	execută comanda1; dacă nu are succes, execută comanda2 secvențial (returnează succes dacă comanda1 sau comanda2 au succes)
comanda > foo	redirecționează ieșirea standard a comenzii comanda către un fișier foo (suprascriptie)
comanda 2> foo	redirecționează ieșirea de eroare standard a comenzii comanda către un fișier foo (suprascriptie)
comanda >> foo	redirecționează ieșirea standard a comenzii comanda către un fișier foo (adăugare)
comanda 2>> foo	redirecționează ieșirea de eroare standard a comenzii comanda către un fișier foo (adăugare)
comanda > foo 2>&1	redirecționează atât ieșirea standard, cât și ieșirea de eroare standard a comenzii comanda către un fișier foo
comanda < foo	redirecționează intrarea standard a comenzii comanda către un fișier foo
comanda << delimitator	redirecționează intrarea standard a comenzii comanda către următoarele linii până când se întâlnește „delimitatorul” (aici, document)
comanda <<- delimitator	redirecționează intrarea standard a comenzii comanda către următoarele linii până când se întâlnește „delimitatorul” (aici document, caracterele de tabulare din fața liniilor de intrare sunt eliminate)

Tabela 1.23: Expresii idiomatice ale comenzii shell

```
$ echo Hello >foo
$ exec 3<foo 4>bar # open files
$ cat <&3 >&4 # redirect stdin to 3, stdout to 4
$ exec 3<&- 4>&- # close files
$ cat bar
Hello
```

Descriptorii de fișiere 0-2 sunt predefiniți.

dispozitiv	descriere	descriptor de fișier
stdîn	intrarea standard	0
stdout	ieșirea standard	1
stderr	ieșirea de eroare standard	2

Tabela 1.24: Descriptori de fișiere predefiniți

1.5.9 Alias de comandă

Puteți defini un alias pentru comanda utilizată frecvent.

De exemplu, încercați următoarele

```
$ alias la='ls -la'
```

Acum, «la» funcționează ca o prescurtare pentru «ls -la», care listează toate fișierele în formatul de listare lungă. Puteți lista orice alias existente prin `alias` (consultați `bash(1)` la «COMENZILE INTERNE ALE SHELL-ului»).

```
$ alias
...
alias la='ls -la'
```

Puteți identifica ruta exactă sau identitatea comenzii prin `type` (consultați `bash(1)` la «COMENZILE INTERNE ALE SHELL-ului»).

De exemplu, încercați următoarele

```
$ type ls
ls is hashed (/bin/ls)
$ type la
la is aliased to ls -la
$ type echo
echo is a shell builtin
$ type file
file is /usr/bin/file
```

Aici, «ls» a fost căutat recent, în timp ce «file» nu a fost, astfel încât «ls» este indexat, adică shell-ul are o înregistrare internă pentru accesul rapid la locația comenzii «ls».

Indicație

A se vedea Secțiune [9.3.6](#).

1.6 Procesarea textului în stilul Unix

În mediul de lucru de tip Unix, procesarea textului se realizează prin transferul textului prin intermediul unor lanțuri de instrumente standard de procesare a textului. Aceasta a fost o altă inovație crucială a sistemului Unix.

1.6.1 Instrumente pentru text în Unix

Există câteva instrumente standard de procesare a textului care sunt utilizate foarte des în sistemele de tip Unix.

- Nu se utilizează nicio expresie regulată:
 - `cat(1)` concatenează fișiere și afișează întregul conținut.
 - `tac(1)` concatenează fișierele și le afișează în ordine inversă.
 - `cut(1)` selectează părți din linii și le afișează.
 - `head(1)` afișează prima parte a fișierelor.
 - `tail(1)` afișează ultima parte a fișierelor.
 - `sort(1)` sortează liniile din fișierele text.
 - `uniq(1)` elimină liniile duplicate dintr-un fișier sortat.
 - `tr(1)` traduce sau șterge caractere.
 - `diff(1)` compară fișierele linie cu linie.
- Expresia regulată de bază („Basic regular expression”: **BRE**) este utilizată în mod implicit:
 - `ed(1)` este un editor de linii primitiv.
 - `sed(1)` este un editor de flux.
 - `grep(1)` potrivește textul cu modelele.
 - `vim(1)` este un editor de ecran.
 - `emacs(1)` este un editor de ecran. (oarecum extins cu **BRE**)
- Se utilizează expresia regulată extinsă („Extended regular expression”: **ERE**):
 - `awk(1)` efectuează procesarea simplă a textului.
 - `egrep(1)` potrivește textul cu modelele.
 - `tc(1)` poate efectua orice operație de procesare a textului: consultați `re_syntax(3)`. Se utilizează adesea împreună cu `tk(3)`.
 - `perl(1)` poate efectua orice operație de procesare a textului imaginabilă. Consultați `perlre(1)`.
 - `pcgrep(1)` din pachetul `pcgrep` potrivește textul cu modelul [Perl Compatible Regular Expressions \(PCRE\)](#).
 - `python(1)` cu modulul `re` poate efectua orice procesare de text imaginabilă. Consultați „`/usr/share/doc/python/h`”.

Dacă nu sunteți sigur ce fac exact aceste comenzi, vă rugăm să utilizați «`man comanda`» pentru a afla singur.

Notă

Ordinea de sortare și expresia intervalului depind de configurația regională. Dacă doriți să obțineți comportamentul tradițional pentru o comandă, utilizați parametrul de configurare regională **C** sau **C.UTF-8** în locul celor normale **UTF-8** (consultați Secțiune [8.1](#)).

Notă

Expresiile regulate [Perl](#)(`perlre(1)`), [Expresii regulate compatibile cu Perl](#) („Perl Compatible Regular Expressions”: [PCRE](#)) și expresiile regulate [Python](#) oferite de modulul `re` au multe extensii comune cu **ERE** normal.

BRE	ERE	descrierea expresiei regulate
\ . [] ^ \$ *	\ . [] ^ \$ *	metacaractere comune
\+ \? \(\) \{ \} \		BRE doar metacaractere eludate „\”
	+ ? () { }	ERE doar metacaractere fără caracterul de eludare „\”
c	c	potrivește cu un non-metacarakter „c”
\c	\c	potrivește un caracter literal „c” chiar dacă „c” este de fapt un metacarakter
.	.	potrivește orice caracter, inclusiv cel de linie nouă
^	^	poziția la începutul unui șir de caractere
\$	\$	poziția la sfârșitul unui șir de caractere
\<	\<	poziția la începutul unui cuvânt
\>	\>	poziția la sfârșitul unui cuvânt
[abc...]	[abc...]	potrivește orice caractere din „abc...”
[^abc...]	[^abc...]	potrivește orice caractere, cu excepția celor din „abc...”
r*	r*	potrivește zero sau mai multe expresii regulate identificate prin „r”
r\+	r+	potrivește una sau mai multe expresii regulate identificate prin „r”
r\?	r?	potrivește zero sau o expresie regulată identificată prin „r”
r1\ r2	r1 r2	potrivește una dintre expresiile regulate identificate prin „r1” sau „r2”
\(r1\ r2\)	(r1 r2)	potrivește una dintre expresiile regulate identificate prin „r1” sau „r2” și o tratează ca o expresie regulată încadrată între paranteze

Tabela 1.25: Metacaractere pentru BRE și ERE

1.6.2 Expresii regulate

Expresiile regulate sunt utilizate în multe instrumente de procesare a textului. Acestea sunt similare cu modelele globale din shell, dar sunt mai complicate și mai puternice.

Expresia regulată descrie modelul de potrivire și este alcătuită din caractere text și **metacaractere**.

Un **metacarakter** este doar un caracter cu o semnificație specială. Există două stiluri principale, **BRE** și **ERE**, în funcție de instrumentele de text descrise mai sus.

Expresia regulată **emacs** este în esență **BRE**, dar a fost extinsă pentru a trata „+” și „?” ca **metacaractere** ca în **ERE**. Astfel, nu este necesar să le eludați cu „\” în expresia regulată a emacs.

grep(1) poate fi utilizat pentru a efectua căutarea textului folosind o expresie regulată.

De exemplu, încercați următoarele

```
$ egrep 'GNU.*LICENSE|Yoyodyne' /usr/share/common-licenses/GPL
GNU GENERAL PUBLIC LICENSE
GNU GENERAL PUBLIC LICENSE
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
```

Indicație

A se vedea Secțiune 9.3.6.

1.6.3 Expresii de înlocuire

Pentru expresia de înlocuire, unele caractere au semnificații speciale.

expresie de înlocuire	descrierea textului care va înlocui expresia de înlocuire
&	ceea ce a fost găsit de expresia regulată (utilizați \& în emacs)
\n	cu ce s-a potrivit a n-a expresie regulată între paranteze (unde „n” este un număr)

Tabela 1.26: Expresia de înlocuire

Pentru înlocuirea șirurilor în Perl, se utilizează „\$&” în loc de „&” și „\$n” în loc de „\n”.

De exemplu, încercați următoarele

```
$ echo zzz1abc2efg3hij4 | \
sed -e 's/(1[a-z]*)[0-9]*\(.*)$/=&/'
zzz=1abc2efg3hij4=
$ echo zzz1abc2efg3hij4 | \
sed -E -e 's/(1[a-z]*)[0-9]*\(.*)$/=&/'
zzz=1abc2efg3hij4=
$ echo zzz1abc2efg3hij4 | \
perl -pe 's/(1[a-z]*)[0-9]*\(.*)$/$&/'
zzz=1abc2efg3hij4=
$ echo zzz1abc2efg3hij4 | \
sed -e 's/(1[a-z]*)[0-9]*\(.*)$/\2===\1/'
zzzefg3hij4===1abc
$ echo zzz1abc2efg3hij4 | \
sed -E -e 's/(1[a-z]*)[0-9]*\(.*)$/\2===\1/'
zzzefg3hij4===1abc
$ echo zzz1abc2efg3hij4 | \
perl -pe 's/(1[a-z]*)[0-9]*\(.*)$/$2===\1/'
zzzefg3hij4===1abc
```


Aici, vă rugăm să acordați o atenție deosebită stilului expresiei regulate **în paranteze** și modului în care șirurile potrivite sunt utilizate în procesul de înlocuire a textului în diferite instrumente.

Aceste expresii regulate pot fi utilizate și pentru mișcările cursorului și acțiunile de înlocuire a textului în unele editoare.

Bară oblică inversă „\” la sfârșitul liniei în linia de comandă a shell-ului eludează caracterul de linie nouă ca caracter spațiu alb și continuă introducerea liniei de comandă a shell-ului în linia următoare.

Vă rugăm să citiți toate paginile de manual aferente pentru a învăța aceste comenzi.

1.6.4 Înlocuire globală cu expresii regulate

Comanda `ed(1)` poate înlocui toate aparițiile „FROM_REGEX” cu „TO_TEXT” în „file”.

```
$ ed file <<EOF
,s/FROM_REGEX/TO_TEXT/g
w
q
EOF
```

Comanda `sed(1)` poate înlocui toate aparițiile „FROM_REGEX” cu „TO_TEXT” în „file”.

```
$ sed -i -e 's/FROM_REGEX/TO_TEXT/g' file
```

Comanda `vim(1)` poate înlocui toate aparițiile „FROM_REGEX” cu „TO_TEXT” în „file” folosind comenzi `ex(1)`.

```
$ vim +%s/FROM_REGEX/TO_TEXT/gc' '+update' '+q' file
```

Indicație

Indicatorul „c” din exemplul de mai sus asigură confirmarea interactivă pentru fiecare substituție.

Mai multe fișiere („file1”, „file2” și „file3”) pot fi procesate cu expresii regulate în mod similar cu `vim(1)` sau `perl(1)`.

```
$ vim '+argdo %s/FROM_REGEX/TO_TEXT/gce|update' '+q' file1 file2 file3
```

Indicație

Indicatorul „e” din exemplul de mai sus împiedică eroarea „No match” (fără corespondență) să întrerupă o căutare.

```
$ perl -i -p -e 's/FROM_REGEX/TO_TEXT/g;' file1 file2 file3
```

În exemplul `perl(1)`, „-i” este pentru editarea pe loc (fără folosirea unui tampon/fișier temporal pentru aceasta) a fiecărui fișier țintă, iar „-p” este pentru bucla implicită peste toate fișierele date.

Indicație

Utilizarea argumentului „-i.bak” în locul „-i” păstrează fiecare fișier original prin adăugarea „.bak” la numele fișierului. Acest lucru facilitează recuperarea în cazul erorilor pentru substituții complexe.

Notă

`ed(1)` și `vim(1)` sunt **BRE**; `perl(1)` este **ERE**.

1.6.5 Extragerea datelor din tabelul fișierului text

Să luăm în considerare un fișier text numit „DPL” în care sunt enumerate numele unor lideri ai proiectului Debian dinainte de 2004 și data inițierii lor, într-un format separat prin spații.

Ian	Murdock	August	1993
Bruce	Perens	April	1996
Ian	Jackson	January	1998
Wichert	Akkerman	January	1999
Ben	Collins	April	2001
Bdale	Garbee	April	2002
Martin	Michlmayr	March	2003

Indicație

Consultați [„O scurtă istorie a Debian”](#) pentru cea mai recentă [istorie a conducerii Debian](#).

Awk este frecvent utilizat pentru extragerea datelor din aceste tipuri de fișiere.

De exemplu, încercați următoarele

```
$ awk '{ print $3 }' <DPL                # month started
August
April
January
January
April
April
March
$ awk '($1=="Ian") { print }' <DPL        # DPL called Ian
Ian   Murdock   August   1993
Ian   Jackson   January  1998
$ awk '($2=="Perens") { print $3,$4 }' <DPL # When Perens started
April 1996
```

Shell-uri precum Bash pot fi, de asemenea, utilizate pentru a analiza acest tip de fișiere.

De exemplu, încercați următoarele

```
$ while read first last month year; do
    echo $month
done <DPL
... same output as the first Awk example
```

Aici, comanda internă `read` utilizează caracterele din „IFS” („internal field separators” - separatoare interne de câmpuri) pentru a împărți liniile în cuvinte.

Dacă schimbați „IFS” cu „:”, puteți analiza „/etc/passwd” cu shell-ul fără probleme.

```
$ oldIFS="$IFS"    # save old value
$ IFS=':'
$ while read user password uid gid rest_of_line; do
    if [ "$user" = "bozo" ]; then
        echo "$user's ID is $uid"
    fi
done < /etc/passwd
bozo's ID is 1000
$ IFS="$oldIFS"    # restore old value
```

(Dacă se utilizează Awk pentru a face același lucru, utilizați „FS=':'” pentru a defini separatorul de câmpuri.)

IFS este utilizat și de shell pentru a împărți rezultatele expansiunii parametrilor, substituției comenzilor și expansiunii aritmetice. Acestea nu apar în cuvintele între ghilimele duble sau simple. Valoarea implicită a IFS este *spațiu, tabulator și linie nouă* combinate.

Aveți grijă când utilizați aceste trucuri IFS ale shell-ului. Pot apărea lucruri ciudate atunci când shell-ul interpretează anumite părți ale scriptului ca fiind **intrare**.

```
$ IFS=":," # use ":" and "," as IFS
$ echo IFS=$IFS, IFS="$IFS" # echo is a Bash builtin
IFS= , IFS=:
$ date -R # just a command output
Sat, 23 Aug 2003 08:30:15 +0200
$ echo $(date -R) # sub shell --> input to main shell
Sat 23 Aug 2003 08 30 36 +0200
$ unset IFS # reset IFS to the default
$ echo $(date -R)
Sat, 23 Aug 2003 08:30:50 +0200
```

1.6.6 Fragmente de script pentru comenzi de direcționare

Următoarele scripturi fac lucruri interesante ca parte a unei conduite (direcționări).

fragment de script (introduceți într-o singură linie)	efectul comenzii
<code>find /usr -print</code>	găsește toate fișierele din „/usr”
<code>seq 1 100</code>	imprimă numere de la 1 până la 100
<code> xargs -n 1 comanda</code>	rulează comanda în mod repetat cu fiecare element din conductă ca argument al său
<code> xargs -n 1 echo</code>	împarte elementele separate prin spații albe din conductă în linii
<code> xargs echo</code>	fuzionează toate liniile din conductă într-o singură linie
<code> grep -e model_expresie-regulată</code>	extrage linii din conductă care conțin <i>model_expresie-regulată</i>
<code> grep -v -e model_expresie-regulată</code>	extrage liniile din conductă care nu conțin <i>model_expresie-regulată</i>
<code> cut -d: -f3 -</code>	extrage al treilea câmp din conductă separat prin „:” (fișier passwd etc.)
<code> awk '{ print \$3 }'</code>	extrage al treilea câmp din șirul separat prin spații
<code> awk -F'\t' '{ print \$3 }'</code>	extrage al treilea câmp din șirul separat prin tabulator
<code> col -bx</code>	elimină „backspace” (retururile de cărucior) și transformă tabulatoarele în spații
<code> expand -</code>	transformă tabulatoarele în spații
<code> sort uniq</code>	sortează și elimină duplicatele
<code> tr 'A-Z' 'a-z'</code>	convertește majusculele în minuscule
<code> tr -d '\n'</code>	concatenează liniile într-o singură linie
<code> tr -d '\r'</code>	elimină CR (retururile de cărucior)
<code> sed 's/^/# /'</code>	adaugă „#” la începutul fiecărei linii
<code> sed 's/\.ext//g'</code>	elimină „.ext”
<code> sed -n -e 2p</code>	imprimă a doua linie
<code> head -n 2 -</code>	imprimă primele 2 linii
<code> tail -n 2 -</code>	imprimă ultimele 2 linii

Tabela 1.27: Lista fragmentelor de script pentru comenzi de direcționare

Un script shell de o singură linie poate parcurge mai multe fișiere folosind `find(1)` și `xargs(1)` pentru a efectua sarcini destul de complicate. Vedeți Secțiune [10.1.5](#) și Secțiune [9.4.9](#).

Când utilizarea modului interactiv al shell-ului devine prea complicată, luați în considerare scrierea unui script shell (consultați Secțiune [12.1](#)).

Capitolul 2

Gestionarea pachetelor Debian

Notă

Acest capitol este scris presupunând că cea mai recentă versiune stabilă este nume în cod: `trixie`.

Sursa de date a sistemului APT este denumită în mod colectiv **lista surselor** în acest document . Aceasta poate fi definită oriunde în fișierul „`/etc/apt/sources.list`”, în fișierele „`/etc/apt/sources.list.d/*.list`” sau în fișierele „`/etc/apt/sources.list.d/*.sources`”.

2.1 Cerințe preliminare pentru gestionarea pachetelor Debian

2.1.1 Sistemul de gestionare a pachetelor Debian

[Debian](#) este o organizație voluntară care creează distribuții **consistente** de pachete binare precompilate de software liber și le distribuie din arhiva sa.

[Arhiva Debian](#) este oferită de [multe situri oglindă la distanță](#) pentru acces prin metode HTTP și FTP. Este disponibilă și pe [CD-ROM/DVD](#).

Sistemul actual de gestionare a pachetelor Debian care poate utiliza toate aceste resurse este [Advanced Packaging Tool \(APT\)](#).

Sistemul de gestionare a pachetelor Debian, **atunci când este utilizat corect**, oferă utilizatorului posibilitatea de a instala **seturi consistente de pachete binare** în sistem din arhivă. În prezent, există 74980 de pachete disponibile pentru arhitectura amd64.

Sistemul de gestionare a pachetelor Debian are o istorie bogată și oferă numeroase opțiuni pentru programul de interfață cu utilizatorul final și metoda de acces la arhiva care urmează să fie utilizată. În prezent, recomandăm următoarele.

- `apt(8)` pentru toate operațiile interactive din linia de comandă, inclusiv instalarea, eliminarea și actualizarea pachetelor.
 - `apt-get(8)` pentru apelarea sistemului de gestionare a pachetelor Debian din scripturi. Este, de asemenea, o opțiune de rezervă atunci când `apt` nu este disponibil (adesea în cazul sistemelor Debian mai vechi).
 - `aptitude(8)` pentru o interfață text interactivă pentru gestionarea pachetelor instalate și căutarea pachetelor disponibile.
-

pachet	popcon(popularitate)	dimensiune	descriere
dpkg	V:894, I:999	6350	sistem de gestionare a pachetelor de nivel scăzut pentru Debian (bazat pe fișiere)
apt	V:882, I:999	4632	interfață APT pentru gestionarea pachetelor cu CLI:(interfață de linie de comandă) apt / apt-get / apt-cache
aptitude	V:36, I:189	4624	interfață APT pentru gestionarea interactivă a pachetelor cu consolă pe ecran complet: aptitude (8)
tasksel	V:35, I:983	349	interfață APT pentru instalarea sarcinilor selectate: tasksel (8)
unattended-upgrades	V:126, I:188	317	pachet de îmbunătățiri pentru APT pentru a permite instalarea automată a actualizărilor de securitate
gnome-software	V:163, I:270	4208	Centru software pentru GNOME (interfață grafică pentru APT)
synaptic	V:36, I:310	7788	administrator grafic de pachete (interfață GTK APT)
apt-utils	V:388, I:998	1148	programe utilitare APT: apt-extracttemplates (1), apt-ftpparchive (1) și apt-sortpkgs (1)
apt-listchanges	V:368, I:886	553	instrument de notificare a istoricului modificărilor pachetelor
apt-listbugs	V:5, I:7	514	listează erorile critice înainte de fiecare instalare APT
apt-file	V:15, I:60	89	instrument de căutare a pachetelor APT — interfață linie de comandă
apt-rdepends	V:0, I:4	39	listează recursiv dependențele pachetului

Tabela 2.1: Lista instrumentelor de gestionare a pachetelor Debian

2.1.2 Configurația pachetului

Iată câteva puncte cheie pentru configurarea pachetelor în sistemul Debian.

- Configurația manuală efectuată de administratorul de sistem este respectată. Cu alte cuvinte, sistemul de configurare a pachetelor nu efectuează nicio configurare intruzivă din motive de comoditate.
- Fiecare pachet vine cu propriul script de configurare cu interfață standardizată numită [debconf](#)(7) pentru a ajuta în procesul de instalare inițială a pachetului.
- Dezvoltatorii Debian fac tot posibilul pentru a vă oferi o experiență de actualizare fără probleme, cu ajutorul scrip-turilor de configurare a pachetelor.
- Administratorul de sistem are la dispoziție toate funcționalitățile pachetului software. Însă cele care prezintă riscuri de securitate sunt dezactivate în instalarea implicită.
- Dacă activați manual un serviciu care prezintă anumite riscuri de securitate, sunteți responsabil pentru limitarea riscurilor.
- Configurația ezoterică poate fi activată manual de administratorul de sistem. Acest lucru poate crea interferențe cu programele generice populare de asistență pentru configurarea sistemului.

2.1.3 Precauții de bază



Avertisment

Nu instalați pachete dintr-un amestec aleatoriu de suite. Probabil că acest lucru va afecta coerența pachetului, ceea ce necesită cunoștințe aprofundate de administrare a sistemului, cum ar fi compilatorul [ABI](#), versiunea [bibliotecii](#), caracteristicile interpretului etc.

Administratorul de sistem Debian [începător](#) ar trebui să rămână la versiunea **stabilă** a Debian, aplicând doar actualizările de securitate. Până când nu înțelegeți foarte bine sistemul Debian, ar trebui să urmați următoarele precauții.

- Nu includeți suitele **testing** (de testare) sau **unstable** (instabilă) în **lista sursă**.
- Nu amestecați Debian standard cu alte arhive non-Debian, cum ar fi Ubuntu, în **lista de surse**.
- Nu creați „/etc/apt/preferences”.
- Nu modificați comportamentul implicit al instrumentelor de gestionare a pachetelor prin fișiere de configurare fără a cunoaște impactul complet al acestora.
- Nu instalați pachete aleatorii folosind comanda „dpkg -i *pachet_aleatoriu*”.
- Nu instalați niciodată pachete aleatorii folosind comanda „dpkg --force-all -i *pachet_aleatoriu*”.
- Nu ștergeți și nu modificați fișierele din „/var/lib/dpkg/”.
- Nu suprascrieți fișierele de sistem prin instalarea programelor software compilate direct din sursă.
 - Instalați-le în „/usr/local” sau „/opt”, dacă este necesar.

Efectele incompatibile cauzate de încălcarea precauțiilor de mai sus asupra sistemului de gestionare a pachetelor Debian pot face sistemul dvs. inutilizabil.

Administratorii de sistem Debian serioși, care gestionează servere critice, ar trebui să ia măsuri de precauție suplimentare.

- Nu instalați niciun pachet, inclusiv actualizările de securitate de la Debian, fără a le testa temeinic cu configurația dvs. specifică, în condiții de siguranță.
 - În calitate de administrator de sistem, dumneavoastră sunteți responsabil în ultimă instanță pentru sistemul dumneavoastră.
 - Istoria îndelungată de stabilitate a sistemului Debian nu constituie o garanție în sine.

2.1.4 O viață cu îmbunătățiri veșnice



Atenție

Pentru **serverul de producție**, se recomandă suita stabilă cu actualizările de securitate. Același lucru se poate spune și despre calculatoarele stații de lucru pe care puteți depune eforturi administrative limitate.

În ciuda avertismentelor mele de mai sus, știu că mulți cititori ai acestui document ar putea dori să ruleze noile suite **testing** sau **unstable**.

[Iluminarea](#) cu următoarele salvează o persoană de lupta eternă [karmică](#) a ascensiunii [iadului](#) și îi permite să atingă [nirvana](#) Debian.

Această listă este destinată mediului de birou **autoadministrat**.

- Utilizați suita **testing**, deoarece aceasta este practic versiunea continuă gestionată automat de infrastructura QA a arhivei Debian, cum ar fi [integrarea continuă Debian](#), [practicile de încărcare numai a surselor](#) și [urmărirea tranziției bibliotecilor](#). Pachetele din suita **testing** sunt actualizate suficient de frecvent pentru a oferi toate funcționalitățile cele mai recente.
 - Stabiliți numele de cod corespunzător suitei **testing** („forky” în timpul ciclului de lansare *trixie-ca-stable*) în **lista surselor**.
 - Actualizați manual acest nume de cod în **lista surselor** cu cel nou numai după ce evaluați situația pe cont propriu timp de aproximativ o lună după lansarea suitei majore. Lista de discuții a utilizatorilor și dezvoltatorilor Debian este, de asemenea, o sursă bună de informații în acest sens.
-

Utilizarea suitei `unstable` nu este recomandată. Suita `unstable` este **potrivită pentru depanarea pachetelor** ca dezvoltator, dar tinde să vă expună la riscuri inutile pentru utilizarea normală a stației de lucru. Chiar dacă suita `unstable` a sistemului Debian pare foarte stabilă în majoritatea cazurilor, au existat unele probleme cu pachetele, iar câteva dintre ele nu au fost atât de ușor de rezolvat.

Iată câteva idei de măsuri de precauție de bază pentru a asigura recuperarea rapidă și ușoară în cazul apariției erorilor în pachetele Debian.

- Faceți sistemul să fie cu **pornire duală** instalând suita `stable` a sistemului Debian pe o altă partiție
- Păstrați CD-ul de instalare la îndemână pentru **pornirea de recuperare**
- Luați în considerare instalarea `apt-listbugs` pentru a verifica informațiile din [Sistemul de urmărire a erorilor Debian \(BTS\)](#) înainte de actualizare.
- Învățați infrastructura sistemului de pachete suficient pentru a rezolva problema

**Atenție**

Dacă nu puteți lua niciuna dintre aceste măsuri de precauție, probabil că nu sunteți pregătit pentru suitele `testing` (de testare) și `unstable` (instabilă).

2.1.5 Noțiuni de bază despre arhiva Debian

Indicație

Politica oficială a arhivei Debian este definită în [Manualul de politici Debian, Capitolul 2 - Arhiva Debian](#).

Să analizăm [arhiva Debian](#) din perspectiva unui utilizator de sistem.

Pentru un utilizator al sistemului, [arhiva Debian](#) este accesată folosind sistemul APT.

Sistemul APT specifică sursa de date ca fiind **lista surselor** și este descris în `sources.list(5)`.

Pentru sistemul `trixie` cu acces HTTP tipic, **lista surselor** în stilul unei singure linii, este ca mai jos:

```
deb http://deb.debian.org/debian/ trixie main non-free-firmware contrib non-free
deb-src http://deb.debian.org/debian/ trixie main non-free-firmware contrib non-free

deb http://security.debian.org/debian-security trixie-security main non-free-firmware ↵
  contrib non-free
deb-src http://security.debian.org/debian-security trixie-security main non-free-firmware ↵
  contrib non-free
```

Alternativ, lista surselor echivalente în stil `deb822` este următoarea.

```
Types: deb deb-src
URIs: http://deb.debian.org/debian/
Suites: trixie
Components: main non-free-firmware contrib non-free
```

```
Types: deb deb-src
URIs: http://security.debian.org/debian-security/
Suites: trixie-security
Components: main non-free-firmware contrib non-free
```

Punctele cheie ale **listei surselor** sunt următoarele.

- Format cu o singură linie
 - Fișierele sale de definiții se află în fișierul „`/etc/apt/sources.list`” și în fișierele „`/etc/apt/sources.list.d/*`”
 - Fiecare linie definește sursa de date pentru sistemul APT.
 - Linia „`deb`” definește pachetele binare.
 - Linia „`deb-src`” definește pachetele sursă.
 - Primul argument este adresa URL rădăcină a arhivei Debian.
 - Al doilea argument este numele distribuției, folosind fie numele suitei, fie numele în cod.
 - Al treilea argument și următoarele sunt lista numelor valide ale zonelor de arhivă din arhiva Debian.
- Format stil Deb822
 - Fișierele sale de definiție se află în fișierele „`/etc/apt/sources.list.d/*.sources`”.
 - Fiecare bloc de linii separate printr-o linie goală definește sursa de date pentru sistemul APT.
 - Strofa „`Types:`” definește lista tipurilor, cum ar fi „`deb`” și „`deb-src`”.
 - Strofa „`URIs:`” definește lista adreselor URI rădăcină ale arhivei Debian.
 - Strofa „`Suites:`” definește lista numelor de distribuție utilizând fie numele suitei, fie numele în cod.
 - Strofa „`Components:`” definește lista numelor valide ale zonelor de arhivă din arhiva Debian.

Definiția pentru „`deb-src`” poate fi omisă în siguranță dacă este doar pentru aptitude, care nu accesează meta-datele legate de sursă. Aceasta accelerează actualizările metadatelor arhivei.

Adresa URL poate fi de tipul „`https://`”, „`http://`”, „`ftp://`”, „`file://`”, ...

Liniile care încep cu „`#`” sunt comentarii și sunt ignorate.

Aici, tind să folosesc numele în cod „`trixie`” sau „`forky`” în loc de numele suitei „`stable`” sau „`testing`” pentru a evita surprizele atunci când va fi lansată următoarea versiune `stable`.

Indicație

Dacă în exemplul de mai sus se utilizează „`sid`” în loc de „`trixie`”, nu este necesară linia „`deb: http://security.debian.org/ ...`” sau conținutul său echivalent `deb822` pentru actualizările de securitate din **lista surselor** nu este necesar. Acest lucru se datorează faptului că nu există arhivă de actualizări de securitate pentru „`sid`” (`unstable`).

Iată lista adreselor URL ale siturilor de arhivă Debian și numele suitei sau numele de cod utilizate în fișierul de configurare după versiunea `trixie`.

Atenție



Numai versiunea pură **stabilă** cu actualizări de securitate oferă cea mai bună stabilitate. Rularea în principal a versiunii **stable** combinată cu unele pachete din versiunea **testing** sau **unstable** este mai riscantă decât rularea versiunii pure **unstable** din cauza incompatibilității versiunilor bibliotecilor etc. Dacă aveți cu adevărat nevoie de cea mai recentă versiune a unor programe din versiunea **stable**, vă rugăm să utilizați pachetele din serviciile [stable-updates](#) și [backports](#) (a se vedea Secțiune 2.7.4). Aceste servicii trebuie utilizate cu precauție sporită.

Atenție



În principiu, ar trebui să enumerați doar una dintre suitele `stable`, `testing` sau `unstable` în linia „`deb`”. Dacă listați orice combinație de suite `stable`, `testing` și `unstable` în linia „`deb`”, programele APT vor funcționa mai lent, iar numai arhiva cea mai recentă va fi eficientă. Listarea multiplă are sens pentru acestea atunci când fișierul „`/etc/apt/preferences`” este utilizat cu obiective clare (a se vedea Secțiune 2.7.7).

adresa URL a arhivei	numele suitei	nume în cod	scopul depozitului
http://deb.debian.org/debian/	stable	trixie	Publicare stabilă cvasi-statică după verificări ample
http://deb.debian.org/debian/	testing	forky	Publicare dinamică a suitei de testare după verificări adecvate și perioade scurte de așteptare
http://deb.debian.org/debian/	unstable	sid	Publicare dinamică a suitei instabile după verificări minime și fără așteptări
http://deb.debian.org/debian/	experimental	N/D	Experimentări pre-publicare efectuate de dezvoltatori (opțional, numai pentru dezvoltatori)
http://deb.debian.org/debian/	stable-proposed-updates	trixie	Actualizări de securitate pentru versiunea stabilă (opțional)
http://deb.debian.org/debian/	stable-updates	trixie	Subset al suitei stable-proposed-updates care necesită actualizări urgente, cum ar fi datele privind fusul orar (opțional)
http://deb.debian.org/debian/	stable-backports	trixie	Colecție aleatorie de pachete recompilate, în mare parte din versiunea testing (opțional)
http://security.debian.org/debian-security/	stable-security	trixie	Actualizări de securitate pentru versiunea stabilă (important)
http://security.debian.org/debian-security/	testing-security	forky	Aceasta nu este susținută în mod activ și nici utilizată de echipa de securitate

Tabela 2.2: Lista siturilor de arhivă Debian

Indicație

Pentru sistemul Debian cu suita stable, este recomandat să includeți conținutul cu „<http://security.debian.org/>” în **lista de surse** pentru a activa actualizările de securitate, așa cum se arată în exemplul de mai sus.

Notă

Erorile/problemele de securitate pentru arhiva stable sunt remediate de echipa de securitate Debian. Această activitate a fost destul de riguroasă și fiabilă. Cele pentru arhiva testing pot fi remediate de echipa de securitate Debian testing. Din **mai multe motive**, această activitate nu este la fel de riguroasă ca cea pentru stable și este posibil să fie necesar să așteptați migrarea pachetelor unstable remediate către arhiva testing. Cele pentru arhiva unstable sunt remediate de către administratorul individual. Pachetele unstable întreținute activ sunt de obicei într-o stare destul de bună, beneficiind de cele mai recente remedieri de securitate din amonte. Consultați [FAQ-ul de securitate Debian](#) pentru a afla cum gestionează Debian erorile/problemele de securitate.

secțiunea	numărul de pachete	criterii privind componentele pachetului
main	73518	conform cu DFSG și fără dependență de non-free
non-free-firmware	50	nu este conform cu DFSG, necesită firmware pentru o experiență rezonabilă de instalare a sistemului
contrib	376	conform cu DFSG, dar cu dependență de non-free
non-free	1036	nu este conform cu DFSG și nu se află în non-free-firmware

Tabela 2.3: Lista secțiunilor de arhivă Debian

Aici numărul de pachete din cele de mai sus este pentru arhitectura amd64. Secțiunea main furnizează sistemul Debian (vedeți Secțiune [2.1.6](#)).

Organizarea arhivei Debian poate fi studiată cel mai bine accesând în navigatorul dvs. fiecare adresă URL a arhivei la care se adaugă `dists` sau `pool`.

Distribuția este menționată în două moduri, versiunea sau [nume în cod](#). Termenul „distribuție” este utilizat alternativ ca sinonim pentru „suită” în multe documentații. Relația dintre versiune și numele în cod poate fi rezumată după cum urmează.

Calendarul	versiunea = <code>stable</code>	versiunea = <code>testing</code>	versiunea = <code>unstable</code>
după publicarea versiunii <code>trixie</code>	nume în cod = <code>trixie</code>	nume în cod = <code>forky</code>	nume în cod = <code>sid</code>
după publicarea versiunii <code>forky</code>	nume în cod = <code>forky</code>	nume în cod = <code>duke</code>	nume în cod = <code>sid</code>

Tabela 2.4: Relația dintre versiune și numele în cod

Istoria numelor în cod este descrisă în [Debian FAQ: 6.2.1 Ce alte nume în cod au fost folosite în trecut?](#)

În terminologia mai strictă a arhivei Debian, cuvântul „secțiune” este utilizat în mod specific pentru clasificarea pachetelor în funcție de domeniul de aplicare. (Cu toate acestea, cuvântul „secțiune principală” poate fi utilizat uneori pentru a descrie zona arhivei Debian denumită „main”).

De fiecare dată când un dezvoltator Debian (DD) efectuează o nouă încărcare în arhiva `unstable` (prin procesarea [incoming](#)), DD trebuie să se asigure că pachetele încărcate sunt compatibile cu cel mai recent set de pachete din cea mai recentă arhivă `unstable`.

Dacă DD încalcă intenționat această compatibilitate pentru actualizări importante ale bibliotecii etc., de obicei se face un anunț pe [lista de discuții debian-devel](#) etc.

Înainte ca un set de pachete să fie mutat de scriptul de întreținere a arhivei Debian din arhiva `unstable` în arhiva `testing`, scriptul de întreținere a arhivei nu numai că verifică maturitatea (aproximativ 2-10 zile) și starea rapoartelor de erori RC pentru pachete, dar încearcă și să se asigure că acestea sunt compatibile cu cel mai recent set de pachete din arhiva `testing`. Acest proces face ca arhiva `testing` să fie foarte actuală și utilizabilă.

Prin procesul gradual de înghețare a arhivei condus de echipa de lansare, arhiva `testing` este maturizată pentru a deveni complet consistentă și fără erori, cu câteva intervenții manuale. Apoi, noua versiune `stable` este creată prin atribuirea numelui în cod al vechii arhive `testing` noii arhive `stable` și crearea unui nou nume în cod pentru noua arhivă `testing`. Conținutul inițial al noii arhive `testing` este exact același cu cel al arhivei `stable` recent lansate.

Atât arhiva `unstable` cât și arhiva `testing` pot suferi defecțiuni temporare din cauza mai multor factori.

- Încărcare pachet defect în arhivă (în special pentru `unstable`)
- Întârzierea acceptării noilor pachete în arhivă (în special pentru `unstable`)
- Problemă de sincronizare a arhivei (atât pentru `testing` cât și pentru `unstable`)
- Intervenția manuală asupra arhivei, cum ar fi eliminarea pachetelor (mai mult pentru `testing`) etc.

Deci, dacă vreodată decideți să utilizați aceste arhive, ar trebui să puteți remedia sau să rezolvați acest tip de probleme.

Atenție



Timp de aproximativ câteva luni după lansarea unei noi versiuni `stable` (stabilă), majoritatea utilizatorilor de stații de lucru ar trebui să utilizeze arhiva `stable` (stabilă) cu actualizările sale de securitate, chiar dacă de obicei utilizează arhivele `unstable` (instabilă) sau `testing` (în testare). În această perioadă de tranziție, arhivele `unstable` și `testing` nu sunt potrivite pentru majoritatea utilizatorilor. Este dificil să mențineți sistemul în stare bună de funcționare cu arhiva `unstable`, deoarece aceasta suferă actualizări majore pentru pachetele de bază. Arhiva `testing` nu este utilă nici ea, deoarece conține în mare parte același conținut ca arhiva `stable`, fără suportul de securitate ([Debian testing-security-announce 2008-12](#)). După aproximativ o lună, arhivele `unstable` sau `testing` pot deveni utile dacă aveți grijă.

Indicație

Când se urmărește arhiva `testing`, o problemă cauzată de un pachet eliminat este de obicei rezolvată prin instalarea pachetului corespunzător din arhiva `unstable`, care este încărcată pentru remediarea erorii.

Consultați [Manualul de politici Debian](#) pentru definițiile arhivelor.

- "Secțiuni"
- "Priorități"
- "Sistem de bază"
- "Pachete esențiale"

2.1.6 Debian este software 100% liber

Debian este software 100% liber datorită următoarelor aspecte:

- Debian instalează în mod implicit numai software liber, pentru a respecta libertățile utilizatorilor.
- Debian oferă numai software liber în `main`.
- Debian recomandă rularea numai a software-ului liber din `main`.
- Niciun pachet din `main` nu depinde și nu recomandă pachete din `non-free`, `non-free-firmware` sau `contrib`.

Unii oameni se întreabă dacă următoarele două fapte se contrazic sau nu.

- „Debian va rămâne 100% gratuit”. (Primul termen din [Contractul social Debian](#))
- Serverele Debian găzduiesc unele pachete `non-free-firmware`, `non-free` și `contrib`.

Acestea nu se contrazic, din următoarele motive.

- Sistemul Debian este 100% liber, iar pachetele sale sunt găzduite de serverele Debian în secțiunea `main`.
- Pachetele din afara sistemului Debian sunt găzduite de serverele Debian în zonele `non-free`, `non-free-firmware` și `contrib`.

Acestea sunt explicate în detaliu în termenii 4 și 5 din [Contractul social Debian](#):

- Prioritățile noastre sunt utilizatorii și software-ul liber
 - Ne vom ghida după nevoile utilizatorilor noștri și ale comunității de software liber. Vom pune interesele lor pe primul loc în lista noastră de priorități. Vom sprijini nevoile utilizatorilor noștri de a opera în multe tipuri diferite de medii informatice. Nu ne vom opune lucrărilor care nu sunt libere și care sunt destinate utilizării pe sistemele Debian și nu vom încerca să percepem taxe persoanelor care creează sau utilizează astfel de lucrări. Vom permite altora să creeze distribuții care conțin atât sistemul Debian, cât și alte lucrări, fără a percepe nicio taxă din partea noastră. În vederea promovării acestor obiective, vom furniza un sistem integrat de materiale de înaltă calitate, fără restricții legale care ar împiedica astfel de utilizări ale sistemului.
 - Lucrările care nu respectă standardele noastre privind software-ul liber
-

- Recunoaștem că unii dintre utilizatorii noștri necesită utilizarea unor lucrări care nu sunt conforme cu Ghidul Debian pentru software liber. Am creat secțiunile „non-free”, „non-free-firmware” și „contrib” în arhiva noastră pentru aceste lucrări. Pachetele din aceste secțiuni nu fac parte din sistemul Debian, deși au fost configurate pentru a fi utilizate cu Debian. Încurajăm producătorii de CD-uri să citească licențele pachetelor din aceste secțiuni și să determine dacă pot distribui pachetele pe CD-urile lor. Astfel, deși lucrările non-free nu fac parte din Debian, noi susținem utilizarea lor și oferim infrastructura necesară pentru pachetele non-free (cum ar fi sistemul nostru de urmărire a erorilor și listele de discuții). Suporturile media oficiale Debian pot include firmware care altfel nu face parte din sistemul Debian, pentru a permite utilizarea Debian cu hardware care necesită un astfel de firmware.

Notă

Textul actual al celui de-al cincilea termen din actualul [Contract social Debian](#) 1.2 este ușor diferit de textul de mai sus. Această abatere editorială este intenționată, pentru a asigura coerența acestui document destinat utilizatorilor, fără a modifica conținutul real al Contractului social.

Utilizatorii trebuie să fie conștienți de riscurile utilizării pachetelor din secțiunile non-free, non-free-firmware și contrib:

- lipsa de libertate pentru astfel de pachete software
- lipsa de asistență din partea Debian pentru astfel de pachete software (Debian nu poate oferi asistență adecvată pentru software fără a avea acces la codul sursă al acestuia)
- contaminarea sistemului Debian 100% liber

[Ghidul Debian pentru software-ul liber](#) reprezintă standardele de software liber pentru [Debian](#). Debian interpretează „software-ul” în sensul cel mai larg, incluzând documentele, firmware-ul, logo-ul și datele grafice din pachet. Acest lucru face ca standardele Debian pentru software-ul liber să fie foarte stricte.

Pachetele tipice non-free, non-free-firmware și contrib includ pachete distribuibile liber de următoarele tipuri:

- Pachete de documente sub [GNU Free Documentation License](#) cu secțiuni invariabile, cum ar fi cele pentru GCC și Make. (se găsesc în principal în secțiunea non-free/doc.)
- Pachete firmware care conțin date binare fără sursă, cum ar fi cele enumerate în Secțiune [9.10.5](#) ca non-free-firmware (se găsesc în principal în secțiunea non-free-firmware/kernel).
- Pachete de jocuri și fonturi cu restricții privind utilizarea comercială și/sau modificarea conținutului.

Vă rugăm să rețineți că numărul pachetelor non-free, non-free-firmware și contrib este mai mic de 2% din numărul pachetelor main. Activarea accesului la secțiunile non-free, non-free-firmware și contrib nu ascunde sursa pachetelor. Utilizarea interactivă pe ecran complet a aptitude(8) vă oferă vizibilitate și control deplin asupra pachetelor instalate din fiecare secțiune, pentru a vă menține sistemul atât de liber pe cât doriți.

2.1.7 Dependențele pachetelor

Sistemul Debian oferă un set consistent de pachete binare prin mecanismul său de declarare a dependențelor binare cu versiuni în câmpurile fișierului de control. Iată o definiție puțin simplificată pentru acestea.

- „Depends” (depinde de)
 - Aceasta declară o dependență absolută și toate pachetele enumerate în acest câmp trebuie instalate în același timp sau în prealabil.
 - „Pre-Depends” (pre-depinde de)
-

- Este similar cu Depends, cu excepția faptului că necesită instalarea completă în prealabil a pachetelor enumerate.
- „Recommends” (recomandate)
 - Aceasta declară o dependență puternică, dar nu absolută. Majoritatea utilizatorilor nu ar dori pachetul decât dacă toate pachetele enumerate în acest câmp sunt instalate.
- „Suggest” (sugerate)
 - Aceasta declară o dependență slabă. Mulți utilizatori ai acestui pachet pot beneficia de instalarea pachetelor enumerate în acest câmp, dar pot avea funcții rezonabile și fără acestea.
- „Enhances” (îmbunătățește)
 - Aceasta declară o dependență slabă, similară cu Suggests, dar funcționează în sens invers.
- „Breaks” (deteriorează)
 - Aceasta declară o incompatibilitate a pachetului, de obicei cu o anumită specificație de versiune. În general, soluția este să actualizați toate pachetele enumerate în acest câmp.
- „Conflicts” (conflicte)
 - Aceasta declară o incompatibilitate absolută. Toate pachetele enumerate în acest câmp trebuie eliminate pentru a instala acest pachet.
- „Replaces” (înlocuiește)
 - Aceasta se declară atunci când fișierele instalate de acest pachet înlocuiesc fișierele din pachetele enumerate.
- „Provides” (furnizează)
 - Acest lucru este declarat atunci când acest pachet furnizează toate fișierele și funcționalitățile din pachetele enumerate.

Notă

Vă rugăm să rețineți că definirea simultană a „Provides”, „Conflicts” și „Replaces” pentru un pachet virtual este configurația corectă. Aceasta asigură că doar un singur pachet real care furnizează acest pachet virtual poate fi instalat la un moment dat.

Definiția oficială, inclusiv dependența de sursă, poate fi găsită în [Manualul de politici: Capitolul 7 - Declararea relațiilor dintre pachete](#).

2.1.8 Fluxul evenimentelor din gestionarea pachetelor

Iată un rezumat al fluxului simplificat al evenimentelor din gestionarea pachetelor de către APT.

- **Actualizare metadata** („apt update”, „aptitude update” sau „apt-get update”):
 1. Preluarea metadatelor din arhiva de la distanță
 2. Reconstruirea și actualizarea metadatelor locale pentru utilizarea de către APT
 - **Modernizare (actualizarea la ultima versiune a pachetelor)** („apt upgrade” și „apt full-upgrade” sau „aptitude safe-upgrade” și „aptitude full-upgrade”, sau „apt-get upgrade” și „apt-get dist-upgrade”):
 1. Alegeți versiunea candidată, care este de obicei cea mai recentă versiune disponibilă pentru toate pachetele instalate (a se vedea Secțiune [2.7.7](#) pentru excepții)
 2. Rezolvarea dependențelor pachetelor
-

3. Preluarea pachetelor binare selectate din arhiva de la distanță dacă versiunea candidată este diferită de versiunea instalată
 4. Despachetarea pachetelor binare descărcate
 5. Rularea scriptului **preinst**
 6. Instalarea fișierelor binare
 7. Rularea scriptului **postinst**
- **Instalare** („`apt install ...`”, „`aptitude install ...`” sau „`apt-get install ...`”):
 1. Alegerea pachetelor listate în linia de comandă
 2. Rezolvarea dependențelor pachetelor
 3. Preluarea pachetelor binare selectate din arhiva de la distanță
 4. Despachetarea pachetelor binare descărcate
 5. Rularea scriptului **preinst**
 6. Instalarea fișierelor binare
 7. Rularea scriptului **postinst**
 - **Eliminare** („`apt remove ...`”, „`aptitude remove ...`” sau „`apt-get remove ...`”):
 1. Alegerea pachetelor listate în linia de comandă
 2. Rezolvarea dependențelor pachetelor
 3. Rularea scriptului **prerm**
 4. Eliminarea fișierelor instalate **cu excepția** fișierelor de configurare
 5. Rularea scriptului **postrm**
 - **Purjare (curățare)** („`apt purge`”, „`aptitude purge ...`” sau „`apt-get purge ...`”):
 1. Alegerea pachetelor listate în linia de comandă
 2. Rezolvarea dependențelor pachetelor
 3. Rularea scriptului **prerm**
 4. Eliminarea fișierelor instalate **inclusiv** fișierele de configurare
 5. Rularea scriptului **postrm**

Aici, am omis intenționat detaliile tehnice pentru a prezenta imaginea de ansamblu.

2.1.9 Primul răspuns la problemele legate de gestionarea pachetelor

Ar trebui să citiți documentația oficială detaliată. Primul document pe care trebuie să-l citiți este cel specific Debian „`/usr/share/doc/nume_pachet/README.Debian`”. Ar trebui consultată și altă documentație din „`/usr/share/doc/nume_pachet/`”. Dacă configurați shell-ul ca în Secțiune 1.4.2, tasteți următoarele.

```
$ cd package_name
$ pager README.Debian
$ mc
```

Pentru informații detaliate, poate fi necesar să instalați pachetul de documentație corespunzător, denumit cu sufixul „-doc”.

Dacă întâmpinați probleme cu un pachet specific, verificați mai întâi siturile [sistemului de urmărire a erorilor Debian \(BTS\)](#).

Căutați cu [Google](#) cu cuvinte cheie precum „`site:debian.org`”, „`site:wiki.debian.org`”, „`site:lists.debian.org`” etc.

Când raportați o eroare, vă rugăm să utilizați comanda `reportbug(1)`.

situl web	comanda
Pagina principală a sistemului de urmărire a erorilor Debian (BTS)	<code>sensible-browser "https://bugs.debian.org/"</code>
Raportul de eroare al unui pachet cunoscut	<code>sensible-browser "https://bugs.debian.org/package_name"</code>
Raportul de eroare al erorii cunoscute cu numărul	<code>sensible-browser "https://bugs.debian.org/bug_number"</code>

Tabela 2.5: Lista siturilor web importante pentru rezolvarea problemelor legate de un pachet specific

2.1.10 Cum să selectați pachetele Debian

Când întâlniți mai mult de două pachete similare și vă întrebați pe care să îl instalați fără a recurge la „încercări și erori”, ar trebui să folosiți **bunul simț**. Consider că următoarele puncte sunt indicii bune pentru alegerea pachetelor preferate.

- Esențial: da > nu
- Secțiunea: main > contrib > non-free
- Prioritate: necesar > important > standard > opțional > suplimentar
- Sarcini: pachete enumerate în sarcini precum „Mediu de birou”
- Pachetele selectate de pachetul de dependență (de exemplu, gcc-10 de către gcc)
- Popcon (popularitate): număr mai mare de voturi și instalări
- Istoric modificări: actualizări periodice de către responsabilul pachetului
- BTS: Fără erori RC (fără erori critice, grave sau serioase)
- BTS: responsabil de pachet receptiv la rapoartele de erori
- BTS: număr mai mare de erori remediate recent
- BTS: număr mai mic de erori rămase care nu sunt incluse ca listă de dorințe (wishlist)

Debian fiind un proiect de voluntariat cu un model de dezvoltare distribuit, arhiva sa conține multe pachete cu orientări și calități diferite. Trebuie să decideți singur ce să faceți cu ele.

2.1.11 Cum să faceți față cerințelor contradictorii

Indiferent de suita sistemului Debian pe care decideți să o utilizați, este posibil să doriți să rulați versiuni ale programelor care nu sunt disponibile în acea suită. Chiar dacă găsiți pachete binare ale acestor programe în alte suite Debian sau în alte resurse non-Debian, cerințele lor pot intra în conflict cu sistemul Debian actual.

Deși puteți modifica sistemul de gestionare a pachetelor cu tehnica **apt-pinning** etc., așa cum este descris în Secțiune 2.7.7 pentru a instala astfel de pachete binare nesincronizate, astfel de abordări de modificare au doar cazuri de utilizare limitate, deoarece pot afecta programele respective și sistemul dvs.

Înainte de a instala în mod brutal astfel de pachete nesincronizate, ar trebui să căutați toate soluțiile tehnice alternative mai sigure disponibile, care sunt compatibile cu sistemul Debian actual.

- Instalați astfel de programe utilizând pachetele binare corespunzătoare din amonte (vezi Secțiune 7.7).
 - Multe programe cu interfață grafică, precum LibreOffice și aplicațiile GNOME, sunt disponibile sub formă de pachete [Flatpak](#), [Snap](#) sau [AppImage](#).

- Creați un mediu chroot sau similar și rulați astfel de programe în acesta (consultați Secțiune 9.11).
 - Comenzile CLI pot fi executate cu ușurință în cadrul chroot compatibil (vedeți Secțiune 9.11.4).
 - Mai multe medii grafice de birou complete pot fi încercate cu ușurință fără a reporni sistemul (a se vedea Secțiune 9.11.5).
- Construiți singur versiunile dorite ale pachetelor binare care sunt compatibile cu sistemul Debian actual.
 - Aceasta este o **sarcină non-trivială** (a se vedea Secțiune 2.7.13).

2.2 Operații de bază de gestionare a pachetelor

Operațiile de gestionare a pachetelor bazate pe depozit în sistemul Debian pot fi efectuate de numeroase instrumente de gestionare a pachetelor bazate pe APT disponibile în sistemul Debian. Aici, explicăm 3 instrumente de bază pentru gestionarea pachetelor: `apt`, `apt-get` / `apt-cache` și `aptitude`.

Pentru operația de gestionare a pachetelor care implică instalarea pachetelor sau actualizarea metadatelor pachetelor, trebuie să aveți privilegii de root.

2.2.1 `apt` vs. `apt-get` / `apt-cache` vs. `aptitude`

Deși `aptitude` este un instrument interactiv foarte bun, pe care autorul îl folosește în principal, trebuie să știți câteva lucruri importante:

- Comanda `aptitude` nu este recomandată pentru actualizarea sistemului de la o versiune la alta pe sistemul Debian stabil după lansarea noii versiuni.
 - The use of "`apt full-upgrade`" or "`apt-get dist-upgrade`" is recommended for it. See [Bug #411280](#).
- Comanda `aptitude` sugerează uneori eliminarea în masă a pachetelor pentru actualizarea sistemului pe sistemul Debian testing sau unstable.
 - Această situație a speriat mulți administratori de sistem. Nu intrați în panică.
 - Aceasta pare să fie cauzată în principal de diferențele de versiune între pachetele dependente sau recomandate de un meta-pachet precum `gnome-core`.
 - Această problemă poate fi rezolvată selectând „Anulează acțiunile în așteptare” din meniul de comenzi `aptitude`, ieșind din `aptitude` și utilizând „`apt full-upgrade`”.

Comenzile `apt-get` și `apt-cache` sunt cele mai **elementare** instrumente de gestionare a pachetelor bazate pe APT.

- `apt-get` și `apt-cache` oferă doar interfața de utilizare a liniei de comandă.
- `apt-get` este cel mai potrivit pentru **actualizări majore ale sistemului** între versiuni etc.
- `apt-get` oferă un rezolvator de dependențe de pachete **robust**.
- `apt-get` solicită mai puține resurse hardware. Consumă mai puțină memorie și rulează mai rapid.
- `apt-cache` oferă o căutare **standard** bazată pe expresii regulate pentru numele și descrierea pachetului.
- `apt-get` și `apt-cache` pot gestiona mai multe versiuni de pachete folosind `/etc/apt/preferences`, dar este destul de complicat.

Comanda `apt` este o interfață de linie de comandă de nivel înalt pentru gestionarea pachetelor. Este practic un înveliș pentru `apt-get`, `apt-cache` și comenzi similare, concepută inițial ca interfață pentru utilizatorul final și care activează în mod implicit unele opțiuni mai potrivite pentru utilizarea interactivă.

- apt oferă o bară de progres intuitivă atunci când instalați pachete folosind `apt install`.
- apt va **elimina** pachetele `.deb` stocate în cache în mod implicit după instalarea cu succes a pachetelor descărcate.

Indicație

Utilizatorilor li se recomandă să utilizeze noua comandă `apt(8)` pentru utilizarea **interactivă** și să utilizeze comenzile `apt-get(8)` și `apt-cache(8)` într-un script shell.

Comanda `aptitude` este cel mai **versatil** instrument de gestionare a pachetelor bazat pe APT.

- `aptitude` oferă interfața text interactivă cu utilizatorul pe ecran complet.
- `aptitude` oferă și interfața de utilizare a liniei de comandă.
- `aptitude` este cel mai potrivit pentru **gestionarea interactivă zilnică a pachetelor**, cum ar fi inspectarea pachetelor instalate și căutarea pachetelor disponibile.
- `aptitude` este mai exigent în ceea ce privește resursele hardware. Consumă mai multă memorie și rulează mai lent.
- `aptitude` oferă o căutare **îmbunătățită** bazată pe expresii regulate pentru toate metadatele pachetului.
- `aptitude` poate gestiona mai multe versiuni ale pachetelor fără a utiliza `/etc/apt/preferences` și este destul de intuitiv.

2.2.2 Operații de bază de gestionare a pachetelor din linia de comandă

Iată operațiile de bază pentru gestionarea pachetelor din linia de comandă, folosind `apt(8)`, `aptitude(8)` și `apt-get(8)`/`apt-cache(8)`.

`apt` / `apt-get` și `aptitude` pot fi combinate fără probleme majore.

„`aptitude why rexpresie-regulată`” poate afișa mai multe informații prin „`aptitude -v why expresie-regulată`”. Informații similare pot fi obținute prin „`apt rdepends pachet`” sau „`apt-cache rdepends pachet`”.

Când comanda `aptitude` este pornită în modul linie de comandă și întâmpină unele probleme, cum ar fi conflicte între pachete, puteți trece la modul interactiv pe ecran complet apăsând tasta «e» în prompt.

Notă

Deși comanda `aptitude` vine cu funcții bogate, cum ar fi rezolvatorul de pachete îmbunătățit, această complexitate a cauzat (sau poate încă cauzează) unele regresii, cum ar fi [Eroarea #411123](#), [Eroarea #514930](#) și [Eroarea #570377](#). În caz de îndoială, vă rugăm să utilizați comenzile `apt`, `apt-get` și `apt-cache` în locul comenzii `aptitude`.

Puteți furniza opțiuni de comandă imediat după „`aptitude`”.

Pentru mai multe informații, consultați `aptitude(8)` și „Manualul utilizatorului `aptitude`” la „`/usr/share/doc/aptitude/R`”.

2.2.3 Utilizarea interactivă a «aptitudie»

Pentru gestionarea interactivă a pachetelor, porniți `aptitude` în modul interactiv din promptul consolei shell, după cum urmează.

```
$ sudo aptitude -u
Password:
```

sintaxa apt	sintaxa aptitude	sintaxa apt-get / apt-cache	descriere
apt update	aptitude update	apt-get update	actualizează metadatele arhivei pachetelor
apt install foo	aptitude install foo	apt-get install foo	instalează versiunea candidată a pachetului „foo” împreună cu dependențele sale
apt upgrade	aptitude safe-upgrade	apt-get upgrade	instalează versiunea candidată a pachetelor instalate fără a elimina alte pachete
apt full-upgrade	aptitude full-upgrade	apt-get dist-upgrade	instalați versiunea candidată a pachetelor instalate, eliminând unele pachete dacă este necesar
apt remove foo	aptitude remove foo	apt-get remove foo	eliminați pachetul „foo” lăsând în același timp fișierele sale de configurare
apt autoremove	N/D	apt-get autoremove	elimină pachetele instalate automat care nu mai sunt necesare
apt purge foo	aptitude purge foo	apt-get purge foo	șterge pachetul „foo” împreună cu fișierele sale de configurare
apt clean	aptitude clean	apt-get clean	șterge complet depozitul local de fișiere ale pachetelor preluate (descărcate)
apt autoclean	aptitude autoclean	apt-get autoclean	șterge din depozitul local fișierele pachetelor descărcate pentru care pachetele învechite
apt show foo	aptitude show foo	apt-cache show foo	afișează informații detaliate despre pachetul „foo”
apt search <i>expresie-regulată</i>	aptitude search <i>expresie-regulată</i>	apt-cache search <i>expresie-regulată</i>	căută pachetele care corespund cu <i>expresia-regulată</i>
N/D	aptitude why <i>expresie-regulată</i>	N/D	explică motivul pentru care ar trebui instalate pachetele care corespund cu <i>expresia-regulată</i>
N/D	aptitude why-not <i>expresie-regulată</i>	N/D	explică motivul pentru care pachetele care corespund cu <i>expresia-regulată</i> nu pot fi instalate
apt list --manual-installed	aptitude search '~i!~M'	apt-mark showmanual	listează pachetele instalate manual

Tabela 2.6: Operații de bază de gestionare a pachetelor din linia de comandă folosind apt(8), aptitude(8) și apt-get(8) / apt-cache(8)

opțiunea de comandă	descriere
-s	simulează rezultatul comenzii
-d	doar descărcare, fără instalare/actualizare
-D	afișează explicații succinte înainte de instalările și deinstalările automate

Tabela 2.7: Opțiuni de comandă demne de menționat pentru aptitude(8)

Aceasta actualizează copia locală a informațiilor din arhivă și afișează lista pachetelor pe ecranul complet cu meniu. Aptitude plasează configurația sa în „~/ .aptitude/config”.

Indicație

Dacă doriți să utilizați configurația root în locul celei a utilizatorului, utilizați „sudo -H aptitude ...” în locul „sudo aptitude ...” în expresia de mai sus.

Indicație

Aptitude stabilește automat **acțiunile în așteptare** atunci când este pornit interactiv. Dacă nu vă place, puteți să-l reinițializați din meniu: „Acțiune” → „Anulează acțiunile în așteptare”.

2.2.4 Combinații de taste pentru aptitude

Tastele demne de menționat pentru a naviga în starea pachetelor și pentru a defini „acțiunea planificată” pentru acestea în acest mod ecran complet sunt următoarele.

tasta	comanda/funcția asociată
F10 sau Ctrl-t	meniu
?	afișează ajutorul pentru apăsarea tastei (listă mai completă)
F10 → Ajutor → Manualul utilizatorului	Afișează manualul utilizatorului
u	actualizează informațiile referitoare la arhiva pachetelor
+	marchează pachetul pentru actualizare sau instalare
-	marcați pachetul pentru eliminare (păstrează fișierele de configurare)
—	marchează pachetul pentru ștergere (elimină fișierele de configurare)
=	pune pachetul în așteptare
U	marchează toate pachetele care pot fi actualizate (funcționează ca full-upgrade)
g	începe descărcarea și instalarea pachetelor selectate
q	iese din ecranul curent și salvează modificările
x	iese din ecranul curent și înlătură modificările
Enter	afișează informații despre un pachet
C	afișează jurnalul de modificări al unui pachet
l	modifică limita pentru pachetele afișate
/	caută prima potrivire
\	repetă ultima căutare

Tabela 2.8: Lista combinațiilor de taste pentru «aptitude»

Specificația numelui fișierului din linia de comandă și promptul meniului după apăsarea „l” și „/” utilizează expresia regulată aptitude, așa cum este descrisă mai jos. Expresia regulată aptitude poate potrivi în mod explicit un nume de pachet utilizând un șir care începe cu „~n” și este urmat de numele pachetului.

Indicație

Trebuie să apăsați tasta „U” pentru a actualiza toate pachetele instalate la **versiunea candidată** în interfața vizuală. În caz contrar, numai pachetele selectate și anumite pachete cu dependențe de versiune față de acestea vor fi actualizate la **versiunea candidată**.

2.2.5 Vizualizarea de pachete în aptitude

În modul interactiv pe ecran complet al `aptitude(8)`, pachetele din lista de pachete sunt afișate conform exemplului următor.

```
idA    libsmclient                -2220kB 3.0.25a-1 3.0.25a-2
```

Aici, această linie înseamnă de la stânga către dreapta, după cum urmează:

- Indicatorul „starea curentă” (prima literă)
- Indicatorul „acțiunea planificată” (a doua literă)
- Indicatorul „automat” (a treia literă)
- Numele pachetului
- Modificarea utilizării spațiului pe disc atribuită „acțiunii planificate”
- Versiunea actuală a pachetului
- Versiunea candidată a pachetului

Indicație

Lista completă a indicatorilor este afișată în partea de jos a ecranului **Ajutor**, care apare apăsând «?».

Versiunea candidată este aleasă în funcție de preferințele locale curente (vezi `apt_preferences(5)` și Secțiune 2.7.7).

Mai multe tipuri de vizualizări ale pachetelor sunt disponibile în meniul „Vizualizări”.

vizualizarea	descrierea vizualizării
Vizualizare pachet	vedeți Tabel 2.10 (implicit)
Recomandări de auditare	lista pachetelor recomandate de unele pachete instalate, dar care nu sunt încă instalate
Lista plană a pachetelor	lista pachetelor fără categorizare (pentru utilizare cu expresii regulate)
Navigator debtags	lista pachetelor clasificate în funcție de intrările lor debtags
Vizualizare pachet sursă	lista pachetelor grupate după pachetele sursă

Tabela 2.9: Lista vizualizărilor pentru aptitude

Notă

Vă rugăm să ne ajutați [să îmbunătățim etichetarea pachetelor cu debtags!](#)

Standardul „Vizualizare pachete” clasifică pachetele într-un mod similar cu `dselect`, cu câteva caracteristici suplimentare.

Indicație

Vizualizarea Sarcini poate fi utilizată pentru a selecta pachetele potrivite pentru sarcina dvs.

categoria	descrierea vizualizării
Pachete actualizabile	lista pachetele organizate ca secțiune → zonă → pachet
Pachete noi	, ,
Pachete instalate	, ,
Pachete neinstalate	, ,
Pachete învechite sau create local	, ,
Pachete virtuale	lista pachetelor cu aceeași funcție
Sarcini	lista pachetelor cu diferite funcții necesare în general pentru o sarcină

Tabela 2.10: Clasificarea vizualizărilor standard ale pachetelor

2.2.6 Opțiuni pentru metoda de căutare cu aptitude

Aptitude oferă mai multe opțiuni pentru căutarea pachetelor folosind formula de expresie regulată.

- Linia de comandă shell:
 - «`aptitude search 'expresie-regulată_aptitude'`» pentru a afișa starea instalării, numele pachetului și o scurtă descriere a pachetelor care corespund căutării
 - «`aptitude show 'nume_pachet'`» pentru a afișa descrierea detaliată a pachetului
- Modul interactiv pe ecran complet:
 - „l” pentru a limita vizualizarea pachetelor la pachetele care se potrivesc căutării
 - „/” pentru căutarea unui pachet care corespunde modelului de căutare
 - „\” pentru căutarea înapoi a unui pachet care corespunde modelului de căutare
 - „n” pentru căutarea următoarei coincidențe
 - „N” pentru căutarea următoarei coincidențe (înapoi)

Indicație

Șirul pentru `nume_pachet` este tratat ca fiind exact identic cu numele pachetului, cu excepția cazului în care începe în mod explicit cu „~” pentru a fi considerat formulă de expresie regulată.

2.2.7 Formula expresiei regulate pentru aptitudine

Formula expresiei regulate aptitude este similară cu cea extinsă mutt **ERE** (a se vedea Secțiune 1.6.2), iar semnificațiile extensiilor speciale ale regulilor de potrivire specifice aptitude sunt următoarele.

- Partea expresie-regulată este aceeași **ERE** ca cea utilizată în instrumentele text tipice de tip Unix care utilizează „^”, „.”, „*”, „\$” etc., ca în `egrep(1)`, `awk(1)` și `perl(1)`.
- *tipul* de dependență este unul dintre (depends, predepends, recommends, suggests, conflicts, replaces, provides) care specifică interrelația dintre pachete.
- *Tipul* implicit de dependență este „depends”.

Indicație

Când `modelul_expresie-regulată` este un șir nul, plasați „~T” imediat după comandă.

descrierea regulii extinse de potrivire	formula expresiei regulate
potrivire cu numele pachetului	<code>~nexpresie-regulată_nume</code>
potrivire cu descrierea	<code>~dexpresie-regulată_descriere</code>
potrivire cu numele sarcinii	<code>~texpresie-regulată_sarcină</code>
potrivire cu debtag	<code>~Gexpresie-regulată_debtag</code>
potrivire cu responsabilul pachetului	<code>~mexpresie-regulată_responsabil</code>
potrivire cu secțiunea pachetului	<code>~sexpresie-regulată_secțiune</code>
potrivire cu versiunea pachetului	<code>~Vexpresie-regulată_versiune</code>
potrivire cu distribuția	<code>~A{trixie,forky,sid}</code>
potrivire cu originea	<code>~O{debian,...}</code>
coincidentă prioritate	<code>~p{extra,important,optional,required,standard}</code>
corespunde cu pachete esențiale	<code>~E</code>
corespunde cu pachete virtuale	<code>~V</code>
corespunde cu pachete noi	<code>~N</code>
potrivire cu acțiunea în așteptare	<code>~a{install,upgrade,downgrade,remove,purge,hold,keep}</code>
potrivire cu pachete instalate	<code>~i</code>
corespunde pachetelor instalate cu marca A (pachete instalate automat)	<code>~M</code>
corespunde pachetelor instalate fără marca A (pachete selectate de administrator)	<code>~i!~M</code>
potrivire cu pachete instalate și actualizabile	<code>~U</code>
potrivire cu pachete eliminate, dar nu șterse	<code>~C</code>
potrivire cu pachete eliminate, șterse sau care pot fi eliminate	<code>~g</code>
potrivire cu pachete care declară o dependență ruptă	<code>~b</code>
potrivire cu pachete care declară o dependență ruptă de <i>tipul</i>	<code>~Btipul</code>
corespunde cu <i>modelul</i> de pachete care declară o dependență de <i>tipul</i>	<code>~D[tipul:]modelul</code>
corespunde cu <i>modelul</i> de pachete care declară o dependență ruptă de <i>tipul</i>	<code>~DB[tipul:]modelul</code>
corespunde cu pachetele pentru care <i>modelul</i> de potrivire corespunde pachetelor care declară o dependență de <i>tipul</i>	<code>~R[tipul:]modelul</code>
corespunde cu pachetele pentru care <i>modelul</i> de potrivire corespunde pachetelor care declară o dependență ruptă de <i>tipul</i>	<code>~RB[tipul:]modlul</code>
corespunde cu pachetele de care depind alte pachete instalate	<code>~R~i</code>
corespunde cu pachetele de care nu depind alte pachete instalate	<code>!~R~i</code>
corespunde cu pachetele de care depind sau pe care le recomandă alte pachete instalate	<code>~R~i ~Rrecommends:~i</code>
potrivire cu <i>modelul</i> de pachet filtrat după versiune	<code>~S filtru-versiune model</code>
corespunde cu toate pachetele (adevărat)	<code>~T</code>
nu corespunde cu niciun pachet (fals)	<code>~F</code>

Tabela 2.11: Lista formulelor de expresii regulate pentru aptitude

Iată câteva scurtături.

- „~Ptermen” == „~Dprovides:termen”
- „~Ctermen” == „~Dconflicts:termen”
- „...~W termen” == „(...|termen)”

Utilizatorii familiarizați cu `mutt` învață repede, deoarece `mutt` a fost sursa de inspirație pentru sintaxa expresiilor. Consultați „CĂUTARE, LIMITARE ȘI EXPRESII - (SEARCHING, LIMITING, AND EXPRESSIONS)” în „Manualul utilizatorului” „/usr/share/doc/aptitude/README”.

Notă

Cu versiunea lenny a `aptitude(8)`, noua sintaxă de **formă lungă**, cum ar fi „?broken”, poate fi utilizată pentru potrivirea expresiilor regulate în locul vechii sale echivalente de **formă scurtă** „~b”. Acum, caracterul spațiu „ ” este considerat unul dintre caracterele de terminare a expresiilor regulate, pe lângă caracterul tildă „~”. Consultați „Manualul utilizatorului” pentru noua sintaxă de **formă lungă**.

2.2.8 Rezolvarea dependențelor de către aptitude

Selectarea unui pachet în `aptitude` nu numai că extrage pachetele definite în lista „Depends:”, ci și pe cele definite în lista „Recommends:”, dacă meniul „F10 → Opțiuni → Preferințe → Gestionarea dependențelor” este configurat corespunzător. Aceste pachete instalate automat sunt eliminate automat dacă nu mai sunt necesare în `aptitude`.

Indicatorul care controlează comportamentul „instalării automate” al comenzii `aptitude` poate fi, de asemenea, manipulat folosind comanda `apt-mark(8)` din pachetul `apt`.

2.2.9 Jurnale de activitate ale pachetelor

Puteți verifica istoricul activității pachetului în fișierele jurnal.

fișier	conținut
/var/log/dpkg.log	Jurnalul activității la nivel dpkg pentru activitățile tuturor pachetelor
/var/log/apt/term.log	Jurnalul activității APT generice
/var/log/aptitude	Jurnalul activității comenzii <code>aptitude</code>

Tabela 2.12: Fișierele jurnal pentru activitățile pachetului

În realitate, nu este atât de ușor să obții rapid informații relevante din aceste jurnale. Vedeți Secțiune 9.3.9 pentru o metodă mai simplă.

2.3 Exemple de operații cu aptitudine

Iată câteva exemple de operații cu `aptitude(8)`.

2.3.1 Căutarea de pachete interesante

Puteți căuta pachete care să vă satisfacă nevoile cu ajutorul `aptitude` din descrierea pachetului sau din lista aflată sub „Sarcini”.

2.3.2 Listarea pachetelor care se potrivesc cu expresia regulată în numele pachetelor

Următoarea comandă listează pachetele cu potriviri de expresie regulată în numele pachetelor.

```
$ aptitude search '~n(pam|nss).*ldap'
p libnss-ldap - NSS module for using LDAP as a naming service
p libpam-ldap - Pluggable Authentication Module allowing LDAP interfaces
```

Acest lucru este foarte util pentru a găsi numele exact al unui pachet.

2.3.3 Navigarea prin rezultatul potrivirilor expresiei regulate

Expresia regulată „~dipv6” din vizualizarea „Listă nouă plană de pachete” cu promptul „l” limitează vizualizarea la pachetele cu descrierea care coincide cu expresia și vă permite să răsfoiți informațiile acestora în mod interactiv.

2.3.4 Stergerea definitivă a pachetelor eliminate

Puteți șterge toate fișierele de configurare rămase ale pachetelor eliminate.

Verificați rezultatele următoarei comenzi.

```
# aptitude search '~c'
```

Dacă considerați că pachetele listate pot fi șterse, executați următoarea comandă.

```
# aptitude purge '~c'
```

Puteți face același lucru în modul interactiv pentru un control mai precis.

Introduceți expresia regulată „~c” în vizualizarea «Vizualizare pachet nou» cu promptul „l”. Aceasta limitează vizualizarea pachetului numai la pachetele care corespund expresiei regulate, adică „eliminate, dar nu șterse”. Toate aceste pachete care corespund expresiei regulate pot fi afișate apăsând „[” în titlurile de nivel superior.

Apoi apăsați „_” în titlurile de nivel superior, cum ar fi „Pachete neinstalate”. Numai pachetele care corespund expresiei regulate din titlu sunt marcate pentru a fi șterse. Puteți exclude unele pachete de la ștergere apăsând „=” în mod interactiv pentru fiecare dintre ele.

Această tehnică este destul de utilă și funcționează pentru multe alte taste de comandă.

2.3.5 Reorganizarea stării instalării automate/manuale

Iată cum ordonez starea instalării automate/manuale pentru pachete (după utilizarea unui program de instalare a pachetelor non-aptitude etc.).

1. Lansați aptitude în modul interactiv ca root.
2. Tastați „u”, „U”, „f” și „g” pentru a actualiza și a îmbunătăți lista de pachete și pachetele.
3. Tastați „l” pentru a introduce limita de afișare a pachetelor ca „~i(~R~i|~Rrecommends:~i)” și tastați „M” peste „Pachete instalate” ca instalate automat.
4. Tastați „l” pentru a introduce limita de afișare a pachetelor ca „~prequired|~pimportant|~pstandard|~E” și tastați „m” peste „Pachete instalate” ca instalate manual.
5. Tastați „l” pentru a introduce limita de afișare a pachetelor ca „~i!~M” și eliminați pachetele neutilizate tastând „-” peste fiecare dintre ele după ce le-ați expus tastând „[” peste „Pachete instalate”.

6. Tastați „l” pentru a introduce limita de afișare a pachetelor ca „~i”; apoi tastați „m” peste „Sarcini” pentru a marca pachetele respective ca fiind instalate manual.
7. Ieșiți din aptitude.
8. Lansați „apt-get -s autoremove | less” ca root pentru a verifica ce nu este utilizat.
9. Reporniți aptitude în modul interactiv și marcați pachetele necesare ca „m”.
10. Reporniți „apt-get -s autoremove | less” ca root pentru a verifica din nou că REMOVED conține numai pachetele așteptate.
11. Porniți „apt-get autoremove | less” ca root pentru a elimina automat pachetele neutilizate.

Acțiunea „m” asupra „Tasks” este opțională, pentru a preveni situații de eliminare în masă a pachetelor în viitor.

2.3.6 Actualizare la nivel de sistem

Notă

Când treceți la o nouă versiune etc., ar trebui să luați în considerare efectuarea unei instalări curate a noului sistem, chiar dacă Debian poate fi actualizat așa cum este descris mai jos. Acest lucru vă oferă șansa de a elimina fișierele inutile acumulate și vă expune la cea mai bună combinație de pachete recente. Desigur, ar trebui să faceți o copie de rezervă completă a sistemului într-un loc sigur (consultați Secțiune 10.2) înainte de a face acest lucru. Vă recomand să configurați un sistem de pornire duală utilizând partiții diferite pentru a avea o tranziție cât mai lină.

Puteți efectua o actualizare la nivel de sistem la o versiune mai nouă modificând conținutul **listei de surse** pentru a indica o nouă versiune și executând comanda „apt update; apt dist-upgrade”.

Pentru a actualiza de la `stable` la `testing` sau `unstable` în timpul ciclului de lansare `trixie-ca-stable`, înlocuiți „trixie” din exemplul **listei sursă** din Secțiune 2.1.5 cu „forky” sau „sid”.

În realitate, este posibil să întâmpinați unele complicații din cauza unor probleme legate de tranziția pachetelor, în principal din cauza dependențelor pachetelor. Cu cât diferența dintre versiuni este mai mare, cu atât este mai probabil să întâmpinați probleme mai mari. Pentru tranziția de la vechiul `stable` la noul `stable` după lansarea acestuia, puteți citi noile [Note de lansare](#) și să urmați procedura exactă descrisă în acestea pentru a minimiza problemele.

Când decideți să treceți de la `stable` la `testing` înainte de lansarea oficială, nu există [Note de lansare](#) care să vă ajute. Diferența dintre `stable` și `testing` ar putea fi destul de mare după lansarea anterioară a `stable`, ceea ce complică situația actualizării.

Ar trebui să luați măsuri de precauție pentru actualizarea completă, în timp ce colectați cele mai recente informații din lista de distribuție și folosiți bunul simț.

1. Citiți „Notele de lansare” anterioare.
 2. Faceți o copie de rezervă a întregului sistem (în special a datelor și informațiilor de configurare).
 3. Aveți la îndemână un suport de pornire pentru cazul în care se strică încărcătorul de pornire.
 4. Informați utilizatorii cu privire la sistem cu suficient timp înainte.
 5. Înregistrați activitatea de actualizare cu `script(1)`.
 6. Aplicați „unmarkauto” pachetelor necesare, de exemplu, „`aptitude unmarkauto vim`”, pentru a preveni eliminarea.
 7. Reduceți la minimum pachetele instalate pentru a diminua riscul de conflicte între pachete, de exemplu, eliminați pachetele de sarcini de medii grafice de birou.
 8. Eliminați fișierul „`/etc/apt/preferences`” (dezactivați **apt-pinning**).
-

9. Încercați să faceți actualizarea pas cu pas: `oldstable` → `stable` → `testing` → `unstable`.
10. Actualizați **lista surselor** pentru a indica numai noua arhivă și rulați „`aptitude update`”.
11. Instalați, opțional, mai întâi noile **pachete de bază**, de exemplu, «`aptitude install perl`».
12. Rulați comanda «`apt-get -s dist-upgrade`» pentru a evalua impactul.
13. În cele din urmă, executați comanda «`apt-get dist-upgrade`».

**Atenție**

Nu este recomandat să săriți peste versiunile majore Debian atunci când faceți actualizări între versiunile `stable` (stabile).

**Atenție**

În „Notele de lansare” anterioare, GCC, Linux Kernel, `initrd-tools`, Glibc, Perl, lanțul de instrumente APT etc. au necesitat o atenție specială pentru actualizarea la nivel de sistem.

Pentru actualizări zilnice în `unstable`, consultați Secțiune [2.4.3](#).

2.4 Operații avansate de gestionare a pachetelor

2.4.1 Operații avansate de gestionare a pachetelor din linia de comandă

Iată o listă cu alte operații de gestionare a pachetelor pentru care `aptitude` este prea complex sau nu dispune de funcționalitățile necesare.

Notă

Pentru un pachet cu caracteristica [multi-arch](#), poate fi necesar să specificați numele arhitecturii pentru unele comenzi. De exemplu, utilizați «`dpkg -L libglib2.0-0:amd64`» pentru a afișa conținutul pachetului `libglib2.0-0` pentru arhitectura `amd64`.

**Atenție**

Instrumentele de nivel inferior, precum „`dpkg -i ...`” și „`debi ...`”, trebuie utilizate cu precauție de către administratorul de sistem. Acestea nu se ocupă automat de dependențele necesare ale pachetelor. Opțiunile liniei de comandă `Dpkg „- -force-all”` și similare (vezi `dpkg(1)`) sunt destinate utilizării numai de către experți. Utilizarea acestora fără a înțelege pe deplin efectele lor poate duce la deteriorarea întregului sistem.

Vă rugăm să rețineți următoarele:

- Toate comenzile de configurare și instalare a sistemului trebuie rulate din contul `root`.
 - Spre deosebire de `aptitude`, care utilizează expresii regulate (a se vedea Secțiune [1.6.2](#)), alte comenzi de gestionare a pachetelor utilizează modele precum modelele globale de shell „shell glob” (a se vedea Secțiune [1.5.6](#)).
 - `apt-file(1)` furnizat de pachetul `apt-file` trebuie să execute în prealabil comanda «`apt-file update`».
 - `configure-debian(8)` furnizat de pachetul `configure-debian` rulează `dpkg-reconfigure(8)` ca motor.
-

comanda	acțiunea
<code>COLUMNS=120 dpkg -l nume-pachet</code>	afișează starea unui pachet instalat pentru raportul de eroare
<code>dpkg -L nume-pachet</code>	afișează conținutul unui pachet instalat
<code>dpkg -L nume-pachet egrep '/usr/share/man/man.*/.+'</code>	listează paginile de manual pentru un pachet instalat
<code>dpkg -S nume-pachet</code>	listează pachetele instalate care au numele de fișier ce coincide cu modelul
<code>apt-file search nume-pachet</code>	listează pachetele din arhivă care au numele de fișier ce coincide cu modelul
<code>apt-file list nume-pachet</code>	listează conținutul pachetelor din arhivă ce coincide cu modelul
<code>dpkg-reconfigure nume-pachet</code>	reconfigurează pachetul specificat
<code>dpkg-reconfigure -p low nume-pachet</code>	reconfigurează pachetul specificat cu nivelul de chestionare cel mai detaliat (ridicat)
<code>configure-debian</code>	reconfigurează pachetele selectate din meniul pe ecran complet
<code>dpkg --audit</code>	sistemul de auditare pentru pachete instalate parțial
<code>dpkg --configure -a</code>	configurează toate pachetele instalate parțial
<code>apt-cache policy nume-pachet-bin</code>	afișează versiunea disponibilă, prioritatea și informațiile de arhivă ale unui pachet binar
<code>apt-cache madison nume-pachet</code>	afișează versiunea disponibilă, informații privind arhiva unui pachet
<code>apt-cache showsrc nume-pachet-bin</code>	afișează informațiile despre pachetul sursă al unui pachet binar
<code>apt-get build-dep nume-pachet</code>	instalează pachete necesare pentru a construi pachetul
<code>aptitude build-dep nume-pachet</code>	instalează pachete necesare pentru a construi pachetul
<code>apt-get source nume-pachet</code>	descarcă pachetul sursă (din arhiva standard)
<code>dget adresa URL pentru fișierul dsc</code>	descarcă pachetele sursă (din altă arhivă)
<code>dpkg-source -x nume-pachet_versiunea-revizia-debian dsc</code>	construiește un arbore sursă dintr-un set de pachete sursă („dsc”, „*.tar.gz” și „*.debian.tar.gz”, „*.diff.gz”)
<code>debuild binary</code>	construiește pachete dintr-un arbore sursă local
<code>make-kpkg kernel_image</code>	construiește un pachet de nucleu dintr-un arbore sursă de nucleu
<code>make-kpkg --initrd kernel_image</code>	construiește un pachet de nucleu dintr-un arbore sursă de nucleu cu initramfs activat
<code>dpkg -i nume-pachet_versiunea-revizia-debian arhitectura.deb</code>	instalează un pachet local în sistem
<code>apt install /ruta/la/nume-fișier_pachet.deb</code>	instalează un pachet local în sistem, încercând în același timp să rezolve automat dependențele
<code>debi nume-pachet_versiunea-revizia-debian arhitectura.dsc</code>	instalează pachetul (pachetele) local(e) în sistem
<code>dpkg --get-selections '*'> selection.txt</code>	salvează informații despre starea selecției pachetului la nivel dpkg
<code>dpkg --set-selections <selection.txt</code>	definește informațiile privind starea selecției pachetelor la nivel dpkg
<code>echo nume-pachet hold dpkg --set-selections</code>	definește starea de selecție a pachetului la nivel dpkg pentru un pachet la hold (echivalent cu „aptitude hold nume_pachet”)

Tabela 2.13: Lista operațiilor avansate de gestionare a pachetelor

- `dpkg-reconfigure(8)` rulează scripturi de pachete folosind `debconf(1)` ca motor.
- Comenzile «`apt-get build-dep`», «`apt-get source`» și «`apt-cache showsrc`» necesită introducerea unei intrări „`deb-src`” în **lista surselor**.
- `dget(1)`, `debuild(1)` și `debi(1)` necesită pachetul `devscripts`.
- Consultați procedura de (re)împachetare folosind «`apt-get source`» în Secțiune 2.7.13.
- Comanda `make-kpkg` necesită pachetul `kernel-package` (a se vedea Secțiune 9.10).
- Consultați Secțiune 12.9 pentru informații generale despre împachetare.

2.4.2 Verificarea fișierelor pachetului instalat

Instalarea `debsums` permite verificarea fișierelor pachetelor instalate în raport cu valorile MD5sum din fișierul „`/var/lib/dp`” cu `debsums(1)`. Consultați Secțiune 10.3.5 pentru a afla cum funcționează MD5sum.

Notă

Deoarece baza de date MD5sum poate fi modificată de către intrus, `debsums(1)` are o utilitate limitată ca instrument de securitate. Este util doar pentru verificarea modificărilor locale efectuate de administrator sau a daunelor cauzate de erori ale suportului media.

2.4.3 Protejarea împotriva problemelor legate de pachete

Mulți utilizatori preferă să urmeze versiunile **de testare** „testing” (sau **instabile** „unstable”) ale sistemului Debian pentru noile sale caracteristici și pachete. Acest lucru face ca sistemul să fie mai predispus la erori critice ale pachetelor.

Instalarea pachetului `apt-listbugs` vă protejează sistemul împotriva erorilor critice, verificând automat Debian BTS pentru erori critice atunci când actualizați cu sistemul APT.

Instalarea pachetului `apt-listchanges` oferă știri importante din „NEWS.Debian” atunci când se efectuează actualizarea cu sistemul APT.

2.4.4 Căutarea în metadatele pachetului

Deși vizitarea sitului Debian <https://packages.debian.org/> facilitează în prezent căutarea ușoară a metadatelor pachetelor, să analizăm câteva metode mai tradiționale.

Comenzile `grep-dctrl(1)`, `grep-status(1)` și `grep-available(1)` pot fi utilizate pentru a căuta orice fișier care are formatul general al unui fișier de control al pachetului Debian.

„`dpkg -S model_nume-fișier`” poate fi utilizat pentru a căuta numele pachetelor care conțin fișiere cu numele corespunzător instalate de `dpkg`. Dar acest lucru ignoră fișierele create de scripturile administratorului.

Dacă trebuie să efectuați o căutare mai detaliată în metadatele `dpkg`, trebuie să rulați comanda «`grep -e model_expressi*`» în directorul „`/var/lib/dpkg/info/`”. Astfel, veți căuta cuvintele menționate în scripturile pachetului și în textele de interogare ale instalării.

Dacă doriți să căutați recursiv dependențele pachetelor, trebuie să utilizați `apt-rdepends(8)`.

2.5 Detalii interne privind gestionarea pachetelor Debian

Să învățăm cum funcționează sistemul de gestionare a pachetelor Debian la nivel intern. Acest lucru ar trebui să vă ajute să creați propria soluție pentru unele probleme legate de pachete.

2.5.1 Metadate de arhivă

Fișierele cu metadate pentru fiecare distribuție sunt stocate în „dist/codename” pe fiecare sit oglindă Debian, de exemplu „<http://deb.debian.org/debian/>”. Structura arhivei poate fi răsfoită cu ajutorul navigatorului web. Există 6 tipuri de metadate cheie.

fișier	locăție	conținut
Release	partea superioară a distribuției	descrierea arhivei și informații privind integritatea
Release.gpg	partea superioară a distribuției	fișierul de semnătură pentru fișierul „Release” semnat cu cheia arhivei
Contents- <i>arhitectura</i>	partea superioară a distribuției	lista tuturor fișierelor pentru toate pachetele din arhiva relevantă
Release	partea superioară a fiecărei combinații distribuție/secțiune/arhitectură	descrierea arhivei utilizată pentru regula apt_preferences(5)
Packages	partea superioară a fiecărei combinații distribuție/secțiune/arhitectură binară	debian/control concatenat pentru pachetele binare
Sources	partea superioară a fiecărei combinații distribuție/secțiune/sursă	debian/control concatenat pentru pachetele sursă

Tabela 2.14: Conținutul metadatelor arhivei Debian

În arhiva recentă, aceste metadate sunt stocate sub formă de fișiere comprimate și diferențiale pentru a reduce traficul de rețea.

2.5.2 Fișierul „Release” de nivel superior și autenticitatea

Indicație

Fișierul de nivel superior „Release” este utilizat pentru semnarea arhivei în cadrul sistemului **APT securizat**.

Fiecare suită din arhiva Debian are un fișier de nivel superior „Release”, de exemplu „<http://deb.debian.org/debian/>”, după cum urmează.

```
Origin: Debian
Label: Debian
Suite: unstable
Codename: sid
Date: Sat, 14 May 2011 08:20:50 UTC
Valid-Until: Sat, 21 May 2011 08:20:50 UTC
Architectures: alpha amd64 armel hppa hurd-i386 i386 ia64 kfreebsd-amd64 kfreebsd-i386 mips ←
               mipsel powerpc s390 sparc
Components: main contrib non-free
Description: Debian x.y Unstable - Not Released
MD5Sum:
  bdc8fa4b3f5e4a715dd0d56d176fc789 18876880 Contents-alpha.gz
  9469a03c94b85e010d116aeeab9614c0 19441880 Contents-amd64.gz
  3d68e206d7faa3aded660dc0996054fe 19203165 Contents-armel.gz
...
```

Notă

Aici puteți găsi motivele pentru care folosesc termenii „suită” și „nume în cod” în Secțiune 2.1.5. Termenul „distribuție” este folosit atunci când se face referire atât la „suită”, cât și la „numele în cod”. Toate numele „secțiunilor” oferite de arhivă sunt listate la „Componente”.

Integritatea fișierului de nivel superior „Re lease” este verificată de infrastructura criptografică numită [secure apt](#) (apt securizat), așa cum este descrisă în `apt - secure(8)`.

- Fișierul cu semnătura criptografică „Re lease . gpg” este creat din fișierul autentic de nivel superior „Re lease” și cheia secretă a arhivei Debian.
- Cheile publice ale arhivei Debian sunt instalate local de cel mai recent pachet `debian - archive - keyring`.
- Sistemul **secure APT** verifică automat integritatea fișierului de nivel superior „Re lease” descărcat, prin criptare, cu ajutorul fișierului „Re lease . gpg” și al cheilor publice ale arhivei Debian instalate local.
- Integritatea tuturor fișierelor „Packages” și „Sources” este verificată utilizând valorile MD5sum din fișierul de nivel superior „Re lease”. Integritatea tuturor fișierelor pachetului este verificată utilizând valorile MD5sum din fișierele „Packages” și „Sources”. Consultați `debsums(1)` și Secțiune 2.4.2.
- Deoarece verificarea semnăturii criptografice este un proces mult mai intens pentru CPU decât calcularea valorii MD5sum, utilizarea valorii MD5sum pentru fiecare pachet în timp ce se utilizează semnătura criptografică pentru fișierul de nivel superior „Re lease” oferă [o bună securitate cu performanță](#) (a se vedea Secțiune 10.3).

Dacă intrarea **listei sursă** specifică opțiunea „signed - by”, integritatea fișierului „Re lease” de nivel superior descărcat este verificată folosind cheia publică specificată. Acest lucru este util atunci când **lista sursă** conține arhive non-Debian.

Indicație

Utilizarea comenzii `apt - key(8)` pentru gestionarea cheilor APT este depreciată.

De asemenea, puteți verifica manual integritatea fișierului „Re lease” cu fișierul „Re lease . gpg” și cheia publică a arhivei Debian publicată pe ftp-master.debian.org folosind `gpg`.

2.5.3 Fișiere „Release” la nivel de arhivă

Indicație

Fișierele „Re lease” la nivel de arhivă sunt utilizate pentru regula `apt _ preferences(5)`.

Există fișiere „Release” la nivel de arhivă pentru toate locațiile de arhivă specificate de **lista sursă**, cum ar fi „<http://deb.debian.org/debian/dists/unstable/main/binary-amd64/Release>” sau „<http://deb.debian.org/debian/dists/testing/main/binary-amd64/Release>” după cum urmează.

```
Archive: unstable
Origin: Debian
Label: Debian
Component: main
Architecture: amd64
```

**Atenție**

Pentru strofa „Archive:”, numele suitei („stable”, „testing”, „unstable”, ...) sunt utilizate în [arhiva Debian](#), în timp ce numele în cod („trusty”, „xenial”, „artful”, ...) sunt utilizate în [arhiva Ubuntu](#).

Pentru unele arhive, cum ar fi `experimental` și `trixie-backports`, care conțin pachete care nu trebuie instalate automat, există o linie suplimentară, de exemplu, „`http://deb.debian.org/debian/dists/experimental/main/b`” după cum urmează.

```
Archive: experimental
Origin: Debian
Label: Debian
NotAutomatic: yes
Component: main
Architecture: amd64
```

Vă rugăm să rețineți că pentru arhivele normale fără „`NotAutomatic: yes`”, valoarea implicită a `Pin-Priority` este 500, în timp ce pentru arhivele speciale cu „`NotAutomatic: yes`”, valoarea implicită a `Pin-Priority` este 1 (consultați `apt_preferences(5)` și Secțiune 2.7.7).

2.5.4 Preluarea metadatelor pentru pachet

Când se utilizează instrumentele APT, cum ar fi `aptitude`, `apt-get`, `synaptic`, `apt-file`, `auto-apt`, ..., trebuie să actualizăm copiile locale ale metadatelor care conțin informațiile din arhiva Debian. Aceste copii locale au următoarele nume de fișiere corespunzătoare numelor specificate distribuție, secțiune și arhitectură din **lista surselor** (a se vedea Secțiune 2.1.5).

- „`/var/lib/apt/lists/deb.debian.org_debian_dists_distribuția_Release`”
- „`/var/lib/apt/lists/deb.debian.org_debian_dists_distribuția_Release.gpg`”
- „`/var/lib/apt/lists/deb.debian.org_debian_dists_distribuția_secțiunea_binary-arhitectura`”
- „`/var/lib/apt/lists/deb.debian.org_debian_dists_distribuția_secțiunea_source_Sources`”
- „`/var/cache/apt/apt-file/deb.debian.org_debian_dists_distribuția_Contents-arhitectura.gz`” (pentru `apt-file`)

Primele 4 tipuri de fișiere sunt comune tuturor comenzilor APT relevante și sunt actualizate din linia de comandă prin „`apt-get update`” sau „`aptitude update`”. Metadatele „`Packages`” sunt actualizate dacă „`deb`” este specificat în **lista de surse**. Metadatele „`Sources`” sunt actualizate dacă „`deb-src`” este specificat în **lista de surse**.

Metadatele „`Packages`” și „`Sources`” conțin „`Filename:`” care indică locația fișierului pachetelor binare și sursă. În prezent, aceste pachete se află în arborele de directoare „`pool/`” pentru o tranziție îmbunătățită între versiuni.

Copiile locale ale metadatelor „`Packages`” pot fi căutate interactiv cu ajutorul `aptitude`. Comanda de căutare specializată `grep-dctrl(1)` poate căuta copii locale ale metadatelor „`Packages`” și „`Sources`”.

Copia locală a metadatelor „`Contents-architecture`” poate fi actualizată cu „`apt-file update`”, iar locația acesteia este diferită de celelalte 4. Consultați `apt-file(1)`. (`auto-apt` utilizează o locație diferită pentru copia locală a „`Contents-architecture.gz`” ca locație implicită.)

2.5.5 Starea pachetului pentru APT

În plus față de metadatele preluate de la distanță, instrumentul APT după `Lenny` stochează informațiile despre starea instalării generate local în „`/var/lib/apt/extended_states`”, care este utilizat de toate instrumentele APT pentru a urmări toate pachetele instalate automat.

2.5.6 Starea pachetului pentru aptitude

În plus față de metadatele preluate de la distanță, comanda `aptitude` stochează informațiile despre starea instalării generate local în „`/var/lib/aptitude/pkgstates`”, care este utilizat numai de aceasta.

2.5.7 Copiile locale ale pachetelor descărcate

Toate pachetele descărcate de la distanță prin mecanismul APT sunt stocate în „/var/cache/apt/archives” până când sunt șterse.

Această politică de curățare a fișierelor cache pentru aptitude poate fi configurată în „Opțiuni” → „Preferințe” și poate fi forțată prin meniul „Curăță cache-ul pachetelor” sau „Curăță fișierele învechite” din „Acțiuni”.

2.5.8 Numele fișierelor pachetelor Debian

Fișierele pachetelor Debian au structuri de nume specifice.

tipul pachetului	structura numelui
Pachetul binar (cunoscut și sub numele de deb)	<i>nume-pachet_versiunea-upstream-revizia-debian_arhitectura</i>
Pachetul binar pentru debian-installer (cunoscut și sub numele de udeb)	<i>nume-pachet_versiunea-upstream-revizia-debian_arhitectura</i>
Pachetul sursă (sursă upstream)	<i>nume-pachet_versiunea-upstream-revizia-debian.orig.tar.gz</i>
Pachetul sursă 1.0 (modificări Debian)	<i>nume-pachet_versiunea-upstream-revizia-debian.diff.gz</i>
Pachetul sursă 3.0 (quilt) (modificări Debian)	<i>nume-pachet_versiunea-upstream-revizia-debian.debian.tar.gz</i>
Pachetul sursă (descrierea)	<i>nume-pachet_versiunea-upstream-revizia-debian.dsc</i>

Tabela 2.15: Structura numelor pachetelor Debian

Indicație

Aici sunt descrise doar formatele de bază ale pachetelor sursă. Pentru mai multe informații, consultați `dpkg-source(1)`.

numele componentei	caractere utilizabile (expresie regulată ERE)	existență
<i>nume-pachet</i>	<code>[a-z0-9] [-a-z0-9.+] +</code>	cerut
<i>epoca:</i>	<code>[0-9] +:</code>	opțional
<i>versiunea-upstream</i>	<code>[-a-zA-Z0-9.+:] +</code>	cerut
<i>revizia-debian</i>	<code>[a-zA-Z0-9.+~] +</code>	opțional

Tabela 2.16: Caracterele utilizabile pentru fiecare componentă din numele pachetelor Debian

Notă

Puteți verifica ordinea versiunilor pachetelor cu ajutorul comenzii `dpkg(1)`, de exemplu, «`dpkg --compare-versions 7.0 gt 7.~pre1 ; echo $?`».

Notă

[Programul de instalare Debian «debian-installer» \(d-i\)](#) utilizează extensia de fișier udeb pentru pachetul său binar, în loc de extensia normală deb. Un pachet udeb este un pachet deb simplificat, din care sunt eliminate câteva conținuturi neesențiale, cum ar fi documentația, pentru a economisi spațiu și a relaxa cerințele politicii privind pachetele. Atât pachetele deb, cât și cele udeb au aceeași structură. Litera „u” înseamnă micro.

2.5.9 Comanda «dpkg»

dpkg(1) este instrumentul de nivel inferior pentru gestionarea pachetelor Debian. Acesta este foarte puternic și trebuie utilizat cu precauție.

În timpul instalării pachetului numit „*nume-pachet*”, dpkg îl procesează în următoarea ordine.

1. Despachetează fișierul deb (echivalentul „*ar -x*”)
2. Execută „*nume-pachet.preinst*” folosind debconf(1)
3. Instalează conținutul pachetului în sistem (echivalentul „*tar -x*”)
4. Execută „*nume-pachet.postinst*” folosind debconf(1)

Sistemul debconf oferă interacțiune standardizată cu utilizatorul, cu suport pentru I18N și L10N (Cap. 8).

fișier	descrierea conținutului
/var/lib/dpkg/info/ <i>nume-pachet</i> .conf	fișierul de configurare. (modificabile de utilizator)
/var/lib/dpkg/info/ <i>nume-pachet</i> .list	lista fișierelor și directoarelor instalate de pachet
/var/lib/dpkg/info/ <i>nume-pachet</i> .md5sums	lista valorilor sumelor de control MD5 pentru fișierele instalate de pachet
/var/lib/dpkg/info/ <i>nume-pachet.preinst</i>	scriptul pachetului care trebuie rulat înainte de instalarea pachetului
/var/lib/dpkg/info/ <i>nume-pachet.postinst</i>	scriptul pachetului care trebuie rulat după instalarea pachetului
/var/lib/dpkg/info/ <i>nume-pachet.prerm</i>	scriptul pachetului care trebuie rulat înainte de eliminarea pachetului
/var/lib/dpkg/info/ <i>nume-pachet.postrm</i>	scriptul pachetului care trebuie rulat după eliminarea pachetului
/var/lib/dpkg/info/ <i>nume-pachet.conff</i>	scriptul pachetului pentru sistemul debconf
/var/lib/dpkg/alternatives/ <i>nume-pachet</i>	informațiile alternative utilizate de comanda <i>update-alternatives</i>
/var/lib/dpkg/available	informațiile privind disponibilitatea tuturor pachetelor
/var/lib/dpkg/diversions	informațiile de redirectionare utilizate de dpkg(1) și stabilite de dpkg-divert(8)
/var/lib/dpkg/statoverride	informațiile de suprascriere a stării utilizate de dpkg(1) și stabilite de dpkg-statoverride(8)
/var/lib/dpkg/status	informațiile privind starea tuturor pachetelor
/var/lib/dpkg/status-old	copia de rezervă de prima generație a fișierului „var/lib/dpkg/status”
/var/backups/dpkg.status*	copia de rezervă de a doua generație și cele mai vechi ale fișierului „var/lib/dpkg/status”

Tabela 2.17: Fișierele importante create de dpkg

Fișierul „status” este utilizat și de instrumente precum dpkg(1), „dselect update” și „apt-get -u dselect-upgrade”.

Comanda de căutare specializată *grep -dctrl(1)* poate căuta în copiile locale ale metadatelor „status” și „available”.

Indicație

În mediul [debian-installer](#), comanda *udpkg* este utilizată pentru a deschide pachetele udeb. Comanda *udpkg* este o versiune simplificată a comenzii *dpkg*.

2.5.10 Comanda «update-alternatives»

Sistemul Debian dispune de un mecanism care permite instalarea fără probleme a programelor care se suprapun într-o oarecare măsură, folosind *update-alternatives(1)*. De exemplu, puteți face ca comanda *vi* să selecteze rularea *vim* în timp ce instalați atât pachetul *vim*, cât și pachetul *nvi*.

```
$ ls -l $(type -p vi)
lrwxrwxrwx 1 root root 20 2007-03-24 19:05 /usr/bin/vi -> /etc/alternatives/vi
$ sudo update-alternatives --display vi
...
$ sudo update-alternatives --config vi
  Selection    Command
-----
          1    /usr/bin/vim
*+        2    /usr/bin/nvi

Enter to keep the default[*], or type selection number: 1
```

Sistemul Debian alternatives păstrează selecția sa sub formă de legături simbolice în „/etc/alternatives/”. Procesul de selecție utilizează fișierul corespunzător din „/var/lib/dpkg/alternatives/”.

2.5.11 Comanda «dpkg-statoverride»

Suprascriderile de stare furnizate de comanda `dpkg-statoverride(8)` sunt o modalitate de a indica `dpkg(1)` să utilizeze un proprietar sau un mod diferit pentru un **fișier** atunci când un pachet este instalat. Dacă se specifică „- -update” și fișierul există, acesta este configurat imediat cu noul proprietar și mod.



Atenție

Modificarea directă a proprietarului sau a modului pentru un **fișier** deținut de pachet folosind comenzile `chmod` sau `chown` de către administratorul de sistem este reinițializată la următoarea actualizare a pachetului.

Notă

Folosesc aici cuvântul **fișier**, dar în realitate acesta poate fi orice obiect al sistemului de fișiere pe care `dpkg` îl gestionează, inclusiv directoare, dispozitive etc.

2.5.12 Comanda «dpkg-divert»

Redirecționările de fișiere furnizate de comanda `dpkg-divert(8)` sunt o modalitate de a forța `dpkg(1)` să nu instaleze un fișier în locația sa implicită, ci într-o locație **redirecționată**. Utilizarea `dpkg-divert` este destinată scripturilor de întreținere a pachetelor. Utilizarea sa ocazională de către administratorul de sistem este depreciață.

2.6 Recuperarea dintr-un sistem defect

Când rulează un sistem de testare sau instabil, administratorul trebuie să remedieze situația de gestionare defectuoasă a pachetelor.



Atenție

Unele metode descrise aici sunt acțiuni cu risc ridicat. Ați fost avertizați!

2.6.1 Instalare eșuată din cauza dependențelor lipsă

Dacă forțați instalarea unui pachet prin „`sudo dpkg -i ...`” într-un sistem în care nu sunt instalate toate pachetele dependente, instalarea pachetului va eșua, fiind instalat parțial.

Trebuie să instalați toate pachetele dependente folosind sistemul APT sau „`sudo dpkg -i ...`”.

Apoi, configurați toate pachetele instalate parțial cu următoarea comandă.

```
# dpkg --configure -a
```

2.6.2 Erori de stocare în cache ale datelor pachetului

Erorile de stocare în cache a datelor pachetului provoacă erori intrigante, cum ar fi „[GPG error: ... invalid: BADSIG ..](#)” cu APT.

Trebuie să ștergeți toate datele din cache cu comanda „`sudo rm -rf /var/lib/apt/*`” și să încercați din nou. (Dacă se utilizează `apt-cacher-ng`, trebuie să executați și comanda „`sudo rm -rf /var/cache/apt-cacher-ng/*`”.)

2.6.3 Incompatibilitate cu configurația veche a utilizatorului

Dacă un program cu interfață grafică pentru mediul grafic de birou a prezentat instabilitate după o actualizare semnificativă a versiunii din amonte, ar trebui să suspectați o interferență cu fișierele de configurare locale vechi create de acesta. Dacă este stabil sub un cont de utilizator nou creat, această ipoteză este confirmată; (aceasta este o eroare de împachetare și, de obicei, este evitată de către responsabilul Debian al pachetului).

Pentru a restabili stabilitatea, trebuie să mutați fișierele de configurare locale corespunzătoare și să reporniți programul cu interfață grafică. Este posibil să fie necesar să citiți conținutul fișierelor de configurare vechi pentru a recupera informațiile de configurare ulterior. (Nu le ștergeți prea repede.)

2.6.4 Pachete diferite cu fișiere suprapuse

Sistemele de gestionare a pachetelor la nivel de arhivă, precum `aptitude(8)` sau `apt-get(1)`, nici măcar nu încercă să instaleze pachete cu fișiere suprapuse utilizând dependențele pachetelor (a se vedea Secțiune 2.1.7).

Erorile comise de responsabilul cu întreținerea pachetului sau implementarea unor surse de arhive mixte inconsistente (a se vedea Secțiune 2.7.6) de către administratorul de sistem pot crea o situație în care dependențele pachetului sunt definite incorect. Când instalați un pachet cu fișiere suprapuse folosind `aptitude(8)` sau `apt-get(1)` într-o astfel de situație, `dpkg(1)`, care despachetează pachetul, se asigură că returnează o eroare programului care l-a apelat, fără a suprascrie fișierele existente.



Atenție

Utilizarea pachetelor terțe introduce riscuri semnificative pentru sistem prin intermediul scripturilor responsabilului cu întreținerea pachetului care sunt rulate cu privilegii de root și pot face orice în sistemul dvs. Comanda `dpkg(1)` protejează doar împotriva suprascrierii prin despachetare.

Puteți remedia o astfel de instalare defectuoasă eliminând mai întâi pachetul vechi care cauzează problema, *pachetul-vechi*:

```
$ sudo dpkg -P old-package
```

2.6.5 Remedierea scriptului pachetului defect

Când o comandă din scriptul pachetului returnează o eroare din anumite motive și scriptul se încheie cu o eroare, sistemul de gestionare a pachetelor întrerupe acțiunea și pachetele rămân parțial instalate. Când un pachet conține erori în scripturile sale de eliminare, pachetul poate deveni imposibil de eliminat și destul de neplăcut.

Pentru problema scriptului pachetului „*nume-pachet*”, ar trebui să consultați următoarele scripturi ale pachetului.

- „`/var/lib/dpkg/info/nume-pachet.preinst`”
- „`/var/lib/dpkg/info/nume-pachet.postinst`”
- „`/var/lib/dpkg/info/nume-pachet.prerm`”
- „`/var/lib/dpkg/info/nume-pachet.postrm`”

Editați scriptul pachetului problematic din directorul rădăcină folosind următoarele tehnici:

- dezactivați linia problematică precedând-o cu „`#`”
- forțați returnarea succesului prin adăugarea la linia problematică a „`|| true`”

Apoi, urmați instrucțiunile din Secțiune 2.6.

2.6.6 Recuperarea cu comanda dpkg

Deoarece dpkg este un instrument de pachete de nivel foarte scăzut, acesta poate funcționa în situații foarte dificile, cum ar fi un sistem care nu poate fi pornit și fără conexiune la rețea. Să presupunem că pachetul *foo* s-a defectat și trebuie înlocuit.

Este posibil să găsiți încă copii cache ale versiunii mai vechi, fără erori, a pachetului *foo* în directorul cache al pachetului: „`/var/cache/apt/archives/`”. (Dacă nu, îl puteți descărca din arhiva <https://snapshot.debian.org/> sau îl puteți copia din cache-ul pachetelor unei mașini funcționale.)

Dacă puteți porni sistemul, îl puteți instala folosind următoarea comandă:

```
# dpkg -i /path/to/foo_old_version_arch.deb
```

Indicație

Dacă defecțiunea sistemului este minoră, puteți alternativ să faceți retrogradarea întregului sistem ca în Secțiune 2.7.11 folosind sistemul APT de nivel superior.

Dacă sistemul dvs. nu poate fi pornit de pe discul dur, trebuie să căutați alte modalități de pornire.

1. Porniți sistemul utilizând CD-ul *debian-installer* în modul de recuperare.
2. Montați sistemul care nu poate fi pornit de pe discul dur în „`/target`”.
3. Instalați versiunea mai veche a pachetului *foo* urmând pașii de mai jos.

```
# dpkg --root /target -i /path/to/foo_old_version_arch.deb
```

Acest exemplu funcționează chiar dacă comanda dpkg de pe discul dur este defectă.

Indicație

Orice sistem GNU/Linux pornit de un alt sistem de pe un disc dur, un CD GNU/Linux live, o unitate USB aptă de-a porni sau prin rețea „netboot” poate fi utilizat în mod similar pentru a salva un sistem defect.

Dacă încercarea de a instala un pachet în acest mod eșuează din cauza unor încălcări ale dependențelor și trebuie neapărat să faceți acest lucru ca ultimă soluție, puteți ignora dependențele folosind opțiunile „- - ignore-depends”, „- - force-depends” și alte opțiuni ale dpkg. Dacă faceți acest lucru, trebuie să depuneți eforturi serioase pentru a restabili dependența corespunzătoare ulterior. Consultați dpkg(8) pentru detalii.

Notă

Dacă sistemul dvs. este grav defect, ar trebui să faceți o copie de rezervă completă a sistemului într-un loc sigur (consultați Secțiune 10.2) și să efectuați o instalare curată. Aceasta consumă mai puțin timp și produce rezultate mai bune în final.

2.6.7 Recuperarea datelor privind selecția pachetelor

Dacă „/var/lib/dpkg/status” se corupe din orice motiv, sistemul Debian pierde datele de selecție a pachetelor și suferă grav. Căutați vechiul fișier „/var/lib/dpkg/status” în „/var/lib/dpkg/status-old” sau „/var/backups/dpkg.status.*”.

Păstrarea „/var/backups/” într-o partiție separată poate fi o idee bună, deoarece acest director conține multe date importante ale sistemului.

În cazul unei defecțiuni grave, recomand să reinstalați sistemul după ce ați făcut o copie de rezervă a acestuia. Chiar dacă totul din „/var/” a dispărut, puteți recupera unele informații din directoarele din „/usr/share/doc/” pentru a vă ghida în noua instalare.

Reinstalați sistemul minimal (mediul de birou).

```
# mkdir -p /path/to/old/system
```

Montați sistemul vechi la „/path/to/old/system/ - (/ruta/către/sistemul/vechi/)”.

```
# cd /path/to/old/system/usr/share/doc
# ls -1 >~/ls1.txt
# cd /usr/share/doc
# ls -1 >>~/ls1.txt
# cd
# sort ls1.txt | uniq | less
```

Apoi vi se prezintă numele pachetelor care trebuie instalate. (Pot exista și unele nume care nu sunt ale pachetelor, cum ar fi „texmf”.)

2.7 Sfaturi pentru gestionarea pachetelor

Pentru simplitate, exemplele **listei surselor** din această secțiune sunt prezentate ca „/etc/apt/sources.list” într-un stil pe o singură linie după versiunea trixie.

2.7.1 Cine a încărcat pachetul?

Deși numele responsabilului cu întreținerea pachetului listat în „/var/lib/dpkg/available” și „/usr/share/doc/pack” oferă unele informații despre „cine se află în spatele activității de împachetare”, persoana care a încărcat efectiv pachetul este oarecum necunoscută. `who-uploads(1)` din pachetul `devscripts` identifică persoana care a încărcat efectiv pachetele sursă Debian.

2.7.2 Limitarea lății de bandă pentru descărcare pentru APT

Dacă doriți să limitați lățimea de bandă de descărcare pentru APT la, de exemplu, 800 Kio/sec (=100 kio/sec), trebuie să configurați APT cu parametrul său de configurare după cum urmează.

```
APT::Acquire::http::DL-Limit "800";
```

2.7.3 Descărcarea și actualizarea automată a pachetelor

Pachetul apt vine cu propriul script cron „/etc/cron.daily/apt” pentru a sprijini descărcarea automată a pachetelor. Acest script poate fi îmbunătățit pentru a efectua actualizarea automată a pachetelor prin instalarea pachetului `unattended-upgrades`. Acestea pot fi personalizate prin parametrii din „/etc/apt/apt.conf.d/02backup” și „/etc/apt/apt.conf.d/50unattended-upgrades”, așa cum este descris în „/usr/share/doc/unattended-upgrades/”.

Pachetul `unattended-upgrades` este destinat în principal actualizării de securitate pentru sistemul stabil. Dacă riscul de a afecta un sistem stabil existent prin actualizarea automată este mai mic decât riscul ca sistemul să fie afectat de un intrus care utilizează o breșă de securitate care a fost remediată prin actualizarea de securitate, ar trebui să luați în considerare utilizarea acestei actualizări automate cu parametrii de configurare următori.

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::Unattended-Upgrade "1";
```

Dacă utilizați un sistem `testing` sau `unstable`, nu este recomandat să utilizați actualizarea automată, deoarece aceasta va duce cu siguranță la defectarea sistemului într-o zi. Chiar și în cazul unui astfel de sistem `testing` sau `unstable`, este posibil să doriți să descărcați pachetele în avans pentru a economisi timp pentru actualizarea interactivă cu parametrii de configurare, după cum urmează.

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::Unattended-Upgrade "0";
```

2.7.4 Actualizări și retro-adaptări (versiuni de software migrate din ramura principală de dezvoltare și adaptate pentru a funcționa cu această versiune)

Există [actualizări stabile](#) („trixie-updates” în timpul ciclului de lansare `trixie-ca-stable`) și arhive backports.debian.org care oferă pachete de actualizare pentru `stable`.

Pentru a utiliza aceste arhive, listați toate arhivele necesare în fișierul „/etc/apt/sources.list” după cum urmează.

```
deb http://deb.debian.org/debian/ trixie main non-free-firmware contrib non-free
deb http://security.debian.org/debian-security trixie-security main non-free-firmware ↵
    contrib non-free
deb http://deb.debian.org/debian/ trixie-updates main non-free-firmware contrib non-free
deb http://deb.debian.org/debian/ trixie-backports main non-free-firmware contrib non-free
```

Nu este necesar să stabiliți explicit valoarea `Pin-Priority` în fișierul „/etc/apt/preferences”. Când apar pachete mai noi, configurația implicită oferă cele mai rezonabile actualizări (consultați Secțiune [2.5.3](#)).

- Toate pachetele vechi instalate sunt actualizate la versiuni mai noi din `trixie-updates`.
- Numai pachetele mai vechi instalate manual din `trixie-backports` sunt actualizate la versiuni mai noi din `trixie-backports`.

Ori de câte ori doriți să instalați manual un pachet numit „*nume-pachet*” cu dependența sa din arhiva `trixie-backports`, utilizați următoarea comandă în timp ce comutați versiunea țintă cu opțiunea „-t”.

```
$ sudo apt-get install -t trixie-backports package-name
```

**Avertisment**

Nu instalați prea multe pachete din arhivele backports.debian.org. Acest lucru poate cauza complicații legate de dependențele pachetelor. Consultați Secțiune [2.1.11](#) pentru soluții alternative.

2.7.5 Arhive de pachete externe

**Avertisment**

Trebuie să știți că pachetul extern obține privilegii de root asupra sistemului dvs. Trebuie să utilizați numai arhiva de pachete externe de încredere. Consultați Secțiune [2.1.11](#) pentru soluții alternative.

Puteți utiliza APT securizat cu arhiva de pachete externe compatibilă cu Debian adăugând-o la **lista de surse** și fișierul cheie al arhivei în directorul „/etc/apt/trusted.gpg.d/”. Consultați `sources.list(5)`, `apt-secure(8)` și `apt-key(8)`.

2.7.6 Pachete din surse mixte de arhive fără apt-pinning

**Atenție**

Instalarea pachetelor din surse mixte de arhive nu este acceptată de distribuția oficială Debian, cu excepția combinațiilor de arhive acceptate oficial, cum ar fi `stable` cu [actualizări de securitate](#) și [stable-updates](#).

Iată un exemplu de operații pentru a include pachete specifice din versiuni upstream mai noi găsite în `unstable` în timp ce se urmărește `testing` pentru o singură ocazie.

1. Modificați temporar fișierul „/etc/apt/sources.list” pentru a include o singură intrare „unstable”.
2. Rulați „`aptitude update`”.
3. Rulați „`aptitude install nume-pachet`”.
4. Recuperați fișierul original „/etc/apt/sources.list” pentru `testing`.
5. Rulați „`aptitude update`”.

Nu creați fișierul „/etc/apt/preferences” și nici nu trebuie să vă faceți griji cu privire la **apt-pinning** cu această abordare manuală. Dar acest lucru este foarte greoi.

**Atenție**

Când utilizați surse mixte de arhive, trebuie să vă asigurați singuri de compatibilitatea pachetelor, deoarece Debian nu o garantează. Dacă există incompatibilități între pachete, puteți deteriora sistemul. Trebuie să fiți capabili să evaluați aceste cerințe tehnice. Utilizarea surselor mixte de arhive aleatorii este o operație complet opțională și nu vă încurajez să o utilizați.

Regulile generale pentru instalarea pachetelor din diferite arhive sunt următoarele.

- Pachetele non-binare („Architecture: all”) sunt **mai sigure** de instalat.
 - pachete de documentație: fără cerințe speciale
 - pachete de programe interpret: trebuie să fie disponibil un interpret compatibil
- Pachetele binare (care nu sunt „Architecture: all”) se confruntă de obicei cu multe obstacole și sunt **nesigure** de instalat.
 - compatibilitatea versiunii bibliotecii (inclusiv „libc”)
 - compatibilitatea versiunii programului utilitar asociat
 - compatibilitatea cu [ABI](#) a nucleului
 - compatibilitatea cu [ABI](#) C++
 - ...

Notă

Pentru a face un pachet mai **sigur** de instalat, unele pachete comerciale de programe binare ne-libere pot fi furnizate cu biblioteci complet legate static. Ar trebui totuși să verificați problemele de compatibilitate [ABI](#) etc. pentru acestea.

Notă

Cu excepția cazului în care doriți să evitați deteriorarea pachetului pe termen scurt, instalarea pachetelor binare din arhive non-Debian este, în general, o idee proastă. Ar trebui să căutați toate soluțiile tehnice alternative mai sigure disponibile, care sunt compatibile cu sistemul Debian actual (consultați Secțiune [2.1.11](#)).

2.7.7 Ajustarea versiunii candidate cu apt-pinning

**Avertisment**

Utilizarea tehnicii **apt-pinning** de către un utilizator începător poate cauza probleme majore. Trebuie să evitați utilizarea acestei tehnici, cu excepția cazurilor în care este absolut necesar.

Fără fișierul „/etc/apt/preferences”, sistemul APT alege cea mai recentă versiune disponibilă ca **versiune candidată** utilizând șirul de versiune. Aceasta este starea normală și utilizarea cea mai recomandată a sistemului APT. Toate combinațiile de arhive acceptate oficial nu necesită fișierul „/etc/apt/preferences”, deoarece unele arhive care nu ar trebui utilizate ca sursă automată de actualizări sunt marcate ca **NotAutomatic** și tratate corespunzător.

Indicație

Regula de comparare a șirurilor de versiuni poate fi verificată, de exemplu, cu „dpkg --compare-versions ver1.1 gt ver1.1~1; echo \$?” (a se vedea dpkg(1)).

Când instalați regulat pachete din surse mixte de arhive (a se vedea Secțiune [2.7.6](#)), puteți automatiza aceste operații complicate creând fișierul „/etc/apt/preferences” cu intrările corespunzătoare și modificând regula de selecție a pachetelor pentru **versiunea candidată**, așa cum este descris în apt_preferences(5). Aceasta se numește **apt-pinning**.

Când utilizați **apt-pinning**, trebuie să vă asigurați singuri de compatibilitatea pachetelor, deoarece Debian nu o garantează. **apt-pinning** este o operație complet opțională și nu vă încurajează să o utilizați.

Fișierele Release la nivel de arhivă (vedeți Secțiune [2.5.3](#)) sunt utilizate pentru regula apt_preferences(5). Astfel, **apt-pinning** funcționează numai cu numele „suitei” pentru [arhivele Debian normale](#) și [arhivele Debian de securitate](#).

(Acest lucru este diferit de arhivele [Ubuntu](#).) De exemplu, puteți face „Pin: release a=unstable”, dar nu puteți face „Pin: release a=sid” în fișierul „/etc/apt/preferences”.

Când utilizați arhive non-Debian ca parte a **apt-pinning**, trebuie să verificați pentru ce sunt destinate și să verificați credibilitatea acestora. De exemplu, Ubuntu și Debian nu sunt destinate a fi combinate.

Notă

Chiar dacă nu creați fișierul „/etc/apt/preferences” , puteți efectua operații de sistem destul de complexe (consultați Secțiune [2.6.6](#) și Secțiune [2.7.6](#)) fără **apt-pinning**.

Iată o explicație simplificată a tehnicii **apt-pinning**.

Sistemul APT alege pachetul cu cea mai mare prioritate Pin **upgrading** din sursele de pachete disponibile definite în fișierul „/etc/apt/sources.list” ca pachet **candidat**. Dacă prioritatea Pin a pachetului este mai mare de 1000, această restricție de versiune pentru **actualizare** este eliminată pentru a permite retrogradarea (vedeți Secțiune [2.7.11](#)).

Valoarea „Pin-Priority” a fiecărui pachet este definită de intrările „Pin-Priority” din fișierul „/etc/apt/preferences” sau se utilizează valoarea implicită a acestuia.

Pin-Priority	efectele apt-pinning asupra pachetului
1001	install the package even if this constitutes a downgrade of the package
990	utilizată ca valoare implicită pentru arhiva versiunii țintă
500	utilizată ca valoare implicită pentru arhiva normală
100	utilizată ca valoare implicită pentru arhiva NotAutomatic și ButAutomaticUpgrades
100	utilizată pentru pachetul instalat
1	utilizată ca valoare implicită pentru arhiva NotAutomatic
-1	nu instalează niciodată pachetul, chiar dacă este recomandat

Tabela 2.18: Lista valorilor notabile (celor mai importante) ale priorității Pin pentru tehnica **apt-pinning**.

Arhiva **versiune țintă** poate fi definită prin opțiunea liniei de comandă, de exemplu, „apt-get install -t testing anumit-pachet”

Arhiva **NotAutomatic** și **ButAutomaticUpgrades** este definită de serverul arhivei care are fișierul Release al nivelului arhivei (a se vedea Secțiune [2.5.3](#)) conținând atât „NotAutomatic: yes” și „ButAutomaticUpgrades: yes”. Arhiva **NotAutomatic** este definită de serverul de arhivă care are fișierul Release la nivel de arhivă conținând doar „NotAutomatic: yes”.

Situația **apt-pinning** a *pachetului* din mai multe surse de arhivă este afișată prin „apt-cache policy *pachet*”.

- O linie care începe cu „Package pin:” listează versiunea pachetului **pin** dacă este definită doar asocierea cu un *pachet*, de exemplu „Package pin: 0.190”.
- Nu există nicio linie cu „Package pin:” dacă nu este definită nicio asociere doar cu un *pachet*.
- Valoarea Pin-Priority asociată doar cu *pachetul* este listată în partea dreaptă a tuturor șirurilor de versiune, de exemplu, „0.181 700”.
- „0” este afișat în partea dreaptă a tuturor șirurilor de versiune dacă nu este definită nicio asociere doar cu *pachetul*, de exemplu, „0.181 0”.
- Valorile Pin-Priority ale arhivelor (definite ca „Package: *” în fișierul „/etc/apt/preferences”) sunt listate în partea stângă a tuturor rutelor de arhivă, de exemplu, „100 http://deb.debian.org/debian/ trixie-backports Packages”.

2.7.8 Blocarea pachetelor instalate de „Recommends”



Avertisment

Utilizarea tehnicii **apt-pinning** de către un utilizator începător poate cauza probleme majore. Trebuie să evitați utilizarea acestei tehnici, cu excepția cazurilor în care este absolut necesar.

Dacă nu doriți să descărcați automat anumite pachete recomandate, trebuie să creați fișierul „/etc/apt/preferences” și să listați în mod explicit toate aceste pachete în partea de sus a acestuia, după cum urmează.

```
Package: package-1
Pin: version *
Pin-Priority: -1
```

```
Package: package-2
Pin: version *
Pin-Priority: -1
```

2.7.9 Urmărirea suitei testing cu unele pachete din unstable



Avertisment

Utilizarea tehnicii **apt-pinning** de către un utilizator începător poate cauza probleme majore. Trebuie să evitați utilizarea acestei tehnici, cu excepția cazurilor în care este absolut necesar.

Iată un exemplu de tehnică **apt-pinning** pentru a include pachete specifice mai noi din versiunea upstream găsite în unstable actualizate regulat în timp ce se urmărește testing. Enumerați toate arhivele necesare în fișierul „/etc/apt/sources.list” după cum urmează.

```
deb http://deb.debian.org/debian/ testing main contrib non-free
deb http://deb.debian.org/debian/ unstable main contrib non-free
deb http://security.debian.org/debian-security testing-security main contrib
```

Configurați fișierul „/etc/apt/preferences” după cum urmează.

```
Package: *
Pin: release a=unstable
Pin-Priority: 100
```

Când doriți să instalați un pachet numit „*nume-pachet*” cu dependențele sale din arhiva unstable în această configurație, executați următoarea comandă care comută versiunea țintă cu opțiunea „-t” (prioritatea Pin a unstable devine 990).

```
$ sudo apt-get install -t unstable package-name
```

Cu această configurație, executarea obișnuită a „apt-get upgrade” și „apt-get dist-upgrade” (sau „aptitude safe-upgrade” și „aptitude full-upgrade”) actualizează pachetele care au fost instalate din arhiva testing utilizând arhiva testing curentă și pachetele care au fost instalate din arhiva unstable utilizând arhiva unstable curentă.



Atenție

Aveți grijă să nu ștergeți intrarea „testing” din fișierul „/etc/apt/sources.list”. Fără intrarea „testing”, sistemul APT actualizează pachetele utilizând arhiva mai nouă unstable.

Indicație

De obicei, editez fișierul „/etc/apt/sources.list” pentru a comenta intrarea din arhivă „unstable” imediat după operația de mai sus. Astfel se evită încetinirea procesului de actualizare din cauza numărului prea mare de intrări din fișierul „/etc/apt/sources.list”, deși acest lucru împiedică actualizarea pachetelor instalate din arhiva unstable folosind arhiva unstable curentă.

Indicație

Dacă se utilizează „Pin-Priority: 1” în loc de „Pin-Priority: 100” în fișierul „/etc/apt/preferences”, pachetele deja instalate care au valoarea Pin-Priority de 100 nu sunt actualizate de arhiva unstable, chiar dacă intrarea „testing” din fișierul „/etc/apt/sources.list” este eliminată.

Dacă doriți să urmăriți automat anumite pachete din unstable fără instalarea inițială „-t unstable”, trebuie să creați fișierul „/etc/apt/preferences” și să enumerați în mod explicit toate aceste pachete în partea de sus a acestuia, după cum urmează.

```
Package: package-1
Pin: release a=unstable
Pin-Priority: 700
```

```
Package: package-2
Pin: release a=unstable
Pin-Priority: 700
```

Acestea stabilesc valoarea Pin-Priority pentru fiecare pachet specific. De exemplu, pentru a urmări cea mai recentă versiune unstable a acestei „Referințe Debian” în limba engleză, ar trebui să aveți următoarele intrări în fișierul „/etc/apt/preferences”.

```
Package: debian-reference-en
Pin: release a=unstable
Pin-Priority: 700

Package: debian-reference-common
Pin: release a=unstable
Pin-Priority: 700
```

Indicație

Această tehnică **apt-pinning** este valabilă chiar și atunci când urmăriți arhiva stable. Din experiența mea, până în prezent, pachetele de documentație au fost întotdeauna sigure de instalat din arhiva unstable.

2.7.10 Urmărirea suitei unstable cu unele pachete din experimental

**Avertisment**

Utilizarea tehnicii **apt-pinning** de către un utilizator începător poate cauza probleme majore. Trebuie să evitați utilizarea acestei tehnici, cu excepția cazurilor în care este absolut necesar.

Iată un alt exemplu de tehnică **apt-pinning** pentru a include pachete specifice mai noi din versiunea upstream găsite în experimental în timp ce se urmărește unstable. Enumerați toate arhivele necesare în fișierul „/etc/apt/sources.list” după cum urmează.

```
deb http://deb.debian.org/debian/ unstable main contrib non-free
deb http://deb.debian.org/debian/ experimental main contrib non-free
deb http://security.debian.org/ testing-security main contrib
```

Valoarea implicită a Pin-Priority pentru arhiva experimental este întotdeauna 1 ($\ll 100$), deoarece este o arhivă **NotAutomatic** (a se vedea Secțiune 2.5.3). Nu este necesar să stabiliți explicit valoarea Pin-Priority în fișierul „/etc/apt/preferences” doar pentru a utiliza arhiva experimental, cu excepția cazului în care doriți să urmăriți automat anumite pachete din aceasta pentru următoarea actualizare.

2.7.11 Retrogradarea de urgență



Avertisment

Utilizarea tehnicii **apt-pinning** de către un utilizator începător poate cauza probleme majore. Trebuie să evitați utilizarea acestei tehnici, cu excepția cazurilor în care este absolut necesar.



Atenție

Retrogradarea nu este acceptată oficial de Debian. Ar trebui să fie făcută doar ca parte a procesului de recuperare de urgență. În ciuda acestei situații, se știe că funcționează bine în multe cazuri. Pentru sistemele critice, ar trebui să faceți o copie de rezervă a tuturor datelor importante din sistem după operația de recuperare și să reinstalați sistemul nou de la zero.

Puteți avea norocul să faceți retrogradarea de la arhiva mai nouă la arhiva mai veche pentru a recupera din actualizarea defectuoasă a sistemului prin manipularea **versiunii candidate** (vezi Secțiune 2.7.7). Aceasta este o alternativă mai ușoară la acțiunile obositoare ale multor comenzi „`dpkg -i pachet-defect_versiune-veche.deb`” (a se vedea Secțiune 2.6.6).

Căutați în fișierul „/etc/apt/sources.list” liniile care conțin `unstable`, după cum urmează.

```
deb http://deb.debian.org/debian/ sid main contrib non-free
```

Înlocuiți-le cu următorul text pentru a urmări suita `testing` (de testare).

```
deb http://deb.debian.org/debian/ forkyn main contrib non-free
```

Configurați fișierul „/etc/apt/preferences” după cum urmează.

```
Package: *
Pin: release a=testing
Pin-Priority: 1010
```

Rulați „`apt-get update; apt-get dist-upgrade`” pentru a forța retrogradarea pachetelor din întregul sistem.

Eliminați acest fișier special „/etc/apt/preferences” după această retrogradare de urgență.

Indicație

Este o idee bună să eliminați (nu să ștergeți!) cât mai multe pachete pentru a minimiza problemele de dependență. Este posibil să fie necesar să eliminați și să instalați manual unele pachete pentru a retrograda sistemul. Nucleul Linux, încărcătorul de pornire, udev, PAM, APT și pachetele legate de rețea, precum și fișierele lor de configurare necesită o atenție specială.

2.7.12 Pachetul equivs

Dacă doriți să compilați un program din sursă pentru a înlocui pachetul Debian, cel mai bine este să îl transformați într-un pachet „debianizat” local real (*.deb) și să utilizați arhiva privată.

Dacă ați ales să compilați un program din sursă și să îl instalați în „/usr/local”, este posibil să fie necesar să utilizați equivs ca ultimă soluție pentru a satisface dependența pachetului lipsă.

```
Package: equivs
Priority: optional
Section: admin
Description: Circumventing Debian package dependencies
 This package provides a tool to create trivial Debian packages.
 Typically these packages contain only dependency information, but they
 can also include normal installed files like other packages do.
.
 One use for this is to create a metapackage: a package whose sole
 purpose is to declare dependencies and conflicts on other packages so
 that these will be automatically installed, upgraded, or removed.
.
 Another use is to circumvent dependency checking: by letting dpkg
 think a particular package name and version is installed when it
 isn't, you can work around bugs in other packages' dependencies.
 (Please do still file such bugs, though.)
```

2.7.13 Adaptarea unui pachet la sistemul stabil



Atenție

Nu există nicio garanție că procedura descrisă aici va funcționa fără eforturi manuale suplimentare din cauza diferențelor dintre sisteme.

Pentru actualizări parțiale ale sistemului stabil, este recomandabil să se reconstruiască un pachet în mediul său utilizând pachetul sursă. Astfel se evită actualizări masive ale pachetelor din cauza dependențelor acestora.

Adăugați următoarele intrări în „/etc/apt/sources.list” al unui sistem stabil.

```
deb-src http://deb.debian.org/debian unstable main contrib non-free
```

Instalați pachetele necesare pentru compilare și descărcați pachetul sursă după cum urmează.

```
# apt-get update
# apt-get dist-upgrade
# apt-get install fakeroot devscripts build-essential
# apt-get build-dep foo
$ apt-get source foo
$ cd foo*
```

Actualizați unele pachete din lanțul de instrumente, cum ar fi dpkg și debhelper din pachetele retro-adaptate „backport”, dacă acestea sunt necesare pentru retro-adaptare „backporting”.

Executați următoarele.

```
$ dch -i
```

Incrementați versiunea pachetului, de exemplu, adăugând „+bp1” în „debian/changelog”

Construiți pachetele și instalați-le în sistem după cum urmează.

```
$ debuild
$ cd ..
# debi foo*.changes
```

2.7.14 Server proxy pentru APT

Deoarece oglindirea întregii subsecțiuni a arhivei Debian consumă spațiu pe disc și lățime de bandă de rețea, implementarea unui server proxy local pentru APT este de dorit atunci când administrați mai multe sisteme într-o rețea locală LAN. APT poate fi configurat pentru a utiliza servere proxy web (http) generice, cum ar fi squid (consultați Secțiune 6.5), așa cum este descris în `apt.conf(5)` și în „`/usr/share/doc/apt/examples/configure-index.gz`”. Variabila de mediu „`$http_proxy`” poate fi utilizată pentru a suprascrie configurația serverului proxy din fișierul „`/etc/apt/apt.conf`”.

Există instrumente proxy special pentru arhiva Debian. Ar trebui să verificați BTS înainte de a le utiliza.

pachet	popcon(popularitate)	dimensiune	descriere
approx	V:0, I:0	8308	server proxy de stocare în cache pentru fișierele arhive Debian (program compilat OCaml)
apt-cacher	V:0, I:0	267	proxy cache pentru pachetele Debian și fișierele sursă (program Perl)
apt-cacher-ng	V:4, I:4	1968	proxy cache pentru distribuirea pachetelor software (program compilat în C++)

Tabela 2.19: Lista instrumentelor proxy special pentru arhiva Debian



Atenție

Când Debian își reorganizează structura arhivei, aceste instrumente proxy specializate tind să necesite rescrierea codului de către administratorul pachetului și pot să nu funcționeze pentru o perioadă. Pe de altă parte, serverele proxy web (http) generice sunt mai robuste și mai ușor de adaptat la astfel de schimbări.

2.7.15 Mai multe informații despre gestionarea pachetelor

Puteți afla mai multe despre gestionarea pachetelor din următoarele documentații.

- Documentație primară privind gestionarea pachetelor:
 - `aptitude(8)`, `dpkg(1)`, `tasksel(8)`, `apt(8)`, `apt-get(8)`, `apt-config(8)`, `apt-secure(8)`, `sources.list(5)`, `apt.conf(5)`, și `apt_preferences(5)`;
 - „`/usr/share/doc/apt-doc/guide.html/index.html`” și „`/usr/share/doc/apt-doc/offline.html/index.html`” din pachetul `apt-doc`; și
 - „`/usr/share/doc/aptitude/html/en/index.html`” din pachetul `aptitude-doc-en`.
- Documentație oficială și detaliată despre arhiva Debian:
 - [Manualul de politici Debian, Capitolul 2 - Arhiva Debian](#),
 - [„Referința dezvoltatorului Debian, Capitolul 4 - Resurse pentru dezvoltatorii Debian 4.6 Arhiva Debian”](#), și
 - [„Întrebări frecvente despre Debian GNU/Linux, Capitolul 6 – Arhivele FTP Debian”](#).
- Tutorial pentru crearea unui pachet Debian pentru utilizatorii Debian:
 - [„Ghid pentru administratorii Debian”](#).

Capitolul 3

Inițializarea sistemului

Este recomandabil ca administratorul de sistem să cunoască în linii mari modul în care sistemul Debian este pornit și configurat. Deși detaliile exacte se găsesc în fișierele sursă ale pachetelor instalate și în documentația acestora, pentru majoritatea dintre noi acestea sunt puțin copleșitoare.

Iată o prezentare generală a punctelor cheie ale inițializării sistemului Debian. Deoarece sistemul Debian este o țintă în mișcare, ar trebui să consultați cea mai recentă documentație.

- [Debian Linux Kernel Handbook - \(Manualul nucleului Debian Linux\)](#) este sursa principală de informații despre nucleul Debian.
- `bootup(7)` descrie procesul de pornire a sistemului bazat pe `systemd`. (Debian recent)
- `boot(7)` descrie procesul de pornire a sistemului bazat pe UNIX System V Release 4. (Debian mai vechi)

3.1 O prezentare generală a procesului de inițializare

Sistemul informatic trece prin mai multe faze ale [proceselor de inițializare](#) de la momentul pornirii până când oferă utilizatorului un sistem de operare (SO) complet funcțional.

Pentru simplitate, voi limita discuția la platforma tipică de PC cu instalarea implicită.

Procesul tipic de inițializare este ca o rachetă cu patru trepte (în cazul nostru, etape). Fiecare treaptă a rachetei predă controlul sistemului treptei următoare.

- Secțiune [3.1.1](#)
- Secțiune [3.1.2](#)
- Secțiune [3.1.3](#)
- Secțiune [3.1.4](#)

Desigur, acestea pot fi configurate diferit. De exemplu, dacă ați compilat propriul nucleu, este posibil să săriți peste pasul cu sistemul mini-Debian. Așadar, vă rugăm să nu presupuneți că acesta este cazul pentru sistemul dvs. până nu verificați personal.

3.1.1 Etapa 1: UEFI

Interfața firmware extensibilă unificată (UEFI) definește un administrator de pornire ca parte a specificației UEFI. Când un calculator este pornit, administratorul de pornire este prima etapă a procesului de pornire, care verifică configurația de pornire și, pe baza parametrilor săi, execută încărcătorul de sistem de operare specificat sau nucleul sistemului de operare (de obicei încărcătorul de pornire). Configurația de pornire este definită de variabile stocate în NVRAM, inclusiv variabile care indică rutele sistemului de fișiere către încărcătoarele de sistem de operare sau nucleele sistemului de operare.

O **partiție de sistem EFI (ESP)** este o partiție a dispozitivului de stocare a datelor utilizată în calculatoarele care respectă specificațiile UEFI. Accesată de firmware-ul UEFI la pornirea calculatorului, aceasta stochează aplicațiile UEFI și fișierele necesare pentru rularea acestor aplicații, inclusiv programele de încărcare a sistemului de operare. (Pe sistemele PC vechi, se poate utiliza în schimb BIOS stocat în MBR.)

3.1.2 Etapa 2: încărcătorul de pornire

Încărcătorul de pornire este a doua etapă a procesului de pornire, care este inițiat de UEFI. Acesta încarcă imaginea nucleului sistemului și imaginea **initrd** în memorie și le transferă controlul. Această imagine **initrd** este imaginea sistemului de fișiere rădăcină, iar suportul acesteia depinde de încărcătorul de pornire utilizat.

Sistemul Debian utilizează în mod normal nucleul Linux ca nucleu implicit al sistemului. Imaginea **initrd** pentru nucleul Linux 5.x actual este, din punct de vedere tehnic, imaginea **initramfs** (sistem de fișiere RAM inițial).

Există multe opțiuni disponibile pentru încărcătoare de pornire și configurare.

pachet	popcon(popularitate)	popularitate	compatibilitate	încărcător de pornire	descriere
grub-efi-amd64	I:430	142	Compatibil	GRUB UEFI	Este suficient de inteligent pentru a înțelege partițiile de disc și sistemele de fișiere precum vfat, ext4, (UEFI)
grub-pc	V:17, I:545	479	Compatibil	GRUB 2	Este suficient de inteligent pentru a înțelege partițiile de disc și sistemele de fișiere precum vfat, ext4, (BIOS)
grub-rescue-pc	V:0, I:0	7183	Compatibil	GRUB 2	Acestea sunt imagini de recuperare care pot fi pornite GRUB 2 (CD și dischetă) (versiunea PC/BIOS)
syslinux	V:2, I:32	325	Compatibil	Isolinux	Acesta înțelege sistemul de fișiere ISO9660. Acesta este utilizat de CD-ul de pornire.
syslinux	V:2, I:32	325	Compatibil	Syslinux	Acesta înțelege sistemul de fișiere MSDOS (FAT) . Acesta este utilizat de discheta de pornire.
loadlin	V:0, I:0	87	Compatibil	Loadlin	Noul sistem este pornit din sistemul FreeDOS/MSDOS.
mbr	V:0, I:3	47	Necompatibil	MBR de Neil Turton	Acesta este un software liber care înlocuiește MSDOS MBR . Acesta recunoaște numai partițiile de disc.

Tabela 3.1: Lista încărcătorilor de pornire

**Avertisment**

Nu vă jucați cu încărcătoarele de pornire fără a avea un suport de recuperare de pornire (stick de memorie USB, CD sau dischetă) creat din imaginile din pachetul `grub-rescue-pc`. Acesta vă permite să porniți sistemul chiar și fără un încărcător de pornire funcțional pe discul dur.

Pentru sistemul UEFI, GRUB2 citește mai întâi partiția ESP și utilizează UUID-ul specificat pentru `search.fs_uuid` în `„/boot/efi/EFI/debian/grub.cfg”` pentru a determina partiția fișierului de configurare a meniului GRUB2 `„/boot/grub/grub.cfg”`.

Partea esențială a fișierului de configurare a meniului GRUB2 arată astfel:

```
menuentry 'Debian GNU/Linux' ... {
    load_video
    insmod gzio
    insmod part_gpt
    insmod ext2
    search --no-floppy --fs-uuid --set=root fe3e1db5-6454-46d6-a14c-071208ebe4b1
    echo    'Loading Linux 5.10.0-6-amd64 ...'
    linux   /boot/vmlinuz-5.10.0-6-amd64 root=UUID=fe3e1db5-6454-46d6-a14c-071208ebe4b1 ←
    ro quiet
    echo    'Loading initial ramdisk ...'
    initrd  /boot/initrd.img-5.10.0-6-amd64
}
```

Pentru această parte din `/boot/grub/grub.cfg`, această intrare din meniu înseamnă următoarele.

parametrul	valoare
module GRUB2 încărcate	<code>gzio, part_gpt, ext2</code>
partiția sistemului de fișiere rădăcină utilizată	partiția identificată prin <code>UUID=fe3e1db5-6454-46d6-a14c-071208ebe4b1</code>
ruta către imaginea nucleului în sistemul de fișiere rădăcină	<code>/boot/vmlinuz-5.10.0-6-amd64</code>
parametru de pornire al nucleului utilizat	<code>"root=UUID=fe3e1db5-6454-46d6-a14c-071208ebe4b1 ro quiet"</code>
ruta către imaginea „initrd” în sistemul de fișiere rădăcină	<code>/boot/initrd.img-5.10.0-6-amd64</code>

Tabela 3.2: Semnificația intrării din meniul din partea de sus a `/boot/grub/grub.cfg`

Indicație

Puteți activa afișarea mesajelor jurnalului de pornire a nucleului eliminând `quiet` din `„/boot/grub/grub.cfg”`. Pentru o modificare permanentă, editați linia `„GRUB_CMDLINE_LINUX_DEFAULT=„quiet”` din `„/etc/default/grub”`.

Indicație

Puteți personaliza imaginea de pornire GRUB definind variabila `GRUB_BACKGROUND` în `„/etc/default/grub”` pentru a indica fișierul imagine sau plasând fișierul imagine în `„/boot/grub/”`.

A se vedea `"info grub"` și `grub-install(8)`.

3.1.3 Etapa 3: sistemul mini-Debian

Sistemul mini-Debian este a treia etapă a procesului de pornire, care este inițiată de încărcătorul de pornire. Acesta rulează nucleul sistemului cu sistemul său de fișiere rădăcină în memorie. Aceasta este o etapă pregătitoare opțională a procesului de pornire.

Notă

Termenul „sistemul mini-Debian” este inventat de autor pentru a descrie această a treia etapă a procesului de pornire în acest document. Acest sistem este denumit în mod obișnuit [initrd](#) sau sistem `initramfs`. Un sistem similar în memorie este utilizat de [programul de instalare Debian](#).

Programul `„/init”` este executat ca primul program din acest sistem de fișiere rădăcină din memorie. Este un program care inițializează nucleul în spațiul utilizatorului și transferă controlul către etapa următoare. Acest mini-sistem Debian oferă flexibilitate procesului de pornire, cum ar fi adăugarea de module de nucleu înainte de procesul principal de pornire sau montarea sistemului de fișiere rădăcină ca unul criptat.

- Programul `„/init”` este un program script shell dacă `initramfs` a fost creat de `initramfs-tools`.
 - Puteți întrerupe această parte a procesului de pornire pentru a obține shell-ul root, furnizând `„break=init”` etc. parametrului de pornire al nucleului. Consultați scriptul `„/init”` pentru mai multe condiții de întrerupere. Acest mediu shell este suficient de sofisticat pentru a efectua o inspecție bună a hardware-ului mașinii dvs.
 - Comenzile disponibile în acest mini-sistem Debian sunt simplificate și furnizate în principal de un instrument GNU numit `busybox(1)`.
- Programul `„/init”` este un program binar `systemd` dacă `initramfs` a fost creat de `dracut`.
 - Comenzile disponibile în acest mini-sistem Debian sunt reduse la mediul `systemd(1)`.



Atenție

Trebuie să utilizați opțiunea `„-n”` pentru comanda `mount` când vă aflați în sistemul de fișiere rădăcină numai pentru citire.

3.1.4 Etapa 4: sistemul Debian normal

Sistemul Debian normal este a patra etapă a procesului de pornire, care este inițiat de sistemul mini-Debian. Nucleul sistemului pentru sistemul mini-Debian continuă să ruleze în acest mediu. Sistemul de fișiere rădăcină este comutat de la cel din memorie la cel de pe sistemul de fișiere al discului dur real.

Programul [init](#) este executat ca primul program cu `PID=1` pentru a efectua procesul principal de pornire a mai multor programe. Ruta implicită a fișierului pentru programul `init` este `„/usr/sbin/init”`, dar poate fi modificată prin parametrul de pornire al nucleului ca `„init=/ruta/către/programul-init”`.

`„/usr/sbin/init”` este legat simbolic de `„/lib/systemd/systemd”` după Debian 8 Jessie (lansat în 2015).

Indicație

Comanda `init` actuală din sistemul dvs. poate fi verificată cu comanda `«ps --pid 1 -f»`.

Indicație

Consultați [Debian wiki: BootProcessSpeedup](#) pentru cele mai recente sfaturi privind accelerarea procesului de pornire.

pachet	popcon(popularitate)	dimensiune	descriere
systemd	V:898, I:978	10634	demon bazat pe evenimente <code>init(8)</code> pentru execuție simultană (alternativă la <code>sysvinit</code>)
cloud-init	V:2, I:6	3231	sistem de inițializare pentru instanțele cloud de infrastructură
systemd-sysv	V:893, I:977	94	paginile de manual și legăturile necesare pentru ca <code>systemd</code> să înlocuiască <code>sysvinit</code>
init-system-helpers	V:910, I:985	133	instrumente auxiliare pentru comutarea între <code>sysvinit</code> și <code>systemd</code>
initscripts	V:19, I:75	203	scripturi pentru inițializarea și oprirea sistemului
sysvinit-core	V:3, I:4	369	instrumente System-V precum <code>init(8)</code>
sysv-rc	V:38, I:80	91	mecanism de schimbare a nivelului de rulare de tip System-V
sysvinit-utils	V:720, I:999	106	instrumente de tip System-V (<code>startpar(8)</code> , <code>bootlogd(8)</code> , ...)
lsb-base	V:266, I:385	12	Linux Standard Base 3.2 Funcționalitatea scriptului <code>init</code>
insserv	V:41, I:79	132	instrument pentru organizarea secvenței de pornire utilizând dependențele scriptului <code>LSB init.d</code>
kexec-tools	V:1, I:5	320	instrumentul <code>kexec</code> pentru reporniri <code>kexec(8)</code> (repornire la cald)
systemd-bootchart	V:0, I:0	131	analizator de performanță al procesului de pornire
mingetty	V:0, I:2	36	numai consolă <code>getty(8)</code>
mgetty	V:0, I:0	315	modem inteligent <code>getty(8)</code> înlocuitor

Tabela 3.3: Lista instrumentelor de pornire pentru sistemul Debian

3.2 Systemd

3.2.1 Init systemd

Când sistemul Debian pornește, `/usr/sbin/init` legat simbolic la `/usr/lib/systemd` este pornit ca proces al sistemului `init` (`PID=1`) deținut de `root` (`UID=0`). Vedeți `systemd(1)`.

Procesul de inițializare `systemd` generează procese în paralel pe baza fișierelor de configurare a unităților (vezi `systemd.unit(5)`), care sunt scrise în stil declarativ, în loc de stilul procedural de tip `SysV`.

Procesele generate sunt plasate în [grupuri de control Linux](#) individuale, denumite după unitatea căreia aparțin în ierarhia privată `systemd` (a se vedea [cgroups](#) și Secțiune 4.7.5).

Unitățile pentru modul sistem sunt încărcate din „Ruta de căutare a unităților sistemului” descrisă în `systemd.unit(5)`. Cele principale sunt următoarele, în ordinea priorității:

- `„/etc/systemd/system/*”`: Unități de sistem create de administrator
- `„/run/systemd/system/*”`: Unități de execuție
- `„/lib/systemd/system/*”`: Unități de sistem instalate de gestionarul de pachete al distribuției

Interdependențele lor sunt specificate de directivele „Wants=”, „Requires=”, „Before=”, „After=”, ... (a se vedea „MAPPING OF UNIT PROPERTIES TO THEIR INVERSES” în `systemd.unit(5)`). De asemenea, sunt definite și controalele resurselor (a se vedea `systemd.resource-control(5)`).

Sufixul fișierului de configurare al unității codifică tipurile acestora astfel:

- ***.service** descrie procesul controlat și supravegheat de `systemd`. Consultați `systemd.service(5)`.
- ***.device** descrie dispozitivul expus în `sysfs(5)` ca arbore de dispozitive `udev(7)`. Consultați `systemd.device(5)`.
- ***.mount** descrie punctul de montare al sistemului de fișiere controlat și supravegheat de `systemd`. Consultați `systemd.mount(5)`.

- ***.automount** descrie punctul de montare automată a sistemului de fișiere controlat și supravegheat de `systemd`. Consultați `systemd.automount(5)`.
- ***.swap** descrie dispozitivul sau fișierul spațiului de interschimb (swap) controlat și supravegheat de `systemd`. Consultați `systemd.swap(5)`.
- ***.path** descrie ruta monitorizată de `systemd` pentru activarea bazată pe rută. Consultați `systemd.path(5)`.
- ***.socket** descrie soclul controlat și supravegheat de `systemd` pentru activarea bazată pe soclu. Consultați `systemd.socket(5)`.
- ***.timer** descrie temporizatorul controlat și supravegheat de `systemd` pentru activarea bazată pe temporizator. Consultați `systemd.timer(5)`.
- ***.slice** gestionează resursele cu ajutorul `cgroups(7)`. Consultați `systemd.slice(5)`.
- ***.scope** este creat programatic folosind interfețele de magistrală ale `systemd` pentru a gestiona un set de procese de sistem. Consultați `systemd.scope(5)`.
- ***.target** grupează alte fișiere de configurare ale unității pentru a crea punctul de sincronizare în timpul pornirii. Consultați `systemd.target(5)`.

La pornirea sistemului (adică, `init`), procesul `systemd` încearcă să pornească „`/lib/systemd/system/default.target`” (în mod normal, legat simbolic la „`graphical.target`”). Mai întâi, unele unități țintă speciale (vezi `systemd.special(7)`), cum ar fi „`local-fs.target`”, „`swap.target`” și „`cryptsetup.target`”, sunt introduse pentru a monta sistemele de fișiere. Apoi, alte unități țintă sunt, de asemenea, introduse de dependențele unității țintă. Pentru detalii, citiți `bootup(7)`.

`systemd` oferă funcții de compatibilitate cu versiunile anterioare. Scripturile de pornire în stil SysV din „`/etc/init.d/rc[0-6]`” sunt în continuare analizate, iar `telinit(8)` este tradus în cereri de activare a unităților `systemd`.

**Atenție**

Nivelurile de execuție emulate 2 până la 4 sunt toate legate prin legături simbolice la aceeași țintă „`multi-user.target`”.

3.2.2 Autentificarea cu `systemd`

Când un utilizator se conectează la sistemul Debian prin `gdm3(8)`, `sshd(8)` etc., `/lib/systemd/system --user` este pornit ca proces de gestionare a serviciilor utilizatorului, deținut de utilizatorul corespunzător. Consultați `systemd(1)`.

Procesul de gestionare a serviciilor utilizatorului `systemd` generează procese în paralel pe baza fișierelor de configurare declarative ale unităților (consultați `systemd.unit(5)` și `user@.service(5)`).

Unitățile pentru modul utilizator sunt încărcate din „Ruta de căutare a unităților utilizatorului” descrisă în `systemd.unit(5)`. Cele principale sunt următoarele, în ordinea priorității:

- „`~/ .config/systemd/user/*`”: Unități de configurare utilizator
- „`/etc/systemd/user/*`”: Unități de utilizator create de administrator
- „`/run/systemd/user/*`”: Unități de execuție
- „`/lib/systemd/user/*`”: Unități utilizator instalate de gestionarul de pachete al distribuției

Acestea sunt gestionate în același mod ca Secțiune [3.2.1](#).

valoarea nivelului de eroare	numele nivelului de eroare	semnificație
0	KERN_EMERG	sistemul este inutilizabil
1	KERN_ALERT	trebuie să se ia imediat măsuri
2	KERN_CRIT	condiții critice
3	KERN_ERR	condiții de eroare
4	KERN_WARNING	condiții de avertizare
5	KERN_NOTICE	condiție normală, dar semnificativă
6	KERN_INFO	informativ
7	KERN_DEBUG	mesaje de nivel de depanare

Tabela 3.4: Lista nivelurilor de eroare ale nucleului

3.3 Mesajele nucleului

Mesajul de eroare al nucleului afișat pe consolă poate fi configurat prin definirea nivelului lui de prag.

```
# dmesg -n3
```

3.4 Mesajele sistemului

În cadrul `systemd`, atât mesajele nucleului, cât și cele ale sistemului sunt înregistrate de serviciul jurnal `systemd-journald` (cunoscut și sub numele de `journald`) fie într-un fișier binar persistent sub `„/var/log/journal”`, fie într-un fișier binar volatil sub `„/run/log/journal”`. Aceste date binare din jurnal sunt accesate prin comanda `journalctl(1)`. De exemplu, puteți afișa jurnalul de la ultima pornire astfel:

```
$ journalctl -b
```

Operația	Fragmente de comandă
Afișarea jurnalului pentru serviciile de sistem și nucleu de la ultima pornire	« <code>journalctl -b --system</code> »
Afișarea jurnalului pentru serviciile utilizatorului curent de la ultima pornire	« <code>journalctl -b --user</code> »
Afișarea jurnalului de activitate al „\$unit” de la ultima pornire	« <code>journalctl -b -u \$unit</code> »
Afișarea jurnalului de activități al „\$unit” (în stilul „ <code>tail -f</code> ”) de la ultima pornire	« <code>journalctl -b -u \$unit -f</code> »

Tabela 3.5: Lista fragmentelor tipice de comandă `journalctl`

În cadrul `systemd`, instrumentul de înregistrare a sistemului `rsyslogd(8)` poate fi dezinstalat. Dacă este instalat, acesta își modifică comportamentul pentru a citi datele volatile din jurnalul binar (în loc de `„/dev/log”` implicit înainte de `systemd`) și pentru a crea date tradiționale permanente ASCII din jurnalul sistemului. Acest lucru poate fi personalizat prin `„/etc/default/rsyslog”` și `„/etc/rsyslog.conf”` atât pentru fișierul jurnal, cât și pentru afișarea pe ecran. Consultați `rsyslogd(8)` și `rsyslog.conf(5)`. Consultați și Secțiune [9.3.2](#).

3.5 Gestionarea sistemului

Sistemul `systemd` oferă nu numai sistemul `init`, ci și operații generice de gestionare a sistemului cu comanda `systemctl(1)`.

Operația	Fragmente de comandă
Listează toate tipurile de unități disponibile	«systemctl list-units --type=help»
Listează toate unitățile țintă din memorie	«systemctl list-units --type=target»
Afișează toate unitățile de serviciu din memorie	«systemctl list-units --type=service»
Listează toate unitățile de dispozitiv din memorie	«systemctl list-units --type=device»
Listează toate unitățile de montare din memorie	«systemctl list-units --type=mount»
Afișează toate unitățile de soclu din memorie	«systemctl list-sockets»
Afișează toate unitățile de temporizare din memorie	«systemctl list-timers»
Pornește „\$unit”	«systemctl start \$unit»
Oprește „\$unit”	«systemctl stop \$unit»
Reîncarcă configurația specifică serviciului	«systemctl reload \$unit»
Oprește și pornește toate „\$unit”	«systemctl restart \$unit»
Pornește „\$unit” și oprește toate celelalte unități	«systemctl isolate \$unit»
Comută la „graphical” (sistem GUI, cu interfață grafică)	«systemctl isolate graphical»
Comută la „multi-user” (sistem CLI, cu interfață de linie de comandă)	«systemctl isolate multi-user»
Comută la „rescue” (sistem CLI, cu interfață de linie de comandă, mono-utilizator)	«systemctl isolate rescue»
Trimite semnalul de omorâre către „\$unit”	«systemctl kill \$unit»
Verifică dacă serviciul „\$unit” este activ	«systemctl is-active \$unit»
Verifică dacă serviciul „\$unit” a eșuat	«systemctl is-failed \$unit»
Verifică starea „\$unit \$PID dispozitiv”	«systemctl status \$unit \$PID \$device»
Afișează proprietățile „\$unit \$job”	«systemctl show \$unit \$job»
Reinițializează „\$unit” eșuată	«systemctl reset-failed \$unit»
Listează dependențele tuturor serviciilor de unitate	«systemctl list-dependencies --all»
Listează fișierele de unitate instalate în sistem	«systemctl list-unit-files»
Activează „\$unit” (adaugă legătură simbolică)	«systemctl enable \$unit»
Dezactivează „\$unit” (elimină legătura simbolică)	«systemctl disable \$unit»
Demască „\$unit” (elimină legătura simbolică către „/dev/null”)	«systemctl unmask \$unit»
Maschează „\$unit” (adaugă legătură simbolică la „/dev/null”)	«systemctl mask \$unit»
Obține configurația țintei implicite	«systemctl get-default»
Stabilește ținta implicită la „graphical” (sistem GUI, cu interfață grafică)	«systemctl set-default graphical»
Stabilește ținta implicită la „multi-user” (sistem CLI, cu interfață de linie de comandă)	«systemctl set-default multi-user»
Afișează mediul de lucru	«systemctl show-environment»
Definește „variabila” din mediul de lucru la „valoare”	«systemctl set-environment variabilă=valoare»
Elimină definiția (valoarea) „variabilei” din mediul de lucru	«systemctl unset-environment variabilă»
Reîncarcă toate fișierele de unitate și demonii	«systemctl daemon-reload»
Închide(oprește) sistemul	«systemctl poweroff»
Oprește și repornește sistemul	«systemctl reboot»
Suspendă sistemul	«systemctl suspend»
Hibernează sistemul	«systemctl hibernate»

Aici, „\$unit” din exemplele de mai sus poate fi un singur nume de unitate (sufixe precum `.service` și `.target` sunt opționale) sau, în multe cazuri, mai multe specificații de unități (modele globale de tip shell „*”, „?”, „[]” folosind `fnmatch(3)` care vor fi comparate cu numele primare ale tuturor unităților aflate în prezent în memorie).

Comenzile de modificare a stării sistemului din exemplele de mai sus sunt precedate de obicei de „sudo” pentru a obține privilegiile administrative necesare.

Rezultatul comenzii `systemctl status $unit|$PID|$device` utilizează culoarea punctului („●”) pentru a rezuma starea unității dintr-o privire.

- Culoarea albă a „●” indică starea „inactivă” sau „dezactivată”.
- Culoarea roșie a „●” indică o stare „eșuată” sau „de eroare”.
- Culoarea verde a „●” indică starea „activă”, „reîncărcare” sau „de activare”.

3.6 Alte monitoare de sistem

Iată o listă cu alte fragmente de comenzi de monitorizare sub `systemd`. Vă rugăm să citiți paginile de manual relevante, inclusiv `cgroups(7)`.

Operația	Fragmente de comandă
Afișează timpul petrecut pentru fiecare etapă de inițializare	«systemd-analyze time»
Listează toate unitățile în funcție de timpul necesar pentru inițializare	«systemd-analyze blame»
Încarcă și detectează erorile din fișierul „\$unit”	«systemd-analyze verify \$unit»
Afișează informații succinte despre starea de rulare a utilizatorului sesiunii apelantului	«loginctl user-status»
Afișează informații succinte despre starea de rulare a sesiunii apelantului	«loginctl session-status»
Urmărește procesul de pornire prin cgroups	«systemd-cgls»
Urmărește procesul de pornire prin cgroups	«ps xawf -eo pid,user,cgroup,args»
Urmărește procesul de pornire prin cgroups	Citiți sysfs sub „/sys/fs/cgroup/”

Tabela 3.7: Lista altor fragmente de comenzi de monitorizare sub `systemd`

3.7 Configurația sistemului

3.7.1 Numele gazdei

Nucleul menține **numele de gazdă** al sistemului. Unitatea de sistem pornită de `systemd-hostnamed.service` definește numele de gazdă al sistemului la pornire ca fiind numele stocat în „/etc/hostname”. Acest fișier trebuie să conțină **numai** numele de gazdă al sistemului, nu un nume de domeniu complet calificat.

Pentru a imprima numele actual al gazdei, executați `hostname(1)` fără niciun argument.

3.7.2 Sistemul de fișiere

Opțiunile de montare ale sistemelor de fișiere normale de disc și de rețea sunt definite în „/etc/fstab”. Consultați `fstab(5)` și Secțiune 9.6.7.

Configurația sistemului de fișiere criptat este definită în „/etc/crypttab”. Consultați `crypttab(5)`

Configurația RAID software cu `mdadm(8)` este definită în „/etc/mdadm/mdadm.conf”. Consultați `mdadm.conf(5)`.



Avertisment

După montarea tuturor sistemelor de fișiere, fișierele temporare din „/tmp”, „/var/lock” și „/var/run” sunt șterse la fiecare pornire.

3.7.3 Inițializarea interfeței de rețea

Interfețele de rețea sunt inițializate de obicei din „networking.service” pentru interfața `lo` și din „NetworkManager.service” pentru alte interfețe în sistemele Debian moderne sub `systemd`.

Consultați Cap. 5 pentru a afla cum să le configurați.

3.7.4 Inițializarea sistemului cloud

Instanța sistemului cloud poate fi lansată ca o clonă a „[Imaginilor cloud oficiale Debian](#)” sau a unor imagini similare. Pentru o astfel de instanță de sistem, caracteristici precum numele gazdei, sistemul de fișiere, rețeaua, configurația regională, cheile SSH, utilizatorii și grupurile pot fi configurate folosind funcționalitățile oferite de pachetele `cloud-init` și `netplan.io` cu mai multe surse de date, cum ar fi fișierele plasate în imaginea originală a sistemului și datele externe furnizate în timpul lansării sale. Aceste pachete permit configurarea declarativă a sistemului utilizând date [YAML](#).

Pentru mai multe informații, consultați „[Cloud Computing cu Debian și descendenții săi](#)”, „[Documentația Cloud-init](#)” și Secțiune 5.4.

3.7.5 Exemplu de personalizare pentru ajustarea serviciului sshd

Cu instalarea implicită, multe servicii de rețea (a se vedea Cap. 6) sunt pornite ca procese demon după `network.target` la pornire de către `systemd`. „sshd” nu face excepție. Să schimbăm acest lucru în pornirea la cerere a „sshd” ca exemplu de personalizare.

Mai întâi, dezactivați unitatea de serviciu instalată în sistem.

```
$ sudo systemctl stop sshd.service
$ sudo systemctl mask sshd.service
```

Sistemul de activare la cerere a soclurilor serviciilor Unix clasice se realiza prin intermediul superserverului `inetd` (sau `xinetd`). În cadrul `systemd`, echivalentul poate fi activat prin adăugarea fișierelor de configurare a unităților ***.socket** și ***.service**.

`sshd.socket` pentru specificarea unui soclu pe care să se asculte

```
[Unit]
Description=SSH Socket for Per-Connection Servers

[Socket]
ListenStream=22
Accept=yes
```

```
[Install]
WantedBy=sockets.target
```

sshd@.service ca fișier de serviciu corespunzător pentru sshd.socket

```
[Unit]
Description=SSH Per-Connection Server

[Service]
ExecStart=-/usr/sbin/sshd -i
StandardInput=socket
```

Apoi reîncărcați.

```
$ sudo systemctl daemon-reload
```

3.8 Sistemul udev

Sistemul [udev](#) oferă un mecanism pentru detectarea și inițializarea automată a hardware-ului (a se vedea [udev\(7\)](#)) începând cu nucleul Linux 2.6. La detectarea fiecărui dispozitiv de către nucleu, sistemul udev pornește un proces de utilizator care folosește informații din sistemul de fișiere [sysfs](#) (a se vedea [Secțiune 1.2.12](#)), încarcă modulele de nucleu necesare utilizând programul [modprobe\(8\)](#) (a se vedea [Secțiune 3.9](#)) și creează nodurile de dispozitiv corespunzătoare.

Indicație

Dacă „`/lib/modules/versiunea-nucleului/modules.dep`” nu a fost generat corect de [depmod\(8\)](#) din anumite motive, este posibil ca modulele să nu fie încărcate așa cum se așteaptă sistemul udev. Executați „`depmod -a`” pentru a remedia problema.

Pentru regulile de montare din „`/etc/fstab`”, nodurile dispozitivelor nu trebuie să fie statice. Puteți utiliza [UUID](#) pentru a monta dispozitive în locul numelor de dispozitive, cum ar fi „`/dev/sda`”. Consultați [Secțiune 9.6.3](#).

Deoarece sistemul udev este o țintă în continuă schimbare, las detaliile pentru alte documentații și descriu aici doar informațiile minimale.



Avertisment

Nu încercați să rulați programe de lungă durată, cum ar fi scriptul de copie de rezervă, cu `RUN` în regulile udev, așa cum se menționează în [udev\(7\)](#). Creați în schimb un fișier `systemd.service(5)` adecvat și activați-l. Consultați [Secțiune 10.2.3.2](#).

3.9 Inițializarea modulelor de nucleu

Programul [modprobe\(8\)](#) ne permite să configurăm nucleul Linux în execuție din procesul utilizatorului prin adăugarea și eliminarea modulelor nucleului. Sistemul udev (a se vedea [Secțiune 3.8](#)) automatizează invocarea acestuia pentru a ajuta la inițializarea modulului nucleului.

Există module non-hardware și module speciale de controlor hardware, precum cele enumerate mai jos, care trebuie preîncărcate prin listarea lor în fișierul „`/etc/modules`” (consultați [modules\(5\)](#)).

- Modulele [TUN/TAP](#) furnizează dispozitive de rețea virtuale punct-la-punct (TUN) și dispozitive de rețea Ethernet virtuale (TAP),
-

- modulele [netfilter](#) care oferă funcționalități de paravan de protecție netfilter (`iptables(8)`, Secțiune [5.7](#)), și
- și modulul controlorului [watchdog timer](#).

Fișierele de configurare pentru programul `modprobe(8)` se află în directorul „`/etc/modprobes.d/`”, așa cum se explică în `modprobe.conf(5)`. (Dacă doriți să evitați încărcarea automată a anumitor module ale nucleului, luați în considerare adăugarea acestora pe lista neagră din fișierul „`/etc/modprobes.d/blacklist`”.)

Fișierul „`/lib/modules/versiune/modules.dep`” generat de programul `depmod(8)` descrie dependențele modulelor utilizate de programul `modprobe(8)`.

Notă

Dacă întâmpinați probleme la încărcarea modulelor la pornirea sistemului sau cu `modprobe(8)`, comanda „`depmod -a`” poate rezolva aceste probleme prin reconstruirea fișierului „`modules.dep`”.

Programul `modinfo(8)` afișează informații despre un modul al nucleului Linux.

Programul `lsmod(8)` formatează conținutul „`/proc/modules`”, arătând ce module ale nucleului sunt încărcate în prezent.

Indicație

Puteți identifica hardware-ul exact din sistemul dvs. Consultați Secțiune [9.5.3](#).

Puteți configura hardware-ul la pornire pentru a activa funcțiile hardware dorite. Consultați Secțiune [9.5.4](#).

Probabil puteți adăuga suport pentru dispozitivul dvs. special prin recompilarea nucleului. Consultați Secțiune [9.10](#).

Capitolul 4

Autentificare și controale de acces

Când o persoană (sau un program) solicită acces la sistem, autentificarea confirmă identitatea ca fiind una de încredere.



Avertisment

Erorile de configurare ale PAM vă pot bloca accesul la propriul sistem. Trebuie să aveți la îndemână un CD de recuperare sau să configurați o partiție de pornire alternativă. Pentru a recupera sistemul, porniți-l cu ajutorul acestora și remediați problemele de acolo.

4.1 Autentificare normală Unix

Autentificarea normală Unix este asigurată de modulul `pam_unix(8)` din cadrul [PAM \(Pluggable Authentication Modules\)](#). Cele trei fișiere de configurare importante, cu intrări separate prin „:”, sunt următoarele.

fișier	permisiuni	utilizator	grupul	descriere
<code>/etc/passwd</code>	<code>-rw-r--r--</code>	<code>root</code>	<code>root</code>	informații despre contul utilizatorului (curățate)
<code>/etc/shadow</code>	<code>-rw-r-----</code>	<code>root</code>	<code>shadow</code>	informații securizate despre contul utilizatorului
<code>/etc/group</code>	<code>-rw-r--r--</code>	<code>root</code>	<code>root</code>	informații despre grup

Tabela 4.1: 3 fișiere de configurare importante pentru `pam_unix(8)`

„`/etc/passwd`” conține următoarele.

```
...
user1:x:1000:1000:User1 Name,,,:/home/user1:/bin/bash
user2:x:1001:1001:User2 Name,,,:/home/user2:/bin/bash
...
```

Așa cum se explică în `passwd(5)`, fiecare intrare separată prin „:” din acest fișier înseamnă următoarele.

- Nume de autentificare
- Intrarea specificării parolei
- ID-ul numeric al utilizatorului

- ID-ul numeric al grupului
- Nume utilizator sau câmp comentariu
- Directorul personal al utilizatorului
- Interpret opțional de comenzi pentru utilizator

A doua intrare din „/etc/passwd” a fost utilizată pentru introducerea parolei criptate. După introducerea „/etc/shadow”, această intrare este utilizată pentru introducerea specificațiilor parolei.

conținut	semnificație
(gol)	cont fără parolă
x	parola criptată se află în „/etc/shadow”

Tabela 4.2: Al doilea conținut al intrării „/etc/passwd”

Fișierul „/etc/shadow” conține următoarele câmpuri.

```
...
user1:$1$Xop0FYH9$IfxyQwBe9b8tiyIkt2P4F/:13262:0:99999:7:::
user2:$1$VGZLVbS$ElyErNf/agUDsm1DehJMS/:13261:0:99999:7:::
...
```

După cum se explică în shadow(5), fiecare intrare separată prin „:” din acest fișier înseamnă următoarele.

- Nume de autentificare
- Parolă criptată (inițialele „\$1\$” indică utilizarea criptării MD5. Simbolul „*” indică faptul că nu există date de conectare.)
- Data ultimei modificări a parolei, exprimată ca număr de zile de la 1 ianuarie 1970
- Numărul de zile pe care utilizatorul va trebui să le aștepte înainte de a i se permite să își schimbe din nou parola
- Numărul de zile după care utilizatorul va trebui să își schimbe parola
- Numărul de zile înainte de expirarea parolei în care utilizatorul trebuie avertizat
- Numărul de zile după expirarea parolei în care parola ar trebui să fie încă acceptată
- Data expirării contului, exprimată ca număr de zile de la 1 ianuarie 1970
- ...

Fișierul „/etc/group” conține următoarele câmpuri.

```
group1:x:20:user1,user2
```

După cum se explică în group(5), fiecare intrare separată prin „:” din acest fișier are următoarea semnificație.

- Numele grupului
- Parolă criptată (nu este utilizată efectiv)
- ID-ul numeric al grupului
- Listă separată prin virgule a numelor de utilizatori

Notă

„/etc/gshadow” oferă o funcție similară cu „/etc/shadow” pentru „/etc/group”, dar nu este utilizat în mod real.

Notă

Apartenența efectivă a unui utilizator la un grup poate fi adăugată dinamic dacă se adaugă linia „auth optional pam_group.so” în „/etc/pam.d/common-auth” și se definește în „/etc/security/group.conf”. Consultați `pam_group(8)`.

Notă

Pachetul `base-passwd` conține o listă oficială a utilizatorilor și grupurilor: „/usr/share/doc/base-passwd/users-and-groups.html”.

4.2 Gestionarea informațiilor privind contul și parola

Iată câteva comenzi importante pentru gestionarea informațiilor contului.

comanda	funcție
<code>getent passwd nume-utilizator</code>	răsfoiește informațiile contului „ <i>nume-utilizator</i> ”
<code>getent shadow nume-utilizator</code>	răsfoiește informațiile contului ascuns al „ <i>nume-utilizator</i> ”
<code>getent group nume-grup</code>	răsfoiește informațiile despre grupul „ <i>nume-grup</i> ”
<code>passwd</code>	gestionează parola pentru cont
<code>passwd -e</code>	definește o parolă unică pentru activarea contului
<code>chage</code>	gestionează informațiile privind expirarea parolelor

Tabela 4.3: Lista comenzilor pentru gestionarea informațiilor contului

Este posibil să aveți nevoie de privilegii de root pentru ca unele funcții să funcționeze. Consultați `crypt(3)` pentru criptarea parolelor și a datelor.

Notă

În sistemul configurat cu PAM și NSS ca mașină Debian [salsa](#), conținutul local „/etc/passwd”, „/etc/group” și „/etc/shadow” poate să nu fie utilizat în mod activ de sistem. Comenzile de mai sus sunt valabile chiar și în astfel de medii.

4.3 Parolă bună

Când creați un cont în timpul instalării sistemului sau cu comanda `passwd(1)`, ar trebui să alegeți o [parolă bună](#) care să conțină cel puțin 6-8 caractere, inclusiv unul sau mai multe caractere din fiecare dintre următoarele seturi, conform `passwd(1)`.

- literele alfabetului în minusculă
- cifrele de la 0 la 9
- Semne de punctuație

**Avertisment**

Nu alegeți cuvinte ușor de ghicit pentru parolă. Numele contului, numărul de asigurare socială, numărul de telefon, adresa, data nașterii, numele membrilor familiei sau al animalelor de companie, cuvinte din dicționar, secvențe simple de caractere precum „12345” sau „qwerty” etc. sunt toate alegeri nepotrivite pentru parolă.

4.4 Crearea unei parole criptate

Există instrumente independente pentru a [genera parole criptate cu „salt” \(sare\)](#).

pachet	popcon(popularitate)	dimensiune	comanda	funcție
whois	V:21, I:215	384	mkpasswd	interfață cu funcții excesive pentru biblioteca crypt(3)
openssl	V:843, I:996	2493	openssl passwd	calculează rezumatele criptografice ale parolelor (OpenSSL). passwd(1ssl)

Tabela 4.4: Lista instrumentelor pentru generarea parolei

4.5 PAM și NSS

Modern [Unix-like](#) systems such as the Debian system provide [PAM \(Pluggable Authentication Modules\)](#) and [NSS \(Name Service Switch\)](#) mechanism to the local system administrator to configure his system. The role of these can be summarized as the following.

- PAM offers a flexible authentication mechanism used by the application software thus involves password data exchange.
- NSS offers a flexible name service mechanism which is frequently used by the [C standard library](#) to obtain the user and group name for programs such as `ls(1)` and `id(1)`.

These PAM and NSS systems need to be configured consistently.

The notable packages of PAM and NSS systems are the following.

- "The Linux-PAM System Administrators' Guide" in `libpam-doc` is essential for learning PAM configuration.
- "System Databases and Name Service Switch" section in `glibc-doc-reference` is essential for learning NSS configuration.

Notă

You can see more extensive and current list by "`aptitude search 'libpam-|libnss-'`" command. The acronym NSS may also mean "Network Security Service" which is different from "Name Service Switch".

Notă

PAM is the most basic way to initialize environment variables for each program with the system wide default value.

Under [systemd](#), `libpam-systemd` package is installed to manage user logins by registering user sessions in the `systemd` control group hierarchy for [logind](#). See `systemd-logind(8)`, `logind.conf(5)`, and `pam_systemd(8)`.

pachet	popcon(popularitate)	descriere
libpam-modules	V:929, I:999 923	Pluggable Authentication Modules (basic service)
libpam-ldapd	V:5, I:14 80	Pluggable Authentication Module allowing LDAP interfaces
libpam-systemd	V:686, I:961 739	Pluggable Authentication Module to register user sessions for logind
libpam-doc	I:6 1504	Pluggable Authentication Modules (documentation in html and text)
libc6	V:932, I:999 5678	GNU C Library: Shared libraries which also provides "Name Service Switch" service
glibc-doc	I:6 3857	GNU C Library: Manpages
glibc-doc-reference	I:3 14261	GNU C Library: Reference manual in info, pdf and html format (non-free)
libnss-mdns	V:209, I:516 141	NSS module for Multicast DNS name resolution
libnss-ldapd	V:6, I:16 131	NSS module for using LDAP as a naming service

Tabela 4.5: List of notable PAM and NSS systems

4.5.1 Configuration files accessed by PAM and NSS

Here are a few notable configuration files accessed by PAM and NSS.

fișier de configurare	funcție
<code>/etc/pam.d/nume-program</code>	set up PAM configuration for the " <i>program_name</i> " program; see <code>pam(7)</code> and <code>pam.d(5)</code>
<code>/etc/nsswitch.conf</code>	set up NSS configuration with the entry for each service. See <code>nsswitch.conf(5)</code>
<code>/etc/nologin</code>	limit the user login by the <code>pam_nologin(8)</code> module
<code>/etc/securetty</code>	limit the tty for the root access by the <code>pam_securetty(8)</code> module
<code>/etc/security/access.conf</code>	set access limit by the <code>pam_access(8)</code> module
<code>/etc/security/group.conf</code>	set group based restraint by the <code>pam_group(8)</code> module
<code>/etc/security/pam_env.conf</code>	set environment variables by the <code>pam_env(8)</code> module
<code>/etc/environment</code>	set additional environment variables by the <code>pam_env(8)</code> module with the " <code>readenv=1</code> " argument
<code>/etc/default/locale</code>	set locale by <code>pam_env(8)</code> module with the " <code>readenv=1 envfile=/etc/default/locale</code> " argument (Debian)
<code>/etc/security/limits.conf</code>	set resource restraint (<code>ulimit</code> , <code>core</code> , ...) by the <code>pam_limits(8)</code> module
<code>/etc/security/time.conf</code>	set time restraint by the <code>pam_time(8)</code> module
<code>/etc/systemd/logind.conf</code>	set systemd login manager configuration (see <code>logind.conf(5)</code> and <code>systemd-logind.service(8)</code>)

Tabela 4.6: Lista fișierelor de configurare accesate de PAM și NSS

The limitation of the password selection is implemented by the PAM modules, `pam_unix(8)` and `pam_cracklib(8)`. They can be configured by their arguments.

Indicație

PAM modules use suffix ".so" for their filenames.

4.5.2 The modern centralized system management

The modern centralized system management can be deployed using the centralized [Lightweight Directory Access Protocol \(LDAP\)](#) server to administer many Unix-like and non-Unix-like systems on the network. The open source implementation of the Lightweight Directory Access Protocol is [OpenLDAP Software](#).

The LDAP server provides the account information through the use of PAM and NSS with `libpam-ldapd` and `libnss-ldapd` packages for the Debian system. Several actions are required to enable this (I have not used this setup and the following is purely secondary information. Please read this in this context.).

- You set up a centralized LDAP server by running a program such as the stand-alone LDAP daemon, `slapd(8)`.
- You change the PAM configuration files in the `/etc/pam.d/` directory to use `"pam_ldap.so"` instead of the default `"pam_unix.so"`.
- You change the NSS configuration in the `/etc/nsswitch.conf` file to use `"ldap"` instead of the default (`"compat"` or `"file"`).
- You must make `libpam-ldapd` to use [SSL \(or TLS\)](#) connection for the security of password.
- You may make `libnss-ldapd` to use [SSL \(or TLS\)](#) connection to ensure integrity of data at the cost of the LDAP network overhead.
- You should run `nscd(8)` locally to cache any LDAP search results in order to reduce the LDAP network traffic.

See documentations in `nsswitch.conf(5)`, `pam.conf(5)`, `ldap.conf(5)`, and `/usr/share/doc/libpam-doc/html/` offered by the `libpam-doc` package and `"info libc 'Name Service Switch'"` offered by the `glibc-doc` package.

Similarly, you can set up alternative centralized systems with other methods.

- Integration of user and group with the Windows system.
 - Access [Windows domain](#) services by the `winbind` and `libpam_winbind` packages.
 - See `winbindd(8)` and [Integrating MS Windows Networks with Samba](#).
- Integration of user and group with the legacy Unix-like system.
 - Access [NIS \(originally called YP\)](#) or [NIS+](#) by the `nis` package.
 - See [The Linux NIS\(YP\)/NIS/NIS+ HOWTO](#).

4.5.3 "Why GNU su does not support the wheel group"

This is the famous phrase at the bottom of the old `"info su"` page by Richard M. Stallman. Not to worry: the current `su` command in Debian uses PAM, so that one can restrict the ability to use `su` to the `root` group by enabling the line with `"pam_wheel.so"` in `/etc/pam.d/su`.

4.5.4 Stricter password rule

Installing the `libpam-cracklib` package enables you to force stricter password rule.

On a typical GNOME system which automatically installs `libpam-gnome-keyring`, `/etc/pam.d/common-password` looks like:

```
# here are the per-package modules (the "Primary" block)
password requisite pam_cracklib.so retry=3 minlen=8 difok=3
password [success=1 default=ignore] pam_unix.so obscure use_authtok try_first_pass ↔
    yescrypt
# here's the fallback if no module succeeds
password requisite pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password required pam_permit.so
# and here are more per-package modules (the "Additional" block)
password optional pam_gnome_keyring.so
# end of pam-auth-update config
```

4.6 Security of authentication

Notă

The information here **may not be sufficient** for your security needs but it should be a **good start**.

4.6.1 Secure password on the Internet

Many popular transportation layer services communicate messages including password authentication in the plain text. It is very bad idea to transmit password in the plain text over the wild Internet where it can be intercepted. You can run these services over "[Transport Layer Security](#)" (TLS) or its predecessor, "Secure Sockets Layer" (SSL) to secure entire communication including password by the encryption.

insecure service name	port	secure service name	port
www (http)	80	https	443
smtp (poștă electronică)	25	ssmtp (smtps)	465
ftp-data	20	ftps-data	989
ftp	21	ftps	990
telnet	23	telnets	992
imap2	143	imaps	993
pop3	110	pop3s	995
ldap	389	ldaps	636

Tabela 4.7: List of insecure and secure services and ports

The encryption costs CPU time. As a CPU friendly alternative, you can keep communication in plain text while securing just the password with the secure authentication protocol such as "Authenticated Post Office Protocol" (APOP) for POP and "Challenge-Response Authentication Mechanism MD5" (CRAM-MD5) for SMTP and IMAP. (For sending mail messages over the Internet to your mail server from your mail client, it is recently popular to use new message submission port 587 instead of traditional SMTP port 25 to avoid port 25 blocking by the network provider while authenticating yourself with CRAM-MD5.)

4.6.2 Shell securizat

The [Secure Shell \(SSH\)](#) program provides secure encrypted communications between two untrusted hosts over an insecure network with the secure authentication. It consists of the [OpenSSH](#) client, `ssh(1)`, and the [OpenSSH](#)

daemon, sshd(8). This SSH can be used to tunnel an insecure protocol communication such as POP and X securely over the Internet with the port forwarding feature.

The client tries to authenticate itself using host-based authentication, public key authentication, challenge-response authentication, or password authentication. The use of public key authentication enables the remote password-less login. See Secțiune 6.3.

4.6.3 Extra security measures for the Internet

Even when you run secure services such as [Secure Shell \(SSH\)](#) and [Point-to-point tunneling protocol \(PPTP\)](#) servers, there are still chances for the break-ins using brute force password guessing attack etc. from the Internet. Use of the firewall policy (see Secțiune 5.7) together with the following security tools may improve the security situation.

pachet	popcon(popularity)	drum (times)	descriere
knockd	V:0, I:1	110	small port-knock daemon knockd(1) and client knock(1)
fail2ban	V:97, I:108	2191	ban IPs that cause multiple authentication errors
libpam-shield	V:0, I:0	115	lock out remote attackers trying password guessing

Tabela 4.8: List of tools to provide extra security measures

4.6.4 Securing the root password

To prevent people to access your machine with root privilege, you need to make following actions.

- Prevent physical access to the hard disk
- Lock UEFI/BIOS and prevent booting from the removable media
- Set password for GRUB interactive session
- Lock GRUB menu from editing

With physical access to hard disk, resetting the password is relatively easy with following steps.

1. Move the hard disk to a PC with CD bootable UEFI/BIOS.
2. Boot system with a rescue media (Debian boot disk, Knoppix CD, GRUB CD, ...).
3. Mount root partition with read/write access.
4. Edit `/etc/passwd` in the root partition and make the second entry for the root account empty.

If you have edit access to the GRUB menu entry (see Secțiune 3.1.2) for grub-rescue-pc at boot time, it is even easier with following steps.

1. Boot system with the kernel parameter changed to something like `"root=/dev/hda6 rw init=/bin/sh"`.
2. Edit `/etc/passwd` and make the second entry for the root account empty.
3. Reboot system.

The root shell of the system is now accessible without password.

Notă

Once one has root shell access, he can access everything on the system and reset any passwords on the system. Further more, he may compromise password for all user accounts using brute force password cracking tools such as `john` and `crack` packages (see Secțiune 9.5.11). This cracked password may lead to compromise other systems.

The only reasonable software solution to avoid all these concerns is to use software encrypted root partition (or `/etc` partition) using `dm-crypt` and `initramfs` (see Secțiune 9.9). You always need password to boot the system, though.

4.7 Other access controls

There are access controls to the system other than the password based authentication and file permissions.

Notă

See Secțiune 9.4.16 for restricting the kernel [secure attention key \(SAK\)](#) feature.

4.7.1 Liste de control al accesului (ACL)

ACLs are a superset of the regular permissions as explained in Secțiune 1.2.3.

You encounter ACLs in action on modern desktop environment. When a formatted USB storage device is auto mounted as, e.g., `/media/penguin/USBSTICK`, a normal user `penguin` can execute:

```
$ cd /media/penguin
$ ls -la
total 16
drwxr-x---+ 1 root    root    16 Jan 17 22:55 .
drwxr-xr-x  1 root    root    28 Sep 17 19:03 ..
drwxr-xr-x  1 penguin penguin 18 Jan  6 07:05 USBSTICK
```

`+` in the 11th column indicates ACLs are in action. Without ACLs, a normal user `penguin` shouldn't be able to list like this since `penguin` isn't in `root` group. You can see ACLs as:

```
$ getfacl .
# file: .
# owner: root
# group: root
user::rwx
user:penguin:r-x
group::---
mask::r-x
other::---
```

Here:

- `"user::rwx"`, `"group::---"`, and `"other::---"` correspond to the regular owner, group, and other permissions.
- The ACL `"user:penguin:r-x"` allows a normal user `penguin` to have `"r-x"` permissions. This enabled `"ls -la"` to list directory content.
- The ACL `"mask::r-x"` sets the upper bound of permissions.

See ["POSIX Access Control Lists on Linux"](#), `acl(5)`, `getfacl(1)`, and `setfacl(1)` for more.

4.7.2 sudo

`sudo(8)` is a program designed to allow a sysadmin to give limited root privileges to users and log root activity. `sudo` requires only an ordinary user's password. Install `sudo` package and activate it by setting options in `/etc/sudoers`. See configuration example at `/usr/share/doc/sudo/examples/sudoers` and Secțiune 1.1.12.

My usage of `sudo` for the single user system (see Secțiune 1.1.12) is aimed to protect myself from my own stupidity. Personally, I consider using `sudo` a better alternative than using the system from the root account all the time. For example, the following changes the owner of `some_file` to `my_name`.

```
$ sudo chown my_name some_file
```

Of course if you know the root password (as self-installed Debian users do), any command can be run under root from any user's account using `su -c`.

4.7.3 PolicyKit

[PolicyKit](#) is an operating system component for controlling system-wide privileges in Unix-like operating systems.

Newer GUI applications are not designed to run as privileged processes. They talk to privileged processes via [PolicyKit](#) to perform administrative operations.

[PolicyKit](#) limits such operations to user accounts belonging to the `sudo` group on the Debian system.

A se vedea `polkit(8)`.

4.7.4 Restricting access to some server services

For system security, it is a good idea to disable as much server programs as possible. This becomes critical for network servers. Having unused servers, activated either directly as [daemon](#) or via [super-server](#) program, are considered security risks.

Many programs, such as `sshd(8)`, use PAM based access control. There are many ways to restrict access to some server services.

- configuration files: `/etc/default/program_name`
- Systemd service unit configuration for [daemon](#)
- [PAM \(Pluggable Authentication Modules\)](#)
- `/etc/inetd.conf` pentru [super-server](#)
- `/etc/hosts.deny` and `/etc/hosts.allow` for [TCP wrapper](#), `tcpd(8)`
- `/etc/rpc.conf` pentru [Sun RPC](#)
- `/etc/at.allow` și `/etc/at.deny` pentru `atd(8)`
- `/etc/cron.allow` și `/etc/cron.deny` pentru `crontab(1)`
- [Network firewall](#) of [netfilter](#) infrastructure

See Secțiune 3.5, Secțiune 4.5.1, and Secțiune 5.7.

Indicație

[Sun RPC](#) services need to be active for [NFS](#) and other RPC based programs.

Indicație

If you have problems with remote access in a recent Debian system, comment out offending configuration such as `"ALL: PARANOID"` in `/etc/hosts.deny` if it exists. (But you must be careful on security risks involved with this kind of action.)

4.7.5 Linux security features

Linux kernel has evolved and supports security features not found in traditional UNIX implementations.

Linux supports [extended attributes](#) which extend the traditional UNIX attributes (see `xattr(7)`).

Linux divides the privileges traditionally associated with superuser into distinct units, known as [capabilities\(7\)](#), which can be independently enabled and disabled. Capabilities are a per-thread attribute since kernel version 2.2.

The [Linux Security Module \(LSM\) framework](#) provides a [mechanism for various security checks](#) to be hooked by new kernel extensions. For example:

- [AppArmor](#)
- [Security-Enhanced Linux \(SELinux\)](#)
- [Smack \(Simplified Mandatory Access Control Kernel\)](#)
- [Tomoyo Linux](#)

Since these extensions may tighten privilege model tighter than the ordinary Unix-like security model policies, even the root power may be restricted. You are advised to read the [Linux Security Module \(LSM\) framework document at kernel.org](#).

Linux [namespaces](#) wrap a global system resource in an abstraction that makes it appear to the processes within the namespace that they have their own isolated instance of the global resource. Changes to the global resource are visible to other processes that are members of the namespace, but are invisible to other processes. Since kernel version 5.6, there are 8 kinds of namespaces (see `namespaces(7)`, `unshare(1)`, `nsenter(1)`).

As of Debian 11 Bullseye (2021), Debian uses unified cgroup hierarchy (a.k.a. [cgroups-v2](#)).

Usage examples of [namespaces](#) with [cgroups](#) to isolate their processes and to allow resource control are:

- [Systemd](#). A se vedea Secțiune [3.2.1](#).
- [Sandbox environment](#). See Secțiune [7.7](#).
- [Linux containers](#) such as [Docker](#), [LXC](#). See Secțiune [9.11](#).

These functionalities can't be realized by Secțiune [4.1](#). These advanced topics are mostly out-of-scope for this introductory document.

Capitolul 5

Configurarea rețelei

Indicație

For modern Debian specific guide to the networking, read [The Debian Administrator's Handbook — Configuring the Network](#).

Indicație

Under [systemd](#), [networkd](#) may be used to manage networks. See `systemd-networkd(8)`.

5.1 The basic network infrastructure

Let's review the basic network infrastructure on the modern Debian system.

5.1.1 Rezoluția numelui de gazdă

The hostname resolution is currently supported by the [NSS \(Name Service Switch\)](#) mechanism too. The flow of this resolution is the following.

1. The `/etc/nsswitch.conf` file with stanza like `hosts: files dns` dictates the hostname resolution order. (This replaces the old functionality of the `order` stanza in `/etc/host.conf`.)
2. The `files` method is invoked first. If the hostname is found in the `/etc/hosts` file, it returns all valid addresses for it and exits. (The `/etc/host.conf` file contains `multi on`.)
3. The `dns` method is invoked. If the hostname is found by the query to the [Internet Domain Name System \(DNS\)](#) identified by the `/etc/resolv.conf` file, it returns all valid addresses for it and exits.

A typical workstation may be installed with its host name set to, e.g., `host_name` and its optional domain name set to an empty string. Then, `/etc/hosts` looks like the following.

```
127.0.0.1 localhost
127.0.1.1 host_name

# The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-localhost ip6-loopback
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters
```

pachete	popcon(popularity)	dimensiune	tipul	descriere
network-manager	V:418, I:478	7809	config::NM	NetworkManager (daemon): manage the network automatically
network-manager-gnome	V:58, I:212	18	config::NM	NetworkManager (GNOME frontend)
netplan.io	V:1, I:7	340	config::NM+networkd	Netplan (generator): Unified, declarative interface between NetworkManager and systemd-networkd backends
ifupdown	V:599, I:973	200	config::ifupdown	standardized tool to bring up and down the network (Debian specific)
isc-dhcp-client	V:179, I:749	2884	config::low-level	DHCP client
pppoeconf	V:0, I:4	174	config::helper	configuration helper for PPPoE connection
wpasupplicant	V:386, I:521	3900	config::helper	client support for WPA and WPA2 (IEEE 802.11i)
wpaui	V:0, I:1	783	config::helper	Qt GUI client for wpa_supplicant
wireless-tools	V:187, I:259	292	config::helper	tools for manipulating Linux Wireless Extensions
iw	V:36, I:482	332	config::helper	tool for configuring Linux wireless devices
iproute2	V:748, I:983	3901	config::iproute2	iproute2 , IPv6 and other advanced network configuration: ip(8) , tc(8) , etc
iptables	V:344, I:636	2410	config::Netfilter	administration tools for packet filtering and NAT (Netfilter)
nftables	V:186, I:840	191	config::Netfilter	administration tools for packet filtering and NAT (Netfilter) (successor to {ip,ip6,arp,eb}tables)
iputils-ping	V:188, I:997	188	test	test network reachability of a remote host by hostname or IP address (iproute2)
iputils-arping	V:1, I:19	53	test	test network reachability of a remote host specified by the ARP address
iputils-tracepath	V:2, I:22	50	test	trace the network path to a remote host
ethtool	V:93, I:255	1068	test	display or change Ethernet device settings
mtr-tiny	V:4, I:40	181	test::low-level	trace the network path to a remote host (curses)
mtr	V:3, I:41	230	test::low-level	trace the network path to a remote host (curses and GTK)
gnome-nettool	V:0, I:10	2480	test::low-level	tools for common network information operations (GNOME)
nmap	V:24, I:189	4607	test::low-level	network mapper / port scanner (Nmap , console)
tcpdump	V:15, I:168	1343	test::low-level	network traffic analyzer (Tcpdump , console)
wireshark	V:3, I:41	11260	test::low-level	network traffic analyzer (Wireshark , GTK)
tshark	V:2, I:23	431	test::low-level	network traffic analyzer (console)
tcptrace	V:0, I:1	407	test::low-level	produce a summarization of the connections from tcpdump output
ntopng	V:0, I:0	15604	test::low-level	display network usage in web browser
dnsutils	I:186	22	test::low-level	network clients provided with BIND : nslookup(8) , nsupdate(8) , dig(8)
dlint	V:0, I:2	51	test::low-level	check DNS zone information using nameserver lookups
dnstracer	V:0, I:1	59	test::low-level	trace a chain of DNS servers to the source

Tabela 5.1: List of network configuration tools

Each line starts with a [IP address](#) and it is followed by the associated [hostname](#).

The IP address 127.0.1.1 in the second line of this example may not be found on some other Unix-like systems. The [Debian Installer](#) creates this entry for a system without a permanent IP address as a workaround for some software (e.g., GNOME) as documented in the [bug #719621](#).

The *host_name* matches the hostname defined in the `/etc/hostname` (see [Secțiune 3.7.1](#)).

For a system with a permanent IP address, that permanent IP address should be used here instead of 127.0.1.1.

For a system with a permanent IP address and a [fully qualified domain name \(FQDN\)](#) provided by the [Domain Name System \(DNS\)](#), that canonical *host_name.domain_name* should be used instead of just *host_name*.

The `/etc/resolv.conf` is a static file if the `resolvconf` package is not installed. If installed, it is a symbolic link. Either way, it contains information that initialize the resolver routines. If the DNS is found at IP="192.168.11.1", it contains the following.

```
nameserver 192.168.11.1
```

The `resolvconf` package makes this `/etc/resolv.conf` into a symbolic link and manages its contents by the hook scripts automatically.

For the PC workstation on the typical adhoc LAN environment, the hostname can be resolved via [Multicast DNS](#) (mDNS) in addition to the basic files and dns methods.

- [Avahi](#) provides a framework for Multicast DNS Service Discovery on Debian.
- It is equivalent of [Apple Bonjour](#) / [Apple Rendezvous](#).
- The `libnss-mdns` plugin package provides host name resolution via mDNS for the GNU Name Service Switch (NSS) functionality of the GNU C Library (glibc).
- The `/etc/nsswitch.conf` file should have stanza like `hosts: files mdns4_minimal [NOTFOUND=return] dns` (see `/usr/share/doc/libnss-mdns/README.Debian` for other configurations).
- A host name suffixed with the [".local"](#) [pseudo-top-level domain](#) is resolved by sending a mDNS query message in a multicast UDP packet using IPv4 address "224.0.0.251" or IPv6 address "FF02::FB".

Notă

The [expansion of generic Top-Level Domains \(gTLD\)](#) in the [Domain Name System](#) is underway. Watch out for the [name collision](#) when choosing a domain name used only within LAN.

Notă

Use of packages such as `libnss-resolve` together with `systemd-resolved`, or `libnss-myhostname`, or `libnss-mymachine`, with corresponding listings on the "hosts" line in the `/etc/nsswitch.conf` file may override the traditional network configuration discussed in the above. See `nss-resolve(8)`, `systemd-resolved(8)`, `nss-myhostname(8)`, and `nss-mymachines(8)` for more.

5.1.2 Numele interfeței de rețea

The `systemd` uses ["Predictable Network Interface Names"](#) such as `enp0s25`.

Clasa	adrese de rețea	masca de rețea	masca de rețea /biți	numărul de subrețele
A	10.x.x.x	255.0.0.0	/8	1
B	172.16.x.x — 172.31.x.x	255.255.0.0	/16	16
C	192.168.0.x — 192.168.255.x	255.255.255.0	/24	256

Tabela 5.2: List of network address ranges

5.1.3 Intervalul de adrese pentru rețeaua locală (LAN)

Let us be reminded of the IPv4 32 bit address ranges in each class reserved for use on the [local area networks \(LANs\)](#) by [rfc1918](#). These addresses are guaranteed not to conflict with any addresses on the Internet proper.

Notă

IP address written with colon are [IPv6 address](#), e.g., "::<1" for localhost.

Notă

If one of these addresses is assigned to a host, then that host must not access the Internet directly but must access it through a gateway that acts as a proxy for individual services or else does [Network Address Translation \(NAT\)](#). The broadband router usually performs NAT for the consumer LAN environment.

5.1.4 The network device support

Although most hardware devices are supported by the Debian system, there are some network devices which require [DFSG](#) non-free firmware to support them. Please see [Secțiune 9.10.5](#).

5.2 The modern network configuration for desktop

Interfețele de rețea sunt inițializate de obicei din „networking.service” pentru interfața `lo` și din „NetworkManager.service” pentru alte interfețe în sistemele Debian moderne sub `systemd`.

Debian can manage the network connection via management [daemon](#) software such as [NetworkManager \(NM\)](#) (network-manager and associated packages).

- They come with their own [GUI](#) and command-line programs as their user interfaces.
- They come with their own [daemon](#) as their backend system.
- They allow easy connection of your system to the Internet.
- They allow easy management of wired and wireless network configuration.
- They allow us to configure network independent of the legacy `ifupdown` package.

Notă

Do not use these automatic network configuration tools for servers. These are aimed primarily for mobile desktop users on laptops.

These modern network configuration tools need to be configured properly to avoid conflicting with the legacy `ifupdown` package and its configuration file `"/etc/network/interfaces"`.

5.2.1 GUI network configuration tools

Official documentations for NM on Debian are provided in `"/usr/share/doc/network-manager/README.Debian"`. Essentially, the network configuration for desktop is done as follows.

1. Make desktop user, e.g. `foo`, belong to group `"netdev"` by the following (Alternatively, do it automatically via [D-bus](#) under modern desktop environments such as GNOME and KDE).

```
$ sudo usermod -a -G netdev foo
```

2. Keep configuration of `"/etc/network/interfaces"` as simple as in the following.

```
auto lo
iface lo inet loopback
```

3. Restart NM by the following.

```
$ sudo systemctl restart NetworkManager
```

4. Configure your network via GUI.

Notă

Only interfaces which are **not** listed in `"/etc/network/interfaces"` are managed by NM to avoid conflict with `ifupdown`.

Indicație

If you wish to extend network configuration capabilities of NM, please seek appropriate plug-in modules and supplemental packages such as `network-manager-openconnect`, `network-manager-openvpn-gnome`, `network-manager-pptp-gnome`, `mobile-broadband-provider-info`, `gnome-bluetooth`, etc.

5.3 The modern network configuration without GUI

Under [systemd](#), the network may be configured in `/etc/systemd/network/` instead. See `systemd-resolved(8)`, `resolved.conf(5)`, and `systemd-networkd(8)`.

This allows the modern network configuration without GUI.

A DHCP client configuration can be set up by creating `"/etc/systemd/network/dhcp.network"`. E.g.:

```
[Match]
Name=en*

[Network]
DHCP=yes
```

A static network configuration can be set up by creating `"/etc/systemd/network/static.network"`. E.g.:

```
[Match]
Name=en*

[Network]
Address=192.168.0.15/24
Gateway=192.168.0.1
```

5.4 The modern network configuration for cloud

The modern network configuration for cloud may use `cloud-init` and `netplan.io` packages (see Secțiune 3.7.4). The `netplan.io` package supports `systemd-networkd` and `NetworkManager` as its network configuration backends, and enables the declarative network configuration using [YAML](#) data. When you change YAML:

- Run `netplan generate` command to generate all the necessary backend configuration from [YAML](#).
- Run `netplan apply` command to apply the generated configuration to the backends.

See ["Netplan documentation"](#), `netplan(5)`, `netplan-generate(8)`, and `netplan-apply(8)`.

See also ["Cloud-init documentation"](#) (especially around ["Configuration sources"](#) and ["Netplan Passthrough"](#)) for how `cloud-init` can integrate `netplan.io` configuration with alternative data sources.

5.4.1 The modern network configuration for cloud with DHCP

A DHCP client configuration can be set up by creating a data source file `/etc/netplan/50-dhcp.yaml`:

```
network:
  version: 2
  ethernets:
    all-en:
      match:
        name: "en*"
      dhcp4: true
      dhcp6: true
```

5.4.2 The modern network configuration for cloud with static IP

A static network configuration can be set up by creating a data source file `/etc/netplan/50-static.yaml`:

```
network:
  version: 2
  ethernets:
    eth0:
      addresses:
        - 192.168.0.15/24
      routes:
        - to: default
          via: 192.168.0.1
```

5.4.3 The modern network configuration for cloud with Network Manager

The network client configuration using Network Manager infrastructure can be set up by creating a data source file `/etc/netplan/00-network-manager.yaml`:

```
network:
  version: 2
  renderer: NetworkManager
```

5.5 The low level network configuration

For the low level network configuration on Linux, use the [iproute2](#) programs (`ip(8)`, ...).

5.5.1 Comenzi «iproute2»

[iproute2](#) commands offer complete low-level network configuration capabilities. Here is a translation table from obsolete [net-tools](#) commands to new [iproute2](#) etc. commands.

obsolete net-tools	new iproute2 etc.	manipulation
<code>ifconfig(8)</code>	<code>ip addr</code>	protocol (IP or IPv6) address on a device
<code>route(8)</code>	<code>ip route</code>	routing table entry
<code>arp(8)</code>	<code>ip neigh</code>	ARP or NDISC cache entry
<code>ipmaddr</code>	<code>ip maddr</code>	multicast address
<code>iptunnel</code>	<code>ip tunnel</code>	tunnel over IP
<code>nameif(8)</code>	<code>ifrename(8)</code>	name network interfaces based on MAC addresses
<code>mii-tool(8)</code>	<code>ethtool(8)</code>	Ethernet device settings

Tabela 5.3: Translation table from obsolete `net-tools` commands to new `iproute2` commands

A se veea `ip(8)` și [Linux Advanced Routing & Traffic Control](#).

5.5.2 Safe low level network operations

You may use low level network commands as follows safely since they do not change network configuration.

comanda	descriere
<code>ip addr show</code>	display the link and address status of active interfaces
<code>route -n</code>	display all the routing table in numerical addresses
<code>ip route show</code>	display all the routing table in numerical addresses
<code>arp</code>	display the current content of the ARP cache tables
<code>ip neigh</code>	display the current content of the ARP cache tables
<code>plog</code>	display ppp daemon log
<code>ping yahoo.com</code>	check the Internet connection to "yahoo.com"
<code>whois yahoo.com</code>	check who registered "yahoo.com" in the domains database
<code>traceroute yahoo.com</code>	trace the Internet connection to "yahoo.com"
<code>tracepath yahoo.com</code>	trace the Internet connection to "yahoo.com"
<code>mtr yahoo.com</code>	trace the Internet connection to "yahoo.com" (repeatedly)
<code>dig [@dns-server.com] example.com [{a mx any}]</code>	check DNS records of "example.com" by "dns-server.com" for a "a", "mx", or "any" record
<code>iptables -L -n</code>	check packet filter
<code>netstat -a</code>	find all open ports
<code>netstat -l --inet</code>	find listening ports
<code>netstat -ln --tcp</code>	find listening TCP ports (numeric)
<code>dlint example.com</code>	check DNS zone information of "example.com"

Tabela 5.4: List of low level network commands

Indicație

Some of these low level network configuration tools reside in `/usr/sbin/`. You may need to issue full command path such as `/usr/sbin/ifconfig` or add `/usr/sbin` to the `$PATH` list in your `~/.bashrc`.

5.6 Optimizarea rețelei

Generic network optimization is beyond the scope of this documentation. I touch only subjects pertinent to the consumer grade connection.

pachete	popcon	popularity	descriere
iftop	V:6, I:91	93	display bandwidth usage information on an network interface
iperf	V:2, I:37	427	Internet Protocol bandwidth measuring tool
ifstat	V:0, I:6	53	InterFace STATistics Monitoring
bmon	V:2, I:20	141	portable bandwidth monitor and rate estimator
ethstatus	V:0, I:2	41	script that quickly measures network device throughput
bing	V:0, I:0	80	empirical stochastic bandwidth tester
bwm-ng	V:1, I:10	95	small and simple console-based bandwidth monitor
ethstats	V:0, I:0	21	console-based Ethernet statistics monitor
ipfm	V:0, I:0	78	bandwidth analysis tool

Tabela 5.5: Instrumente de optimizare a rețelei

5.6.1 Finding optimal MTU

NM normally sets optimal [Maximum Transmission Unit \(MTU\)](#) automatically.

In some occasion, you may wish to set MTU manually after experiments with `ping(8)` with `-M do` option to send a ICMP packet with various data packet size. MTU is the maximum succeeding data packet size without IP fragmentation plus 28 bytes for the IPv4 and plus 48 bytes for the IPv6. For example the following finds MTU for IPv4 connection to be 1460 and MTU for IPv6 connection to be 1500.

```
$ ping -4 -c 1 -s $((1500-28)) -M do www.debian.org
PING (149.20.4.15) 1472(1500) bytes of data.
ping: local error: message too long, mtu=1460

--- ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

$ ping -4 -c 1 -s $((1460-28)) -M do www.debian.org
PING (130.89.148.77) 1432(1460) bytes of data.
1440 bytes from klecker-misc.debian.org (130.89.148.77): icmp_seq=1 ttl=50 time=325 ms

--- ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 325.318/325.318/325.318/0.000 ms
$ ping -6 -c 1 -s $((1500-48)) -M do www.debian.org
PING www.debian.org(mirror-csail.debian.org (2603:400a:ffff:bb8::801f:3e)) 1452 data bytes
1460 bytes from mirror-csail.debian.org (2603:400a:ffff:bb8::801f:3e): icmp_seq=1 ttl=47 ↔
time=191 ms

--- www.debian.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 191.332/191.332/191.332/0.000 ms
```

network environment	MTU	rationale
Dial-up link (IP: PPP)	576	standard
Ethernet link (IP: DHCP or fixed)	1500	standard and default

Tabela 5.6: Basic guide lines of the optimal MTU value

This process is [Path MTU \(PMTU\) discovery \(RFC1191\)](#) and the `tracepath(8)` command can automate this. In addition to these basic guide lines, you should know the following.

- Any use of tunneling methods ([VPN](#) etc.) may reduce optimal MTU further by their overheads.
- The MTU value should not exceed the experimentally determined PMTU value.
- The bigger MTU value is generally better when other limitations are met.

The [maximum segment size \(MSS\)](#) is used as an alternative measure of packet size. The relationship between MSS and MTU are the following.

- $MSS = MTU - 40$ pentru IPv4
- $MSS = MTU - 60$ pentru IPv6

Notă

The `iptables(8)` (see Secțiune [5.7](#)) based optimization can clamp packet size by the MSS and is useful for the router. See "TCP MSS" in `iptables(8)`.

5.6.2 WAN TCP optimization

The TCP throughput can be maximized by adjusting TCP buffer size parameters as in "[TCP tuning](#)" for the modern high-bandwidth and high-latency WAN. So far, the current Debian default settings serve well even for my LAN connected by the fast 1G bps FTTP service.

5.7 Infrastructura netfilter

[Netfilter](#) provides infrastructure for [stateful firewall](#) and [network address translation \(NAT\)](#) with [Linux kernel](#) modules (see Secțiune [3.9](#)).

Main user space program of [netfilter](#) is `iptables(8)`. You can manually configure [netfilter](#) interactively from shell, save its state with `iptables-save(8)`, and restore it via init script with `iptables-restore(8)` upon system reboot.

Configuration helper scripts such as [shorewall](#) ease this process.

See documentations at [Netfilter Documentation](#) (or in `/usr/share/doc/iptables/html/`).

- [Linux Networking-concepts HOWTO](#)
- [Linux 2.4 Packet Filtering HOWTO](#)
- [Linux 2.4 NAT HOWTO](#)

Indicație

Although these were written for Linux **2.4**, both `iptables(8)` command and netfilter kernel function apply for Linux **2.6** and **3.x** kernel series.

pachete	popcon(popularitate)	nr. de instalări	descriere
nftables	V:186, I:840	191	administration tools for packet filtering and NAT (Netfilter) (successor to {ip,ip6,arp,eb}tables)
iptables	V:344, I:636	2410	administration tools for netfilter (iptables(8) for IPv4, ip6tables(8) for IPv6)
arptables	V:0, I:1	102	administration tools for netfilter (arptables(8) for ARP)
ebtables	V:14, I:25	276	administration tools for netfilter (ebtables(8) for Ethernet bridging)
iptstate	V:0, I:1	122	continuously monitor netfilter state (similar to top(1))
ufw	V:69, I:96	859	Uncomplicated Firewall (UFW) is a program for managing a netfilter firewall
gufw	V:5, I:10	3663	graphical user interface for Uncomplicated Firewall (UFW)
firewalld	V:15, I:24	2482	firewalld is a dynamically managed firewall program with support for network zones
firewall-config	V:0, I:3	1076	graphical user interface for firewalld
shorewall-init	V:0, I:0	88	Shoreline Firewall initialization
shorewall	V:2, I:5	3090	Shoreline Firewall , netfilter configuration file generator
shorewall-lite	V:0, I:0	71	Shoreline Firewall , netfilter configuration file generator (light version)
shorewall6	V:0, I:1	1334	Shoreline Firewall , netfilter configuration file generator (IPv6 version)
shorewall6-lite	V:0, I:0	71	Shoreline Firewall , netfilter configuration file generator (IPv6, light version)

Tabela 5.7: Lista instrumentelor de paravan de protecție

Capitolul 6

Network applications

After establishing network connectivity (see Cap. 5), you can run various network applications.

Indicație

For modern Debian specific guide to the network infrastructure, read [The Debian Administrator's Handbook — Network Infrastructure](#).

Indicație

If you enabled "2-Step Verification" with some ISP, you need to obtain an application password to access POP and SMTP services from your program. You may need to approve your host IP in advance.

6.1 Navigatoare Web

There are many [web browser](#) packages to access remote contents with [Hypertext Transfer Protocol](#) (HTTP).

pachet	popcon	popularitate	tipul	descrierea navigatorului web
chromium	V:31, I:103	279987	X	Chromium , (open-source browser from Google)
firefox	V:16, I:21	274760	, ,	Firefox , (open-source browser from Mozilla, only available in Debian Unstable)
firefox-esr	V:202, I:437	265679	, ,	Firefox ESR , (Firefox Extended Support Release)
epiphany-browser	V:2, I:12	2258	, ,	GNOME , HIG compliant, Epiphany
konqueror	V:26, I:116	7931	, ,	KDE , Konqueror
dillo	V:0, I:4	1585	, ,	Dillo , (light weight browser, FLTK based)
w3m	V:11, I:151	2853	text	w3m
lynx	V:27, I:450	1972	, ,	Lynx
elinks	V:2, I:17	1755	, ,	ELinks
links	V:2, I:22	2321	, ,	Links (text only)
links2	V:0, I:11	5466	grafice	Links (console graphics without X)

Tabela 6.1: List of web browsers

6.1.1 Spoofing the User-Agent string

In order to access some overly restrictive web sites, you may need to spoof the [User-Agent](#) string returned by the web browser program. See:

- [MDN Web Docs: userAgent](#)
- [Chrome Developers: Override the user agent string](#)
- [How to change your user agent](#)
- [How to Change User-Agent in Chrome, Firefox, Safari, and more](#)
- [How to Change Your Browser's User Agent Without Installing Any Extensions](#)
- [How to change the User Agent in Gnome Web \(epiphany\)](#)

**Atenție**

Spoofed user-agent string may cause [bad side effects with Java](#).

6.1.2 Browser extension

All modern GUI browsers support source code based [browser extension](#) and it is becoming standardized as [web extensions](#).

6.2 Sistemul de poștă electronică

This section focuses on typical mobile workstations on consumer grade Internet connections.

**Atenție**

If you are to set up the mail server to exchange mail directly with the Internet, you should be better than reading this elementary document.

6.2.1 Email basics

An [email](#) message consists of three components, the message envelope, the message header, and the message body.

- The "To" and "From" information in the message envelope is used by the [SMTP](#) to deliver the email. (The "From" information in the message envelope is also called [bounce address](#), `From_`, etc.).
- The "To" and "From" information in the message header is displayed by the [email client](#). (While it is most common for these to be the same as ones in the message envelope, such is not always the case.)
- The email message format covering header and body data is extended by [Multipurpose Internet Mail Extensions \(MIME\)](#) from the plain ASCII text to other character encodings, as well as attachments of audio, video, images, and application programs.

Full featured GUI based [email clients](#) offer all the following functions using the GUI based intuitive configuration.

- It creates and interprets the message header and body data using [Multipurpose Internet Mail Extensions \(MIME\)](#) to deal the content data type and encoding.
- It authenticates itself to the ISP's SMTP and IMAP servers using the legacy [basic access authentication](#) or modern [OAuth 2.0](#). (For [OAuth 2.0](#), set it via Desktop environment settings. E.g., "Settings" -> "Online Accounts".)
- It sends the message to the ISP's smarthost SMTP server listening to the message submission port (587).
- It receives the stored message on the ISP's server from the TLS/IMAP4 port (993).
- It can filter mails by their attributes.
- It may offer additional functionalities: Contacts, Calendar, Tasks, Memos.

pachet	popcon	popularity	tipul
evolution	V:26, I:236	492	X GUI program (GNOME3, groupware suite)
thunderbird	V:46, I:111	273862	X GUI program (GTK, Mozilla Thunderbird)
kmail	V:42, I:106	25074	X GUI program (KDE)
mutt	V:12, I:100	7118	character terminal program probably used with vim
mew	V:0, I:0	2319	character terminal program under (x)emacs

Tabela 6.2: List of mail user agent (MUA)

6.2.2 Modern mail service limitation

Modern mail service are under some limitations in order to minimize exposure to the spam (unwanted and unsolicited email) problems.

- It is not realistic to run SMTP server on the consumer grade network to send mail directly to the remote host reliably.
- A mail may be rejected by any host en route to the destination quietly unless it appears as authentic as possible.
- It is not realistic to expect a single smarthost to send mails of unrelated source mail addresses to the remote host reliably.

This is because:

- The SMTP port (25) connections from hosts serviced by the consumer grade network to the Internet are blocked.
- The SMTP port (25) connections to hosts serviced by the consumer grade network from the Internet are blocked.
- The outgoing messages from hosts serviced by the consumer grade network to the Internet can only be sent via the message submission port (587).
- [Anti-spam techniques](#) such as [DomainKeys Identified Mail \(DKIM\)](#), [Sender Policy Framework \(SPF\)](#), and [Domain-based Message Authentication, Reporting and Conformance \(DMARC\)](#) are widely used for the [email filtering](#).
- The [DomainKeys Identified Mail](#) service may be provided for your mail sent through the smarthost.
- The smarthost may rewrite the source mail address in the message header to your mail account on the smarthost to prevent email address spoofing.

6.2.3 Historic mail service expectation

Some programs on Debian expect to access the `/usr/sbin/sendmail` command to send emails as their default or customized setting since the mail service on a UNIX system functioned historically as:

- An email is created as a text file.
- The email is handed to the `/usr/sbin/sendmail` command.
- For the destination address on the same host, the `/usr/sbin/sendmail` command makes local delivery of the email by appending it to the `/var/mail/$username` file.
 - Commands expecting this feature: `apt - listchanges`, `cron`, `at`, ...
- For the destination address on the remote host, the `/usr/sbin/sendmail` command makes remote transfer of the email to the destination host found by the DNS MX record using SMTP.
 - Commands expecting this feature: `popcon`, `reportbug`, `bts`, ...

6.2.4 Agentul de transport al poștei electronice („Mail transport agent”: MTA)

Debian mobile workstations can be configured just with full featured GUI based [email clients](#) without [mail transfer agent \(MTA\)](#) program after Debian 12 Bookworm.

Debian traditionally installed some MTA program to support programs expecting the `/usr/sbin/sendmail` command. Such MTA on mobile workstations must cope with Secțiune 6.2.2 and Secțiune 6.2.3.

For mobile workstations, the typical choice of MTA is either `exim4-daemon-light` or `postfix` with its installation option such as "Mail sent by smarthost; received via SMTP or fetchmail" selected. These are light weight MTAs that respect `/etc/aliases`.

Indicație

Configuring `exim4` to send the Internet mail via multiple corresponding smarthosts for multiple source email addresses is non-trivial. If you need such capability for some programs, set them up to use `msmtp` which is easy to set up for multiple source email addresses. Then leave main MTA only for a single email address.

6.2.4.1 The configuration of `exim4`

For the Internet mail via smarthost, you (re)configure `exim4-*` packages as the following.

```
$ sudo systemctl stop exim4
$ sudo dpkg-reconfigure exim4-config
```

Select "mail sent by smarthost; received via SMTP or fetchmail" for "General type of mail configuration".

Set "System mail name:" to its default as the FQDN (see Secțiune 5.1.1).

Set "IP-addresses to listen on for incoming SMTP connections:" to its default as "127.0.0.1 ; ::1".

Unset contents of "Other destinations for which mail is accepted:".

Unset contents of "Machines to relay mail for:".

Set "IP address or host name of the outgoing smarthost:" to "smtp.hostname.dom:587".

Select "No" for "Hide local mail name in outgoing mail?". (Use `/etc/email-addresses` as in Secțiune 6.2.4.3, instead.)

Reply to "Keep number of DNS-queries minimal (Dial-on-Demand)?" as one of the following.

pachet	popcon(popularity)	download rate	descriere
exim4-daemon-light	V:224, I:231	1649	Exim4 mail transport agent (MTA: Debian default)
exim4-daemon-heavy	V:5, I:5	1814	Exim4 mail transport agent (MTA: flexible alternative)
exim4-base	V:229, I:237	1646	Exim4 documentation (text) and common files
exim4-doc-html	I:1	3798	Exim4 documentation (html)
exim4-doc-info	I:0	648	Exim4 documentation (info)
postfix	V:110, I:116	4002	Postfix mail transport agent (MTA: secure alternative)
postfix-doc	I:5	4840	Postfix documentation (html+text)
saslm2-bin	V:5, I:11	368	Cyrus SASL API implementation (supplement postfix for SMTP AUTH)
cyrus-saslm2-doc	I:0	2142	Cyrus SASL - documentație
msmtp	V:7, I:13	811	Light weight MTA
msmtp-mta	V:5, I:7	136	Light weight MTA (sendmail compatibility extension to msmtp)
esmtm	V:0, I:0	123	Light weight MTA
esmtm-run	V:0, I:0	27	Light weight MTA (sendmail compatibility extension to esmtm)
nullmailer	V:7, I:8	483	Strip down MTA, no local mail
ssmtm	V:4, I:6	133	Strip down MTA, no local mail
sendmail-bin	V:11, I:11	1962	Full featured MTA (only if you are already familiar)
courier-mta	V:0, I:0	2674	Full featured MTA (web interface etc.)
git-email	V:0, I:11	1187	git-send-email(1) program for sending series of patch emails

Tabela 6.3: List of basic mail transport agent related packages

- "No" if the system is connected to the Internet while booting.
- "Yes" if the system is **not** connected to the Internet while booting.

Set "Delivery method for local mail:" to "mbox format in /var/mail/".

Select "Yes" for "Split configuration into small files?".

Create password entries for the smarthost by editing "/etc/exim4/passwd.client".

```
$ sudo vim /etc/exim4/passwd.client
...
$ cat /etc/exim4/passwd.client
^smtp.*\.hostname\.dom:username@hostname.dom:password
```

Configure exim4(8) with "QUEUERUNNER='queueonly'", "QUEUERUNNER='nodaemon'", etc. in "/etc/default/exim4" to minimize system resource usages. (optional)

Start exim4 by the following.

```
$ sudo systemctl start exim4
```

The host name in "/etc/exim4/passwd.client" should not be the alias. You check the real host name with the following.

```
$ host smtp.hostname.dom
smtp.hostname.dom is an alias for smtp99.hostname.dom.
smtp99.hostname.dom has address 123.234.123.89
```

I use regex in "/etc/exim4/passwd.client" to work around the alias issue. SMTP AUTH probably works even if the ISP moves host pointed by the alias.

You can manually update exim4 configuration by the following:

- Update `exim4` configuration files in `"/etc/exim4/"`.
 - creating `"/etc/exim4/exim4.conf.localmacros"` to set MACROs and editing `"/etc/exim4/exim4.conf.template"` (non-split configuration)
 - creating new files or editing existing files in the `"/etc/exim4/exim4.conf.d"` subdirectories. (split configuration)
- Run `"systemctl reload exim4"`.

**Atenție**

Starting `exim4` takes long time if "No" (default value) was chosen for the `debconf` query of "Keep number of DNS-queries minimal (Dial-on-Demand)?" and the system is **not** connected to the Internet while booting.

Please read the official guide at: `"/usr/share/doc/exim4-base/README.Debian.gz"` and `update-exim4.conf(8)`.

**Avertisment**

For all practical consideration, use [SMTP](#) with [STARTTLS](#) on port 587 or [SMTPS](#) (SMTP over SSL) on port 465, instead of plain SMTP on port 25.

6.2.4.2 The configuration of postfix with SASL

For the Internet mail via smarthost, you should first read [postfix documentation](#) and key manual pages.

comanda	funcție
<code>postfix(1)</code>	Postfix control program
<code>postconf(1)</code>	Postfix configuration utility
<code>postconf(5)</code>	Postfix configuration parameters
<code>postmap(1)</code>	Postfix lookup table maintenance
<code>postalias(1)</code>	Postfix alias database maintenance

Tabela 6.4: List of important postfix manual pages

You (re)configure `postfix` and `sasl2-bin` packages as follows.

```
$ sudo systemctl stop postfix
$ sudo dpkg-reconfigure postfix
```

Chose "Internet with smarthost".

Set "SMTP relay host (blank for none):" to `"[smtp.hostname.dom]:587"` and configure it by the following.

```
$ sudo postconf -e 'smtp_sender_dependent_authentication = yes'
$ sudo postconf -e 'smtp_sasl_auth_enable = yes'
$ sudo postconf -e 'smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd'
$ sudo postconf -e 'smtp_sasl_type = cyrus'
$ sudo vim /etc/postfix/sasl_passwd
```

Create password entries for the smarthost.

```
$ cat /etc/postfix/sasl_passwd
[smtp.hostname.dom]:587      username:password
$ sudo postmap hash:/etc/postfix/sasl_passwd
```

Start the `postfix` by the following.

```
$ sudo systemctl start postfix
```

Here the use of "[" and "]" in the `dpkg-reconfigure` dialog and `"/etc/postfix/sasl_passwd"` ensures not to check MX record but directly use exact hostname specified. See "Enabling SASL authentication in the Postfix SMTP client" in `"/usr/share/doc/postfix/html/SASL_README.html"`.

6.2.4.3 The mail address configuration

There are a few [mail address configuration files for mail transport, delivery and user agents](#).

fișier	funcție	aplicație
<code>/etc/mailname</code>	default host name for (outgoing) mail	Debian specific, <code>mailname(5)</code>
<code>/etc/email-addresses</code>	host name spoofing for outgoing mail	<code>exim(8)</code> specific, <code>exim4-config_files(5)</code>
<code>/etc/postfix/generic</code>	host name spoofing for outgoing mail	<code>postfix(1)</code> specific, activated after <code>postmap(1)</code> command execution.
<code>/etc/aliases</code>	account name alias for incoming mail	general, activated after <code>newaliases(1)</code> command execution.

Tabela 6.5: List of mail address related configuration files

The **mailname** in the `"/etc/mailname"` file is usually a fully qualified domain name (FQDN) that resolves to one of the host's IP addresses. For the mobile workstation which does not have a hostname with resolvable IP address, set this **mailname** to the value of `"hostname -f"`. (This is safe choice and works for both `exim4-*` and `postfix`.)

Indicație

The contents of `"/etc/mailname"` is used by many non-MTA programs for their default behavior. For `mutt`, set `"hostname"` and `"from"` variables in `~/muttrc` file to override the **mailname** value. For programs in the `devscripts` package, such as `bts(1)` and `dch(1)`, export environment variables `"$DEBFULLNAME"` and `"$DEBEMAIL"` to override it.

Indicație

The `popularity-contest` package normally send mail from root account with FQDN. You need to set `MAILFROM` in `/etc/popularity-contest.conf` as described in the `/usr/share/popularity-contest/default.conf` file. Otherwise, your mail will be rejected by the `smarthost` SMTP server. Although this is tedious, this approach is safer than rewriting the source address for all mails from root by MTA and should be used for other daemons and cron scripts.

When setting the **mailname** to `"hostname -f"`, the spoofing of the source mail address via MTA can be realized by the following.

- `"/etc/email-addresses"` file for `exim4(8)` as explained in the `exim4-config_files(5)`
- `"/etc/postfix/generic"` file for `postfix(1)` as explained in the `generic(5)`

For `postfix`, the following extra steps are needed.

```
# postmap hash:/etc/postfix/generic
# postconf -e 'smtp_generic_maps = hash:/etc/postfix/generic'
# postfix reload
```

You can test mail address configuration using the following.

- `exim(8)` cu opțiunile `-brw`, `-bf`, `-bF`, `-bV`, ...
- `postmap(1)` cu opțiunea `-q`.

Indicație

Exim comes with several utility programs such as `exiqgrep(8)` and `exipick(8)`. See "`dpkg -L exim4-base | grep man8/`" for available commands.

6.2.4.4 Basic MTA operations

There are several basic MTA operations. Some may be performed via `sendmail(1)` compatibility interface.

comanda «exim»	comanda «postfix»	descriere
<code>sendmail</code>	<code>sendmail</code>	read mails from standard input and arrange for delivery (<code>-bm</code>)
<code>mailq</code>	<code>mailq</code>	list the mail queue with status and queue ID (<code>-bp</code>)
<code>newaliases</code>	<code>newaliases</code>	initialize alias database (<code>-I</code>)
<code>exim4 -q</code>	<code>postqueue -f</code>	flush waiting mails (<code>-q</code>)
<code>exim4 -qf</code>	<code>postsuper -r ALL deferred; postqueue -f</code>	flush all mails
<code>exim4 -qff</code>	<code>postsuper -r ALL; postqueue -f</code>	flush even frozen mails
<code>exim4 -Mg queue_id</code>	<code>postsuper -h queue_id</code>	freeze one message by its queue ID
<code>exim4 -Mrm queue_id</code>	<code>postsuper -d queue_id</code>	remove one message by its queue ID
<code>N/D</code>	<code>postsuper -d ALL</code>	elimină toate mesaje

Tabela 6.6: List of basic MTA operation

Indicație

It may be a good idea to flush all mails by a script in `"/etc/ppp/ip-up.d/*"`.

6.3 The remote access server and utilities (SSH)

The [Secure SHell](#) (SSH) is the **secure** way to connect over the Internet. A free version of SSH called [OpenSSH](#) is available as `openssh-client` and `openssh-server` packages in Debian.

For the user, `ssh(1)` functions as a smarter and more secure `telnet(1)`. Unlike `telnet` command, `ssh` command does not stop on the `telnet` escape character (initial default CTRL-`]`).

Although `shellinabox` is not a SSH program, it is listed here as an interesting alternative for the remote terminal access.

See also Secțiune [7.9](#) for connecting to remote X client programs.



Atenție

See Secțiune [4.6.3](#) if your SSH is accessible from the Internet.

pachet	popcon(popularity)	limite	instrument	descriere
openssh-client	V:902, I:996	5131	ssh(1)	Secure shell client
openssh-server	V:752, I:808	3501	sshd(8)	Secure shell server
ssh-askpass	I:18	103	ssh-askpass(1)	asks user for a pass phrase for ssh-add (plain X)
ssh-askpass-gnome	V:0, I:3	215	ssh-askpass-gnome(1)	asks user for a pass phrase for ssh-add (GNOME)
ssh-askpass-fullscreen	V:0, I:0	41	ssh-askpass-fullscreen(1)	asks user for a pass phrase for ssh-add (GNOME) with extra eye candy
shellinabox	V:0, I:1	525	shellinabox(1)	web server for browser accessible VT100 terminal emulator

Tabela 6.7: List of remote access server and utilities

Indicație

Please use the screen(1) program to enable remote shell process to survive the interrupted connection (see Secțiune 9.1.2).

6.3.1 Basics of SSH

The OpenSSH SSH daemon supports SSH protocol 2 only.

Please read "/usr/share/doc/openssh-client/README.Debian.gz", ssh(1), sshd(8), ssh-keygen(1), ssh-add(1) and ssh-agent(1).

**Avertisment**

"/etc/ssh/sshd_not_to_be_run" must not be present if one wishes to run the OpenSSH server. Don't enable rhost based authentication (HostbasedAuthentication in /etc/ssh/sshd_config).

fișier de configurare	descrierea fișierului de configurare
/etc/ssh/ssh_config	SSH client defaults, see ssh_config(5)
/etc/ssh/sshd_config	SSH server defaults, see sshd_config(5)
~/.ssh/authorized_keys	default public SSH keys that clients use to connect to this account on this SSH server
~/.ssh/id_rsa	secret SSH-2 RSA key of the user
~/.ssh/id_key-type-name	secret SSH-2 <i>key-type-name</i> key such as ecdsa, ed25519, ... of the user

Tabela 6.8: List of SSH configuration files

The following starts an ssh(1) connection from a client.

6.3.2 User name on the remote host

If you use the same user name on the local and the remote host, you can eliminate typing "username@".

Even if you use different user name on the local and the remote host, you can eliminate it using "~/.ssh/config". For [Debian Salsa service](#) with account name "foo-guest", you set "~/.ssh/config" to contain the following.

comanda	descriere
ssh nume-utilizator@nume-gazdă.domeniu.ext	connect with default mode
ssh -v nume-utilizator@nume-gazdă.domeniu.ext	connect with default mode with debugging messages
ssh -o PreferredAuthentications=parolă nume-utilizator@nume-gazdă.domeniu.ext	force to use password with SSH version 2
ssh -t nume-utilizator@nume-gazdă.domeniu.ext parolă	run a passwd program to update password on a remote host

Tabela 6.9: List of SSH client startup examples

```
Host salsa.debian.org people.debian.org
User foo-guest
```

6.3.3 Connecting without remote passwords

One can avoid having to remember passwords for remote systems by using "PubkeyAuthentication" (SSH-2 protocol).

On the remote system, set the respective entries, "PubkeyAuthentication yes", in "/etc/ssh/sshd_config". Generate authentication keys locally and install the public key on the remote system by the following.

```
$ ssh-keygen -t rsa
$ cat .ssh/id_rsa.pub | ssh user1@remote "cat - >>.ssh/authorized_keys"
```

You can add options to the entries in "~/.ssh/authorized_keys" to limit hosts and to run specific commands. See sshd(8) "AUTHORIZED_KEYS FILE FORMAT".

6.3.4 Dealing with alien SSH clients

There are some free [SSH](#) clients available for other platforms.

mediu	free SSH program
Windows	puTTY (PuTTY: a free SSH and Telnet client) (GPL)
Windows (cygwin)	SSH in cygwin (Cygwin: Get that Linux feeling - on Windows) (GPL)
Mac OS X	OpenSSH; use ssh in the Terminal application (GPL)

Tabela 6.10: List of free SSH clients for other platforms

6.3.5 Setting up ssh-agent

It is safer to protect your SSH authentication secret keys with a pass phrase. If a pass phrase was not set, use "ssh-keygen -p" to set it.

Place your public SSH key (e.g. "~/.ssh/id_rsa.pub") into "~/.ssh/authorized_keys" on a remote host using a password-based connection to the remote host as described above.

```
$ ssh-agent bash
$ ssh-add ~/.ssh/id_rsa
Enter passphrase for /home/username/.ssh/id_rsa:
Identity added: /home/username/.ssh/id_rsa (/home/username/.ssh/id_rsa)
```

No remote password needed from here on for the next command.

```
$ scp foo username@remote.host:foo
```

Press ^D to terminating ssh-agent session.

For the X server, the normal Debian startup script executes ssh-agent as the parent process. So you only need to execute ssh-add once. For more, read ssh-agent(1) and ssh-add(1).

6.3.6 Sending a mail from a remote host

If you have an SSH shell account on a server with proper DNS settings, you can send a mail generated on your workstation as an email genuinely sent from the remote server.

```
$ ssh username@example.org /usr/sbin/sendmail -bm -ti -f "username@example.org" < mail_data ←
.txt
```

6.3.7 Port forwarding for SMTP/POP3 tunneling

To establish a pipe to connect to port 25 of remote-server from port 4025 of localhost, and to port 110 of remote-server from port 4110 of localhost through ssh, execute on the local host as the following.

```
# ssh -q -L 4025:remote-server:25 4110:remote-server:110 username@remote-server
```

This is a secure way to make connections to SMTP/POP3 servers over the Internet. Set the "AllowTcpForwarding" entry to "yes" in "/etc/ssh/sshd_config" of the remote host.

6.3.8 How to shutdown the remote system on SSH

You need to protect the process doing "shutdown -h now" (see Secțiune 1.1.8) from the termination of SSH using the at(1) command (see Secțiune 9.4.13) by the following.

```
# echo "shutdown -h now" | at now
```

Running "shutdown -h now" in screen(1) (see Secțiune 9.1.2) session is another way to do the same.

6.3.9 Soluționarea problemelor SSH

If you have problems, check the permissions of configuration files and run ssh with the "-v" option.

Use the "-p" option if you are root and have trouble with a firewall; this avoids the use of server ports 1 — 1023.

If ssh connections to a remote site suddenly stop working, it may be the result of tinkering by the sysadmin, most likely a change in "host_key" during system maintenance. After making sure this is the case and nobody is trying to fake the remote host by some clever hack, one can regain a connection by removing the "host_key" entry from "~/.ssh/known_hosts" on the local host.

6.4 The print server and utilities

In the old Unix-like system, the BSD [Line printer daemon \(lpr\)](#) was the standard and the standard print out format of the classic free software was [PostScript \(PS\)](#). Some filter system was used along with [Ghostscript](#) to enable printing to the non-PostScript printer. See Secțiune [11.4.1](#).

In the modern Debian system, the [Common UNIX Printing System \(CUPS\)](#) is the de facto standard and the standard print out format of the modern free software is [Portable Document Format \(PDF\)](#).

The CUPS uses [Internet Printing Protocol \(IPP\)](#). The IPP is now supported by other OSs such as Windows XP and Mac OS X and has become new cross-platform de facto standard for remote printing with bi-directional communication capability.

Thanks to the file format dependent auto-conversion feature of the CUPS system, simply feeding any data to the `lpr` command should generate the expected print output. (In CUPS, `lpr` can be enabled by installing the `cups-bsd` package.)

The Debian system has some notable packages for the print servers and utilities.

pachet	popcon	popul	limitat	ipocet	descriere
lpr	V:2, I:2	364	printer (515)		BSD <code>lpr/lpd</code> (Line printer daemon)
lprng	V:0, I:0	3078	, ,		, , (Enhanced)
cups	V:94, I:456	1092	IPP (631)		Internet Printing CUPS server
cups-client	V:114, I:470	433	, ,		System V printer commands for CUPS: <code>lp(1)</code> , <code>lpstat(1)</code> , <code>lpoptions(1)</code> , <code>cancel(1)</code> , <code>lpmove(8)</code> , <code>lpinfo(8)</code> , <code>lpadmin(8)</code> , ...
cups-bsd	V:33, I:197	131	, ,		BSD printer commands for CUPS: <code>lpr(1)</code> , <code>lpq(1)</code> , <code>lprm(1)</code> , <code>lpc(8)</code>
printer-driver-gutenprint	V:14, I:65	1194	Not applicable		printer drivers for CUPS

Tabela 6.11: List of print servers and utilities

Indicație

You can configure CUPS system by pointing your web browser to "<http://localhost:631/>".

6.5 Other network application servers

Here are other network application servers.

Common Internet File System Protocol (CIFS) is the same protocol as [Server Message Block \(SMB\)](#) and is used widely by Microsoft Windows.

Indicație

See Secțiune [4.5.2](#) for integration of server systems.

Indicație

The hostname resolution is usually provided by the [DNS](#) server. For the host IP address dynamically assigned by [DHCP](#), [Dynamic DNS](#) can be set up for the hostname resolution using `bind9` and `isc-dhcp-server` as described in the [DDNS page on the Debian wiki](#).

pachet	popcon(popularity)	limita de conexiuni	protocol	descriere
telnetd	V:0, I:1	51	TELNET	TELNET server
telnetd-ssl	V:0, I:0	158	, ,	, , (SSL support)
nfs-kernel-server	V:46, I:56	798	NFS	Unix file sharing
samba	V:107, I:123	5019	SMB	Windows file and printer sharing
netatalk	V:0, I:1	814	ATP	Apple/Mac file and printer sharing (AppleTalk)
proftpd-basic	V:4, I:10	452	FTP	General file download
apache2	V:192, I:231	587	HTTP	General web server
squid	V:9, I:10	9349	, ,	General web proxy server
bind9	V:36, I:39	887	DNS	IP address for other hosts
isc-dhcp-server	V:15, I:26	6102	DHCP	IP address of client itself

Tabela 6.12: List of other network application servers

Indicație

Use of proxy server such as squid is much more efficient for saving bandwidth than use of local mirror server with the full Debian archive contents.

6.6 Other network application clients

Here are other network application clients.

6.7 The diagnosis of the system daemons

The `telnet` program enables manual connection to the system daemons and its diagnosis.

For testing plain [POP3](#) service, try the following

```
$ telnet mail.ispname.net pop3
```

For testing the [TLS/SSL](#) enabled [POP3](#) service by some ISPs, you need TLS/SSL enabled `telnet` client by the `telnet-ssl` or `openssl` packages.

```
$ telnet -z ssl pop.gmail.com 995
```

```
$ openssl s_client -connect pop.gmail.com:995
```

The following [RFCs](#) provide required knowledge to each system daemon.

The port usage is described in `"/etc/services"`.

pachet	popcon(popularit)	dimensiune	protocol	descriere
netcat-traditional	V:43, I:903	139	TCP/IP	TCP/IP swiss army knife
netcat-openbsd	V:20, I:122	109	TCP/IP	TCP/IP swiss army knife with support for IPv6, proxies, and Unix sockets
openssl	V:843, I:996	2493	SSL	Secure Socket Layer (SSL) binary and related cryptographic tools
stunnel4	V:6, I:10	569	, ,	universal SSL Wrapper
telnet	V:13, I:256	51	TELNET	Client TELNET
telnet-ssl	V:0, I:2	199	, ,	, , (SSL support)
nfs-common	V:145, I:206	1194	NFS	Unix file sharing
smbclient	V:26, I:211	2106	SMB	MS Windows file and printer sharing client
cifs-utils	V:30, I:120	351	, ,	mount and umount commands for remote MS Windows file
ftp	V:3, I:69	53	FTP	Client FTP
lftp	V:3, I:24	2377	, ,	, ,
ncftp	V:1, I:10	1407	, ,	full screen FTP client
wget	V:185, I:982	3784	HTTP and FTP	web downloader
curl	V:212, I:683	504	, ,	, ,
axel	V:0, I:3	228	, ,	accelerated downloader
aria2	V:4, I:22	1984	, ,	accelerated downloader with BitTorrent and Metalink supports
bind9-host	V:121, I:941	140	DNS	host(1) from bind9, "Priority: standard"
dnsutils	I:186	22	, ,	dig(1) from bind, "Priority: standard"
isc-dhcp-client	V:179, I:749	2884	DHCP	obtain IP address
ldap-utils	V:10, I:58	788	LDAP	obtain data from LDAP server

Tabela 6.13: List of network application clients

RFC	descriere
rfc1939 și rfc2449	POP3 service
rfc3501	IMAP4 service
rfc2821 (rfc821)	SMTP service
rfc2822 (rfc822)	Mail file format
rfc2045	Multipurpose Internet Mail Extensions (MIME)
rfc819	DNS service
rfc2616	HTTP service
rfc2396	URI definition

Tabela 6.14: List of popular RFCs

Capitolul 7

GUI System

7.1 GUI desktop environment

There are several choices for the full featured [GUI](#) desktop environment on the Debian system.

task package	popcon(popularity)	driftage	descriere
task-gnome-desktop	1:196	9	GNOME desktop environment
task-xfce-desktop	1:90	9	Xfce desktop environment
task-kde-desktop	1:95	6	KDE Plasma desktop environment
task-mate-desktop	1:36	9	MATE desktop environment
task-cinnamon-desktop	1:39	9	Cinnamon desktop environment
task-lxde-desktop	1:23	9	LXDE desktop environment
task-lxqt-desktop	1:17	9	LXQt desktop environment
task-gnome-flashback-desktop	1:11	6	GNOME Flashback desktop environment

Tabela 7.1: List of desktop environment

Indicație

Dependency packages selected by a task metapackage may be out of sync with the latest package transition state under the Debian unstable/testing environment. For task-gnome-desktop, you may need to adjust package selections as follows:

- Start aptitude(8) as `sudo aptitude -u`.
 - Move cursor to "Tasks" and press "Enter".
 - Move cursor to "End-user" press "Enter".
 - Move cursor to "GNOME" press "Enter".
 - Move cursor to task-gnome-desktop and press "Enter".
 - Move cursor to "Depends" and press "m" (manually selected).
 - Move cursor to "Recommends" and press "m" (manually selected).
 - Move cursor to "task-gnome-desktop and press "-". (drop)
 - Adjust selected packages while dropping problematic ones causing package conflicts.
 - Press "g" to start install.
-

This chapter will focus mostly on the default desktop environment of Debian: task-gnome-desktop offering [GNOME](#) on [wayland](#).

7.2 GUI communication protocol

GUI communication protocol used on the GNOME desktop can be:

- [Wayland \(display server protocol\)](#) (native)
- [X Window System core protocol](#) (via xwayland)

Please check freedesktop.org site for how Wayland architecture is different from X Window architecture.

From user's perspective, differences can be colloquially summarized as:

- Wayland is a same-host GUI communication protocol: new, simpler, faster, no setuid root binary
- X Window is a network-capable GUI communication protocol: traditional, complex, slower, setuid root binary

For applications using Wayland protocol, the access to their display contents from a remote host is supported by the [VNC](#) or [RDP](#). See Secțiune 7.8

Modern X servers have the [MIT Shared Memory Extension](#) and communicate with their local X clients using the local shared memory. This bypasses the network transparent [Xlib](#) interprocess communication channel and gains performance. This situation was the [background](#) of creating Wayland as a local-only GUI communication protocol.

Using the xeyes program started from the GNOME terminal, you can check GUI communication protocol used by each GUI application.

```
$ xeyes
```

- If the mouse cursor is on an application such as "GNOME terminal" which uses Wayland display server protocol, eyes don't move with the mouse cursor.
-

- If the mouse cursor is on an application such as "xterm" which uses X Window System core protocol, eyes move with the mouse cursor exposing not-so-isolated nature of X Window architecture.

As of April 2021, many popular GUI applications such as GNOME and [LibreOffice \(LO\)](#) applications have been migrated to the Wayland display server protocol. I see xterm, gitk, chromium, firefox, gimp, dia, and KDE applications still use X Window System core protocol.

Notă

For both the xwayland on Wayland or the native X Window System, the old X server configuration file `/etc/X11/xorg.conf` shouldn't exist on the system. The graphics and input devices are now configured by the kernel with [DRM](#), [KMS](#), and [udev](#). The native X server has been rewritten to use them. See "[modedb default video mode support](#)" in the Linux kernel documentation.

7.3 GUI infrastructure

Here are notable GUI infrastructure packages for the GNOME on Wayland environment.

pachet	popcon(popularitate)	dimensiunea pachetului	descriere
mutter	V:0, I:30	221	GNOME's mutter window manager [auto]
xwayland	V:250, I:343	2541	An X server running on top of wayland [auto]
gnome-remote-desktop	V:110, I:246	2215	Remote desktop daemon for GNOME using PipeWire [auto]
gnome-tweaks	V:19, I:238	1145	Advanced configuration settings for GNOME
gnome-shell-extension-prefs	V:8, I:153	82	Tool to enable / disable GNOME Shell extensions

Tabela 7.2: List of notable GUI infrastructure packages

Here, "**[auto]**" means that these packages are automatically installed when `task-gnome-desktop` is installed.

Indicație

`gnome-tweaks` is the indispensable configuration utility. For example:

- You can force "Over-Amplification" of sound volume from "General".
 - You can force "Caps" to become "Esc" from "Keyboard & Mouse" -> "Keyboard" -> "Additional Layout Option".
-

Indicație

Detail features of GNOME desktop environment can be configured with utilities started by typing "settings", "tweaks", or "extensions" after pressing Super-key.

7.4 Aplicații cu interfață grafică

Many useful GUI applications are available on Debian now. Installing software packages such as `scribus` (KDE) on GNOME desktop environment are quite acceptable since corresponding functionality is not available under GNOME desktop environment. But installing too many packages with duplicated functionalities may clutter your system.

Here is a list of GUI applications which caught my eyes.

7.5 User directories

Default names for user directories such as `~/Desktop`, `~/Documents`, ..., used by the Desktop environment depend on the locale used for the system installation. You can reset them to the English ones by:

```
$ LANGUAGE=C xdg-user-dirs-update --force
```

Then you manually move all the data to the newer directories. See `xdg-user-dirs-update(1)`.

You can also set them to any names by editing `~/ .config/user-dirs.dirs`. See `user-dirs.dirs(5)`.

7.6 Tipuri de litere

Many useful scalable fonts are available for users on Debian. User's concern is how to avoid redundancy and how to configure parts of installed fonts to be disabled. Otherwise, useless font choices may clutter your GUI application menus.

Debian system uses [FreeType](#) 2.0 library to rasterise many scalable font formats for screen and print:

- [Type 1 \(PostScript\) fonts](#) which use cubic [Bézier curves](#) (almost obsolete format)
- [TrueType fonts](#) which use quadratic [Bézier curves](#) (good choice format)
- [OpenType fonts](#) which use cubic [Bézier curves](#) (best choice format)

7.6.1 Basic fonts

The following table is compiled in the hope to help users to chose appropriate scalable fonts with clear understanding of the metric compatibility and the glyph coverage. Most fonts cover all Latin, Greek, and Cyril characters. The final choice of activated fonts can also be affected by your aesthetics. These fonts can be used for the screen display or for the paper printing.

Here:

- "MCM" stands for "metric compatible with fonts provided by Microsoft"
- "MCMATC" stands for "metric compatible with fonts provided by Microsoft: [Arial](#), [Times New Roman](#), [Courier New](#)"
- "MCAHTC" stands for "metric compatible with fonts provided by [Adobe](#): Helvetica, Times, Courier"
- Numbers in font type columns stands for the rough relative "M" width for the same point size font.
- "P" in mono font type columns stands for its usability for programming having clearly distinguishable "0"/"O" and "1"/"l"/"I".
- The `ttf-mscorefonts-installer` package downloads Microsoft's "[Core fonts for the Web](#)" and installs [Arial](#), [Times New Roman](#), [Courier New](#), [Verdana](#), These installed font data are non-free data.

Many free Latin fonts have their lineage traced to [URW Nimbus](#) family or [Bitstream Vera](#).

Indicație

If your locale needs fonts not covered well by the above fonts, please use aptitude to check under task packages listed under "Tasks" -> "Localization". The font packages listed as "Depends:" or "Recommends:" in the localization task packages are the primary candidates.

pachet	popcon(popularitate)	dimensiunea pachetului	tipul	descriere
evolution	V:26, I:236	492	GNOME	Personal information Management (groupware and email)
thunderbird	V:46, I:111	273862	GTK	Email client (Mozilla Thunderbird)
kontakt	V:1, I:11	2258	KDE	Personal information Management (groupware and email)
libreoffice-writer	V:117, I:439	33053	LO	procesor de text
abiword	V:0, I:5	3576	GNOME	procesor de text
calligrawords	V:0, I:5	6937	KDE	procesor de text
scribus	V:1, I:14	32052	KDE	desktop publishing editor to edit PDF files
glabels	V:0, I:2	1283	GNOME	editor de etichete
libreoffice-calc	V:111, I:435	28164	LO	foaie de calcul
gnumeric	V:3, I:11	9958	GNOME	foaie de calcul
calligrasheets	V:0, I:4	13593	KDE	foaie de calcul
libreoffice-impress	V:92, I:434	2469	LO	prezentare
calligrastage	V:0, I:4	6017	KDE	prezentare
libreoffice-base	V:24, I:82	5031	LO	database management
kexi	V:0, I:0	7565	KDE	database management
libreoffice-draw	V:92, I:434	10985	LO	vector graphics editor (draw)
inkscape	V:12, I:86	113183	GNOME	vector graphics editor (draw)
karbon	V:0, I:5	3962	KDE	vector graphics editor (draw)
dia	V:1, I:19	4086	GTK	flowchart and diagram editor
gimp	V:45, I:232	32135	GTK	bitmap graphics editor (paint)
shotwell	V:15, I:255	6334	GTK	digital photo organizer
digikam	V:1, I:9	302	KDE	digital photo organizer
darktable	V:4, I:12	35873	GTK	lighttable and darkroom for photographers
planner	V:0, I:5	1400	GNOME	project management
calligraplan	V:0, I:3	19241	KDE	project management
gnucash	V:2, I:7	29748	GNOME	personal accounting
homebank	V:0, I:1	3196	GTK	personal accounting
lilypond	V:0, I:6	16924	-	music typesetter
kmymoney	V:0, I:2	18826	KDE	personal accounting
librecad	V:1, I:15	9100	Qt-app	computer-aided design (CAD) system (2D)
freecad	I:20	110	Qt-app	computer-aided design (CAD) system (3D)
kicad	V:3, I:15	163802	GTK	electronic schematic and PCB design software
xsane	V:10, I:135	1512	GTK	interfață scanner
libreoffice-math	V:83, I:437	1904	LO	mathematical equation/formula editor
calibre	V:7, I:27	65174	KDE	e-book converter and library management
fbreader	V:0, I:7	3783	GTK	cititor cărți electronice
evince	V:79, I:300	963	GNOME	document(pdf) viewer
okular	V:41, I:135	4415	KDE	document(pdf) viewer
x11-apps	V:31, I:467	2461	pure X-app	xeyes(1), etc.
x11-utils	V:217, I:565	651	pure X-app	xev(1), xwininfo(1), etc.

Tabela 7.3: List of notable GUI applications

pachet	popcon(popularity)	con(continues)	names	serif	mono	note on font
fonts-cantarell	V:180, I:302	213	59	-	-	Cantarell (GNOME 3, display)
fonts-noto	I:158	31	61	63	40	Noto fonts (Google, multi-lingual with CJK)
fonts-dejavu	I:405	35	58	68	40	DejaVu (GNOME 2, MCM: Verdana , extended Bitstream Vera)
fonts-liberation2	V:71, I:239	15	56	60	40	Liberation fonts for LibreOffice (Red Hat, MCMATC)
fonts-croscore	V:21, I:39	5274	56	60	40	Chrome OS: Arimo, Tinos and Cousine (Google, MCMATC)
fonts-crosextra-carlito	V:18, I:102	2696	57	-	-	Chrome OS: Carlito (Google, MCM: Calibri)
fonts-crosextra-caladea	V:10, I:97	347	-	55	-	Chrome OS: Caladea (Google, MCM: Cambria) (Latin only)
fonts-freefont-ttf	V:78, I:209	14460	57	59	40	GNU FreeFont (extended URW Nimbus)
fonts-quicksand	V:207, I:460	392	56	-	-	Debian task-desktop, Quicksand (display, Latin only)
fonts-hack	V:32, I:140	2507	-	-	40 P	A typeface designed for source code Hack (Facebook)
fonts-sil-gentiumplus	I:30	14345	-	54	-	Gentium SIL
fonts-sil-charis	I:29	6704	-	59	-	Charis SIL
fonts-urw-base35	V:189, I:536	15560	56	60	40	URW Nimbus (Nimbus Sans , Roman No. 9 L , Mono L , MCAHTC)
fonts-ubuntu	V:2, I:5	4339	58	-	33 P	Ubuntu fonts (display)
fonts-terminus	I:4	452	-	-	33	Cool retro terminal fonts
ttf-mscorefonts-installer	V:0, I:43	85	56?	60	40	Downloader of Microsoft non-free fonts (see below)

Tabela 7.4: List of notable [TrueType](#) and [OpenType](#) fonts

7.6.2 Font rasterization

Debian uses [FreeType](#) to rasterize fonts. Its font choice infrastructure is provided by the [Fontconfig](#) font configuration library.

pachet	popcon(popularitate)	nr. de pachete	descriere
libfreetype6	V:578, I:996	1021	FreeType font rasterization library
libfontconfig1	V:569, I:831	624	Fontconfig font configuration library
fontconfig	V:459, I:708	681	<code>fc - *</code> : CLI commands for Fontconfig
font-manager	V:2, I:7	1121	Font Manager : GUI command for Fontconfig
nautilus-font-manager	V:0, I:0	39	Nautilus extension for Font Manager

Tabela 7.5: List of notable font environment and related packages

Indicație

Some font packages such as `fonts-noto*` install too many fonts. You may also want to keep some font packages installed but disabled under the normal use situation. The multiple [glyphs](#) are expected for some [Unicode](#) code points due to [Han unification](#) and unwanted glyphs may be chosen by the unconfigured [Fontconfig](#) library. One of the most annoying case is "U+3001 IDEOGRAPHIC COMMA" and "U+3002 IDEOGRAPHIC FULL STOP" among CJK countries. You can avoid this problematic situation easily by configuring font availability using [Font Manager](#) GUI ([font-manager](#)).

You can list font configuration state from the command line, too.

- `"fc-match(1)"` for fontconfig font default
- `"fc-list(1)"` for available fontconfig fonts

You can configure font configuration state from the text editor but this is non-trivial. See `fonts.conf(5)`.

7.7 Cutia cu nisip (sandbox)

Many mostly GUI applications on Linux are available in binary formats from non-Debian sources.

- [AppImage -- Linux apps that run anywhere](#)
- [FLATHUB -- Apps for Linux, right here](#)
- [snapcraft -- The app store for Linux](#)



Avertisment

Binaries from these sites may include proprietary non-free software packages.

There is some *raison d'être* for these binary format distributions for Free Software aficionados using Debian since these can accommodate clean set of libraries used for each application by the respective upstream developer independent of the ones provided by Debian.

The inherent risk of running external binaries can be reduced by using the [sandbox environment](#) which leverages modern Linux security features (see Secțiune [4.7.5](#)).

- For binaries from AppImage and some upstream sites, run them in [firejail](#) with [manual configuration](#).
- For binaries from FLATHUB, run them in [Flatpak](#) . (No manual configuration required.)
- For binaries from snapcraft, run them in [Snap](#) . (No manual configuration required. Compatible with daemon programs.)

The `xdg-desktop-portal` package provides a standardized API to common desktop features. See [xdg-desktop-portal \(flatpak\)](#) and [xdg-desktop-portal \(snap\)](#) .

pachet	popcon	popularity	descriere
flatpak	V:97, I:103	8284	Flatpak application deployment framework for desktop apps
gnome-software-plugin-flatpak	V:29, I:40	285	Flatpak support for GNOME Software
snapd	V:67, I:70	71722	Daemon and tooling that enable snap packages
gnome-software-plugin-snap	V:1, I:2	144	Snap support for GNOME Software
xdg-desktop-portal	V:359, I:442	2169	desktop integration portal for Flatpak and Snap
xdg-desktop-portal-gtk	V:326, I:441	715	xdg-desktop-portal backend for gtk (GNOME)
xdg-desktop-portal-kde	V:78, I:111	2688	xdg-desktop-portal backend for Qt (KDE)
xdg-desktop-portal-wlr	V:1, I:6	160	xdg-desktop-portal backend for wlroots (Wayland)
firejail	V:1, I:4	1759	a SUID security sandbox program firejail for use with AppImage

Tabela 7.6: List of notable sandbox environment and related packages

This sandbox environment technology is very much like apps on smart phone OS where apps are executed under controlled resource accesses.

Some large GUI applications such as web browsers on Debian also use sandbox environment technology internally to make them more secure.

7.8 Mediu de birou la distanță

7.9 X server connection

There are several ways to connect from an application on a remote host to the X server including `xwayland` on the local host.

7.9.1 X server local connection

Access to the local X server by the local applications which use X core protocol can be locally connected through a local UNIX domain socket. This can be authorized by the authority file holding [access cookie](#). The authority file location is identified by the `$XAUTHORITY` environment variable and X display is identified by the `$DISPLAY` environment variable. Since these are normally set automatically, no special action is needed, e.g. `gitk` as the following.

```
username $ gitk
```

Notă

For `xwayland`, `XAUTHORITY` holds value like `"/run/user/1000/.mutter-Xwaylandauth.YVSU30"`.

pachet	popcon(popularitate)	limita	protocol	descriere
gnome-remote-desktop	V:110, I:246	2215	RDP	GNOME Remote Desktop server
xrdp	V:25, I:28	4506	RDP	xrdp, Remote Desktop Protocol (RDP) server
x11vnc	V:8, I:41	1835	RFB (VNC)	x11vnc, Remote Framebuffer Protocol (VNC) server
tigervnc-standalone-server	V:4, I:14	2967	RFB (VNC)	TigerVNC, Remote Framebuffer Protocol (VNC) server
gnome-connections	V:5, I:112	1599	RDP, RFB (VNC)	GNOME remote desktop client
vinagre	V:1, I:30	4249	RDP, RFB (VNC), SPICE, SSH	Vinagre: GNOME remote desktop client
remmina	V:14, I:65	971	RDP, RFB (VNC), SPICE, SSH, ...	Remmina: GTK remote desktop client
krdc	V:1, I:17	4052	RDP, RFB (VNC)	KRDC: KDE remote desktop client
virt-viewer	V:5, I:46	1278	RFB (VNC), SPICE	Virtual Machine Manager's GUI display client of guest OS

Tabela 7.7: List of notable remote access server

pachet	popcon(popularitate)	limita	comanda	descriere
openssh-server	V:752, I:808	3501	sshd cu opțiunea X11-forwarding	SSH server (secure)
openssh-client	V:902, I:996	5131	ssh -X	SSH client (secure)
xauth	V:183, I:970	81	xauth	X authority file utility
x11-xserver-utils	V:306, I:537	559	xhost	server access control for X

Tabela 7.8: List of connection methods to the X server

7.9.2 X server remote connection

Access to the local X server display from the remote applications which use X core protocol is supported by using the X11 forwarding feature.

- Open an `gnome-terminal` on the local host.
- Run `ssh(1)` with `-X` option to establish a connection with the remote site as the following.

```
localname @ localhost $ ssh -q -X loginname@remotehost.domain
Password:
```

- Run an X application command, e.g. `"gitk"`, on the remote site as the following.

```
loginname @ remotehost $ gitk
```

This method can display the output from a remote X client as if it were locally connected through a local UNIX domain socket.

See Secțiune 6.3 for SSH/SSHD.



Avertisment

A remote [TCP/IP](#) connection to the X server is disabled by default on the Debian system for security reasons. Don't enable them by simply setting `"xhost +"` nor by enabling [XDMCP connection](#), if you can avoid it.

7.9.3 X server chroot connection

Access to the X server by the applications which use X core protocol and run on the same host but in an environment such as `chroot` where the authority file is not accessible, can be authorized securely with `xhost` by using the [User-based access](#), e.g. `"gitk"` as the following.

```
username $ xhost + si:localuser:root ; sudo chroot /path/to
# cd /src
# gitk
# exit
username $ xhost -
```

7.10 Clipboard

For clipping text to clipboard, see Secțiune 1.4.4.

For clipping graphics to clipboard, see Secțiune 11.6.

Some CLI commands can manipulate character clipboard (PRIMARY and CLIPBOARD), too.

pachet	popcon(popularitate)	dimensiunea pachetului	target	descriere
xsel	V:7, I:43	55	X	command line interface to X selections (clipboard)
xclip	V:13, I:74	62	X	command line interface to X selections (clipboard)
wl-clipboard	V:6, I:22	162	Wayland	wl-copy wl-paste: command line interface to Wayland clipboard
gpm	V:9, I:10	545	Consola Linux	a daemon that captures mouse events on Linux console

Tabela 7.9: List of programs related to manipulating character clipboard

Capitolul 8

I18N și L10N

[Multilingualization \(M17N\) or Native Language Support](#) for an application software is done in 2 steps.

- Internationalization (I18N): To make a software potentially handle multiple locales.
- Localization (L10N): To make a software handle an specific locale.

Indicație

There are 17, 18, or 10 letters between "m" and "n", "i" and "n", or "l" and "n" in multilingualization, internationalization, and localization which correspond to M17N, I18N, and L10N. See [Internationalization and localization](#) for details.

8.1 The locale

The behavior of programs supporting internationalization are configured by the environment variable "\$LANG" to support localization. Actual support of locale dependent features by the `libc` library requires to install `locales` or `locales-all` packages. The `locales` package requires to be initialized properly.

If neither `locales` or `locales-all` package are installed, support of locale features are lost and system uses US English messages and handles data as **ASCII**. This behavior is the same way as "\$LANG" is set by "LANG=", "LANG=C", or "LANG=POSIX".

The modern software such as GNOME and KDE are multilingualized. They are internationalized by making them handle **UTF-8** data and localized by providing their translated messages through the `gettext(1)` infrastructure. Translated messages may be provided as separate localization packages.

The current Debian desktop GUI system normally sets the locale under GUI environment as "LANG=xx_YY.UTF-8". Here, "xx" is [ISO 639 language codes](#) and "YY" is [ISO 3166 country codes](#). These values are set by the desktop configuration GUI dialogue and change the program behavior. See [Secțiune 1.5.2](#)

8.1.1 Rationale for UTF-8 locale

The simplest representation of the text data is **ASCII** which is sufficient for English and uses less than 127 characters (representable with 7 bits).

Even plain English text may contain non-ASCII characters, e.g. slightly curly left and right quotation marks are not available in ASCII.

```
b'"b'double quoted textb'"b' is not "double quoted ASCII"
b''b'single quoted textb''b' is not 'single quoted ASCII'
```

In order to support more characters, many character sets and encoding systems have been used to support many languages (see Tabel 11.2).

Unicode character set can represent practically all characters known to human with 21 bit code point range (i.e., 0 to 10FFFF in hexadecimal notation).

Text encoding system UTF-8 fits Unicode code points into a sensible 8 bit data stream mostly compatible with the ASCII data processing system. This makes UTF-8 the modern preferred choice. UTF stands for Unicode Transformation Format. When ASCII plain text data is converted to UTF-8 one, it has exactly the same content and size as the original ASCII one. So you loose nothing by deploying UTF-8 locale.

Under UTF-8 locale with the compatible application program, you can display and edit any foreign language text data as long as required fonts and input methods are installed and enabled. For example under "LANG=fr_FR.UTF-8" locale, `gedit(1)` (text editor for the GNOME desktop) can display and edit Chinese character text data while presenting menus in French.

Indicație

Both the new standard "en_US.UTF-8" locale and the old standard "C"/"POSIX" locale use the standard US English message, they have subtle differences in sorting order etc. If you want to handle not only ASCII characters but also handle all UTF-8 encoded characters gracefully while maintaining the old "C" local behavior, use the non-standard "C.UTF-8" locale on Debian.

Notă

Some programs consume more memory after supporting I18N. This is because they are coded to use UTF-32(UCS4) internally to support Unicode for speed optimization and consume 4 bytes per each ASCII character data independent of locale selected. Again, you loose nothing by deploying UTF-8 locale.

8.1.2 The reconfiguration of the locale

In order for the system to access a particular locale, the locale data must be compiled from the locale database.

The `locales` package does **not** come with pre-compiled locale data. You need to configure it as:

```
# dpkg-reconfigure locales
```

This process involves 2 steps.

1. Select all required locale data to be compiled into the binary form. (Please make sure to include at least one UTF-8 locale)
2. Set the system wide default locale value by creating `/etc/default/locale` for use by PAM (see Secțiune 4.5).

The system wide default locale value set in `/etc/default/locale` may be overridden by the GUI configuration for GUI applications.

Notă

Actual traditional encoding system can be identified by `/usr/share/i18n/SUPPORTED`. Thus, the "LANG=en_US" is "LANG=en_US.ISO-8859-1".

The `locales-all` package comes with pre-compiled locale data for all locale data. Since it doesn't create `/etc/default/locale`, you may still need to install the `locales` package, too.

Indicație

The `locales` package of some Debian derivative distributions come with pre-compiled locale data for all locale data. You need to install both `locales` and `locales-all` packages on Debian to emulate such system environment.

8.1.3 Codificarea numelor de fișiere

For cross platform data exchanges (see Secțiune 10.1.7), you may need to mount some filesystem with particular encodings. For example, mount(8) for [vfat filesystem](#) assumes [CP437](#) if used without option. You need to provide explicit mount option to use [UTF-8](#) or [CP932](#) for filenames.

Notă

When auto-mounting a hot-pluggable USB memory stick under modern desktop environment such as GNOME, you may provide such mount option by right clicking the icon on the desktop, click "Drive" tab, click to expand "Setting", and entering "utf8" to "Mount options:". The next time this memory stick is mounted, mount with UTF-8 is enabled.

Notă

If you are upgrading system or moving disk drives from older non-UTF-8 system, file names with non-ASCII characters may be encoded in the historic and deprecated encodings such as [ISO-8859-1](#) or [eucJP](#). Please seek help of text conversion tools to convert them to [UTF-8](#). See Secțiune 11.1.

[Samba](#) uses Unicode for newer clients (Windows NT, 200x, XP) but uses [CP850](#) for older clients (DOS and Windows 9x/Me) as default. This default for older clients can be changed using "dos charset" in the "/etc/samba/smb.conf" file, e.g., to [CP932](#) for Japanese.

8.1.4 Localized messages and translated documentation

Translations exist for many of the text messages and documents that are displayed in the Debian system, such as error messages, standard program output, menus, and manual pages. [GNU gettext\(1\) command tool chain](#) is used as the backend tool for most translation activities.

Under "Tasks" → "Localization" [aptitude\(8\)](#) provides an extensive list of useful binary packages which add localized messages to applications and provide translated documentation.

For example, you can obtain the localized message for manpage by installing the *manpages-*LANG** package. To read the Italian-language manpage for *programname* from "/usr/share/man/it/", execute as the following.

```
LANG=it_IT.UTF-8 man programname
```

GNU gettext can accommodate priority list of translation languages with \$LANGUAGE environment variable. For example:

```
$ export LANGUAGE="pt:pt_BR:es:it:fr"
```

For more, see `info gettext` and read the section "The LANGUAGE variable".

8.1.5 Effects of the locale

The sort order of characters with `sort(1)` and `ls(1)` are affected by the locale. Exporting `LANG=en_US.UTF-8` sorts in the dictionary A->a->B->b . . . ->Z->z order, while exporting `LANG=C.UTF-8` sorts in ASCII binary A->B->. . . ->Z->a->b order.

The date format of `ls(1)` is affected by the locale (see Secțiune 9.3.4).

The date format of `date(1)` is affected by the locale. For example:

```
$ unset LC_ALL
$ LANG=en_US.UTF-8 date
Thu Dec 24 08:30:00 PM JST 2023
$ LANG=en_GB.UTF-8 date
Thu 24 Dec 20:30:10 JST 2023
$ LANG=es_ES.UTF-8 date
jue 24 dic 2023 20:30:20 JST
$ LC_TIME=en_DK.UTF-8 date
2023-12-24T20:30:30 JST
```

Number punctuation are different for locales. For example, in English locale, one thousand point one is displayed as "1,000.1" while in German locale, it is displayed as "1.000,1". You may see this difference in spreadsheet program.

Each detail feature of "\$LANG" environment variable may be overridden by setting "\$LC_*" variables. These environment variables can be overridden again by setting "\$LC_ALL" variable. See `locale(7)` manpage for the details. Unless you have strong reason to create complicated configuration, please stay away from them and use only "\$LANG" variable set to one of the UTF-8 locales.

8.2 Introducere de la tastatură

8.2.1 The keyboard input for Linux console and X Window

The Debian system can be configured to work with many international keyboard arrangements using the `keyboard-configuration` and `console-setup` packages.

```
# dpkg-reconfigure keyboard-configuration
# dpkg-reconfigure console-setup
```

For the Linux console and the X Window system, this updates configuration parameters in `/etc/default/keyboard` and `/etc/default/console-setup`. This also configures the Linux console font. Many non-ASCII characters including accented characters used by many European languages can be made available with [dead key](#), [AltGr key](#), and [compose key](#).

8.2.2 The keyboard input for Wayland

For GNOME on Wayland desktop system, Secțiune 8.2.1 can't support non-English European languages. [IBus](#) was made to support not only Asian languages but also European languages. The package dependency of GNOME desktop Environment recommends "ibus" via "gnome-shell". The code of "ibus" has been updated to integrate `setxkbmap` and `XKB` option functionalities. You need to configure ibus from "GNOME Settings" or "GNOME Tweaks" for the multilingualized keyboard input.

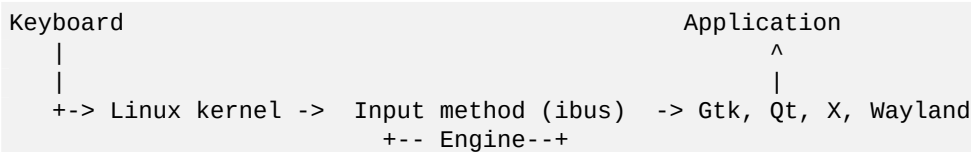
Notă

If ibus is active, your classic X keyboard configuration by the `setxkbmap` may be overridden by ibus even under classic X-based desktop environment. You can disable installed ibus using `im-config` to set input method to "None". For more, see [Debian Wiki on keyboard](#).

8.2.3 The input method support with IBus

Since GNOME desktop Environment recommends "ibus" via "gnome-shell", "ibus" is the good choice for input method.

Multilingual input to the application is processed as:



The list of IBus and its engine packages are the following.

pachet	popcon(popularity)	popularity	supported locale
ibus	V:215, I:260	1857	input method framework using dbus
ibus-mozc	V:2, I:3	980	japoneză
ibus-anthy	V:0, I:1	8900	, ,
ibus-skk	V:0, I:0	243	, ,
ibus-kkc	V:0, I:0	211	, ,
ibus-libpinyin	V:1, I:4	2768	chineză (pentru zh_CN)
ibus-chewing	V:0, I:0	288	, , (pentru zh_TW)
ibus-libzhuyin	V:0, I:0	41008	, , (pentru zh_TW)
ibus-rime	V:0, I:0	78	, , (pentru zh_CN/zh_TW)
ibus-cangjie	V:0, I:0	235	, , (pentru zh_HK)
ibus-hangul	V:0, I:2	261	coreeană
ibus-libthai	V:0, I:0	84	thailandeză
ibus-table-thai	I:0	58	thailandeză
ibus-unikey	V:0, I:0	286	vietnameză
ibus-keyman	V:0, I:0	191	Multilingual: Keyman engine for over 2000 languages
ibus-table	V:0, I:1	2271	table engine for IBus
ibus-m17n	I:1	448	Multilingual: Indic, Arabic and others
plasma-widgets-addons	V:66, I:113	5132	additional widgets for Plasma 5 containing Keyboard Indicator

Tabela 8.1: List of IBus and its engine packages

Notă

For Chinese, "fcitx5" may be an alternative input method framework. For Emacs aficionados, "uim" may be an alternative. Either cases, you may need to do extra manual configuration with `im-config`. Some old classic [input methods](#) such as "kinput2" may still exist in Debian repository but are not recommended for the modern environment.

8.2.4 Un exemplu pentru japoneză

I find the Japanese input method started under English environment ("en_US.UTF-8") very useful. Here is how I did this with IBus for GNOME on Wayland:

1. Install the Japanese input tool package `ibus-mozc` (or `ibus-anthy`) with its recommended packages such as `im-config`.
2. Select "Settings" → "Keyboard" → "Input Sources" → click "+" in "Input Sources" → "Japanese" → "Japanese mozc (or anthy)" and click "Add" if it hasn't been activated.
3. You may chose as many input sources.
4. Relogin to user's account.
5. Setup each input source by right clicking the GUI toolbar icon.

6. Switch among installed input sources by SUPER-SPACE. (SUPER is normally the Windows key.)

Indicație

If you wish to have access to alphabet only keyboard environment with the physical Japanese keyboard on which shift-2 has " (double quotation mark) engraved, you select "Japanese" in the above procedure. You can enter Japanese using "Japanese mozc (or anthy)" with physical "US" keyboard on which shift-2 has @ (at mark) engraved.

- The GUI menu entry for `im-config(8)` is "Input method".
- Alternatively, execute "`im-config`" from user's shell.
- `im-config(8)` behaves differently if command is executed from root or not.
- `im-config(8)` enables the best input method on the system as default without any user actions.

8.3 The display output

Linux console can only display limited characters. (You need to use special terminal program such as `jfbterm(1)` to display non-European languages on the non-GUI console.)

GUI environment (Cap. 7) can display any characters in the UTF-8 as long as required fonts are installed and enabled. (The encoding of the original font data is taken care and transparent to the user.)

8.4 East Asian Ambiguous Character Width Characters

Under the East Asian locale, the box drawing, Greek, and Cyrillic characters may be displayed wider than your desired width to cause the unaligned terminal output (see [Unicode Standard Annex #11](#)).

You can work around this problem:

- `gnome-terminal`: Preferences → Profiles → *Profile name* → Compatibility → Ambiguous-wide characters → Narrow
 - `ncurses`: Set environment export `NCURSES_NO_UTF8_ACS=0`.
-

Capitolul 9

System tips

Here, I describe basic tips to configure and manage systems, mostly from the console.

9.1 The console tips

There are some utility programs to help your console activities.

pachet	popcon	popularity	descriere
mc	V:42, I:189	1590	See Secțiune 1.3
bsdutils	V:429, I:999	335	<code>script(1)</code> command to make a record of terminal session
screen	V:55, I:205	1009	terminal multiplexer with VT100/ANSI terminal emulation
tmux	V:72, I:154	1292	terminal multiplexer alternative (Use "Control-B" instead)
fzf	V:7, I:30	4651	fuzzy text finder
fzy	V:0, I:0	59	fuzzy text finder
rlwrap	V:1, I:12	328	readline feature command line wrapper
ledit	V:0, I:8	375	readline feature command line wrapper
rlfe	V:0, I:0	45	readline feature command line wrapper
ripgrep	V:8, I:28	5342	fast recursive string search in the source code tree with automatic filtering

Tabela 9.1: List of programs to support console activities

9.1.1 Recording the shell activities cleanly

The simple use of `script(1)` (see Secțiune [1.4.9](#)) to record shell activity produces a file with control characters. This can be avoided by using `col(1)` as the following.

```
$ script
Script started, file is typescript
```

Do whatever ... and press `Ctrl-D` to exit `script`.

```
$ col -bx < typescript > cleanedfile
$ vim cleanedfile
```

There are alternative methods to record the shell activities:

- Use `tee` (usable during the boot process in the `initramfs`):

```
$ sh -i 2>&1 | tee typescript
```

- Use `gnome-terminal` with the `extend_line_buffer` for scrollbar.
- Use `screen` with `"^A H"` (see Secțiune 9.1.2) to perform recording of console.
- Use `vim` with `":terminal"` to enter the terminal mode. Use `"Ctrl-W N"` to exit from terminal mode to normal mode. Use `":w typescript"` to write the buffer to a file.
- Use `emacs` with `"M-x shell"`, `"M-x eshell"`, or `"M-x term"` to enter recording console. Use `"C-x C-w"` to write the buffer to a file.

9.1.2 Programul «screen»

`screen(1)` not only allows one terminal window to work with multiple processes, but also allows **remote shell process to survive interrupted connections**. Here is a typical use scenario of `screen(1)`.

1. You login to a remote machine.
2. You start `screen` on a single console.
3. You execute multiple programs in `screen` windows created with `^A c` ("Control-A" followed by "c").
4. You switch among the multiple `screen` windows by `^A n` ("Control-A" followed by "n").
5. Suddenly you need to leave your terminal, but you don't want to lose your active work by keeping the connection.
6. You may **detach** the `screen` session by any methods.
 - Brutally unplug your network connection
 - Type `^A d` ("Control-A" followed by "d") and manually logging out from the remote connection
 - Type `^A DD` ("Control-A" followed by "DD") to have `screen` detach and log you out
7. You log in again to the same remote machine (even from a different terminal).
8. You start `screen` as `"screen -r"`.
9. `screen` magically **reattaches** all previous `screen` windows with all actively running programs.

Indicație

You can save connection fees with `screen` for metered network connections such as dial-up and packet ones, because you can leave a process active while disconnected, and then re-attach it later when you connect again.

In a `screen` session, all keyboard inputs are sent to your current window except for the command keystroke. All `screen` command keystrokes are entered by typing `^A` ("Control-A") plus a single key [plus any parameters]. Here are important ones to remember.

Pentru detalii, consultați `screen(1)`.

See `tmux(1)` for functionalities of the alternative command.

comanda/funcția asociată	semnificație
<code>^A ?</code>	show a help screen (display key bindings)
<code>^A c</code>	create a new window and switch to it
<code>^A n</code>	go to next window
<code>^A p</code>	go to previous window
<code>^A 0</code>	go to window number 0
<code>^A 1</code>	go to window number 1
<code>^A w</code>	show a list of windows
<code>^A a</code>	send a Ctrl-A to current window as keyboard input
<code>^A h</code>	write a hardcopy of current window to file
<code>^A H</code>	begin/end logging current window to file
<code>^A ^X</code>	lock the terminal (password protected)
<code>^A d</code>	detach screen session from the terminal
<code>^A DD</code>	detach screen session and log out

Tabela 9.2: List of key bindings for screen

9.1.3 Navigating around directories

In Secțiune 1.4.2, 2 tips to allow quick navigation around directories are described: `$CDPATH` and `mc`.

If you use fuzzy text filter program, you can do without typing the exact path. For `fzf`, include following in `~/ .bashrc`.

```
FZF_KEYBINDINGS_PATH=/usr/share/doc/fzf/examples/key-bindings.bash
if [ -f $FZF_KEYBINDINGS_PATH ]; then
  . $FZF_KEYBINDINGS_PATH
fi
```

De exemplu:

- You can jump to a very deep subdirectory with minimal efforts. You first type `"cd **"` and press Tab. Then you will be prompted with candidate paths. Typing in partial path strings, e.g., `s/d/b foo`, will narrow down candidate paths. You select the path to be used by `cd` with cursor and return keys.
- You can select a command from the command history more efficiently with minimal efforts. You press `Ctrl-R` at the command prompt. Then you will be prompted with candidate commands. Typing in partial command strings, e.g., `vim d`, will narrow down candidates. You select the one to be used with cursor and return keys.

9.1.4 Readline wrapper

Some commands such as `/usr/bin/dash` which lacks command line history editing capability can add such functionality transparently by running under `rlwrap` or its equivalents.

```
$ rlwrap dash -i
```

This provides convenient platform to test subtle points for `dash` with friendly bash-like environment.

9.1.5 Scanning the source code tree

The `rg(1)` command in the `ripgrep` package offers a faster alternative to the `grep(1)` command for scanning the source code tree for typical situation. It takes advantage of modern multi-core CPUs and automatically applies reasonable filters to skip some files.

9.2 Personalizarea vim

After you learn basics of vim(1) through Secțiune 1.4.8, please read Bram Moolenaar's ["Seven habits of effective text editing \(2000\)"](#) to understand how vim should be used.

9.2.1 Personalizarea vim cu caracteristicile interne

The behavior of vim can be changed significantly by enabling its internal features through the Ex-mode commands such as "set ..." to set vim options.

These Ex-mode commands can be included in user's vimrc file, traditional "~/.vimrc" or git-friendly "~/.vim/vimrc". Here is a very simple example 1:

```
"""" Generic baseline Vim and Neovim configuration (~/.vimrc)
"""" - For NeoVim, use "nvim -u ~/.vimrc [filename]"
""""
let mapleader = ' ' " :h mapleader
""""
set nocompatible " :h 'cp -- sensible (n)vim mode
syntax on " :h :syn-on
filetype plugin indent on " :h :filetype-overview
set encoding=utf-8 " :h 'enc (default: latin1) -- sensible encoding
"""" current vim option value can be verified by :set encoding?
set backspace=indent,eol,start " :h 'bs (default: nobs) -- sensible BS
set statusline=%<%f%m%r%h%w%=%y[U+%04B]%2l/%2L=%P,%2c%V
set listchars=eol:␣,tab:b'␣b'\ ,extends:b'␣b',precedes:b'␣b',nbsp:b'␣b'
set viminfo=!,100,<5000,s100,h " :h 'vi -- bigger copy buffer etc.
"""" Pick "colorscheme" from blue darkblue default delek desert elflord evening
"""" habamax industry koehler lunaperche morning murphy pablo peachpuff quiet ron
"""" shine slate torte zellner
colorscheme industry
"""" don't pick "colorscheme" as "default" which may kill SpellUnderline settings
set scrolloff=5 " :h 'scr -- show 5 lines around cursor
set laststatus=2 " :h 'ls (default 1) k
"""" boolean options can be unset by prefixing "no"
set ignorecase " :h 'ic
set smartcase " :h 'scs
set autoindent " :h 'ai
set smartindent " :h 'si
set nowrap " :h 'wrap
"set list " :h 'list (default nolist)
set noerrorbells " :h 'eb
set novisualbell " :h 'vb
set t_vb= " :h 't_vb -- termcap visual bell
set spell " :h 'spell
set spelllang=en_us,cjk " :h 'spl -- english spell, ignore CJK
set clipboard=unnamedplus " :h 'cb -- cut/copy/paste with other app
set hidden " :h 'hid
set autowrite " :h 'aw
set timeoutlen=300 " :h 'tm
```

The keymap of vim can be changed in user's vimrc file. E.g.:



Atenție

Don't try to change the default key bindings without very good reasons.

1More elaborate customization examples: ["Vim Galore"](#), ["sensible.vim"](#), ...

```

"""" Popular mappings (imitating LazyVim etc.)
"""" Window moves without using CTRL-W which is dangerous in INSERT mode
nnoremap <C-H> <C-W>h
nnoremap <C-J> <C-W>j
nnoremap <C-K> <C-W>k
silent! nnoremap <C-L> <C-W>l
"""" Window resize
nnoremap <C-LEFT> <CMD>vertical resize -2<CR>
nnoremap <C-DOWN> <CMD>resize -2<CR>
nnoremap <C-UP> <CMD>resize +2<CR>
nnoremap <C-RIGHT> <CMD>vertical resize +2<CR>
"""" Clear hlsearch with <ESC> (<C-L> is mapped as above)
nnoremap <ESC> <CMD>noh<CR><ESC>
inoremap <ESC> <CMD>noh<CR><ESC>
"""" center after jump next
nnoremap n nzz
nnoremap N Nzz
"""" fast "jk" to get out of INSERT mode (<ESC>)
inoremap jk <CMD>noh<CR><ESC>
"""" fast "<ESC><ESC>" to get out of TERM mode (CTRL-\ CTRL-N)
tnoremap <ESC><ESC> <C-\><C-N>
"""" fast "jk" to get out of TERM mode (CTRL-\ CTRL-N)
tnoremap jk <C-\><C-N>
"""" previous/next trouble/quickfix item
nnoremap [q <CMD>cprevious<CR>
nnoremap ]q <CMD>cnext<CR>
"""" buffers
nnoremap <S-H> <CMD>bprevious<CR>
nnoremap <S-L> <CMD>bnext<CR>
nnoremap [b <CMD>bprevious<CR>
nnoremap ]b <CMD>bnext<CR>
"""" Add undo break-points
inoremap , ,<C-G>u
inoremap . .<C-G>u
inoremap ; ;<C-G>u
"""" save file
inoremap <C-S> <CMD>w<CR><ESC>
xnoremap <C-S> <CMD>w<CR><ESC>
nnoremap <C-S> <CMD>w<CR><ESC>
snoremap <C-S> <CMD>w<CR><ESC>
"""" better indenting
vnoremap < <gv
vnoremap > >gv
"""" terminal (Somehow under Linux, <C-/> becomes <C-_> in Vim)
nnoremap <C-_> <CMD>terminal<CR>
nnoremap <C-/> <CMD>terminal<CR>
""""
if ! has('nvim')
"""" Toggle paste mode with <SPACE>p for Vim (no need for Nvim)
set pastetoggle=<leader>p
"""" nvim default mappings for Vim. See :h default-mappings in nvim
"""" copy to EOL (no delete) like D for d
noremap Y y$
"""" sets a new undo point before deleting
inoremap <C-U> <C-G>u<C-U>
inoremap <C-W> <C-G>u<C-W>
"""" <C-L> is re-purposed as above
"""" execute the previous macro recorded with Q
nnoremap Q @@
"""" repeat last substitute and *KEEP* flags
nnoremap & :&&<CR>

```

```

""" search visual selected string for visual mode
xnoremap * y/\V<C-R>"<CR>
xnoremap # y?/\V<C-R>"<CR>
endif

```

In order for the above keybindings to function properly, the terminal program needs to be configured to generate "ASCII DEL" for Backspace-key and "Escape sequence" for Delete-key.

Other miscellaneous configuration can be changed in user's vimrc file. E.g.:

```

""" Use faster 'rg' (ripgrep package) for :grep
if executable("rg")
  set grepprg=rg\ --vimgrep\ --smart-case
  set grepformat=%f:%l:%c:%m
endif
=====
""" Retain last cursor position :h ''
augroup RetainLastCursorPosition
  autocmd!
  autocmd BufReadPost *
    \ if line("\'") > 0 && line("\'") <= line("$") |
    \   exe "normal! g'\'" |
    \ endif
augroup END
=====
""" Force to use underline for spell check results
augroup SpellUnderline
  autocmd!
  autocmd ColorScheme * highlight SpellBad term=Underline gui=Undercurl
  autocmd ColorScheme * highlight SpellCap term=Underline gui=Undercurl
  autocmd ColorScheme * highlight SpellLocal term=Underline gui=Undercurl
  autocmd ColorScheme * highlight SpellRare term=Underline gui=Undercurl
augroup END
=====
""" highlight trailing spaces except when typing as red (set after colorscheme)
highlight TailingWhitespaces ctermbg=red guibg=red
""" \s\+      1 or more whitespace character: <Space> and <Tab>
""" \%#\@<! Matches with zero width if the cursor position does NOT match.
match TailingWhitespaces /\s\+\%#\@<!$/

```

9.2.2 Personalizarea vim cu pachete externe

Interesting external plugin packages can be found:

- [Vim - the ubiquitous text editor](#) -- The official upstream site of Vim and vim scripts
- [VimAwesome](#) -- The listing of Vim plugins
- [vim-scripts](#) -- Debian package: a collection of vim scripts

Plugin packages in the [vim-scripts](#) package can be enabled using user's vimrc file. E.g.:

```

packadd! secure-modelines
packadd! winmanager
" IDE-like UI for files and buffers with <space>w
nnoremap <leader>w          :WMToggle<CR>

```

The new native Vim package system works nicely with "git" and "git submodule". One such example configuration can be found at [my git repository: dot-vim](#). This does essentially:

- By using "git" and "git submodule", latest external packages, such as "*name*", are placed into `~/.vim/pack/*/opt/` and similar.
- By adding `:packadd! name` line to user's vimrc file, these packages are placed on runtimepath.
- Vim loads these packages on runtimepath during its initialization.
- At the end of its initialization, tags for the installed documents are updated with "`helptags ALL`".

For more, please start vim with "`vim --startuptime vimstart.log`" to check actual execution sequence and time spent for each step.

It is quite confusing to see too many ways² to manage and load these external packages to vim. Checking the original information is the best cure.

combinația de taste	informații
<code>:help package</code>	explanation on the vim package mechanism
<code>:help runtimepath</code>	explanation on the runtimepath mechanism
<code>:version</code>	internal states including candidates for the vimrc file
<code>:echo \$VIM</code>	the environment variable "\$VIM" used to locate the vimrc file
<code>:set runtimepath?</code>	list of directories which will be searched for all runtime support files
<code>:echo \$VIMRUNTIME</code>	the environment variable "\$VIMRUNTIME" used to locate various system provided runtime support files

Tabela 9.3: Informații despre inițializarea vim

9.3 Data recording and presentation

9.3.1 Demonul de jurnalizare

Many traditional programs record their activities in the text file format under the `/var/log/` directory.

`logrotate(8)` is used to simplify the administration of log files on a system which generates a lot of log files.

Many new programs record their activities in the binary file format using `systemd-journald(8)` Journal service under the `/var/log/journal` directory.

You can log data to the `systemd-journald(8)` Journal from a shell script by using the `systemd-cat(1)` command.

See Secțiune 3.4 and Secțiune 3.3.

9.3.2 Log analyzer

Here are notable log analyzers ("`~Gsecurity::log-analyzer`" in `aptitude(8)`).

Notă

[CRM114](#) provides language infrastructure to write **fuzzy** filters with the [TRE regex library](#). Its popular use is spam mail filter but it can be used as log analyzer.

²[vim-pathogen](#) was popular.

pachet	popcon(popularitate)	descriere	
logwatch	V:9, I:11	2412	log analyzer with nice output written in Perl
fail2ban	V:97, I:108	2191	ban IPs that cause multiple authentication errors
analog	V:3, I:88	3739	web server log analyzer
awstats	V:5, I:9	6935	powerful and featureful web server log analyzer
sarg	V:0, I:0	863	squid analysis report generator
pflogsumm	V:1, I:4	167	Postfix log entry summarizer
fwlogwatch	V:0, I:0	487	firewall log analyzer
squidview	V:0, I:0	189	monitor and analyze squid access.log files
swatch	V:0, I:0	99	log file viewer with regexp matching, highlighting, and hooks
crm114	V:0, I:0	1365	Controllable Regex Mutilator and Spam Filter (CRM114)
icmpinfo	V:0, I:0	42	interpret ICMP messages

Tabela 9.4: List of system log analyzers

9.3.3 Customized display of text data

Although pager tools such as `more(1)` and `less(1)` (see Secțiune 1.4.5) and custom tools for highlighting and formatting (see Secțiune 11.1.8) can display text data nicely, general purpose editors (see Secțiune 1.4.6) are most versatile and customizable.

Indicație

For `vim(1)` and its pager mode alias `view(1)`, `":set hls"` enables highlighted search.

9.3.4 Customized display of time and date

The default display format of time and date by the `"ls -l"` command depends on the **locale** (see Secțiune 1.2.6 for value). The `"$LANG"` variable is referred first and it can be overridden by the `"$LC_TIME"` or `"$LC_ALL"` exported environment variables.

The actual default display format for each locale depends on the version of the standard C library (the `libc6` package) used. I.e., different releases of Debian had different defaults. For iso-formats, see [ISO 8601](#).

If you really wish to customize this display format of time and date beyond the **locale**, you should set the **time style value** by the `"- -time-style"` argument or by the `"$TIME_STYLE"` value (see `ls(1)`, `date(1)`, `info coreutils 'ls invocation'`).

time style value	configurația regională	afișarea datei și orei
iso	oricare	01-19 00:15
long-iso	oricare	2009-01-19 00:15
full-iso	oricare	2009-01-19 00:15:16.000000000 +0900
locale	C	Jan 19 00:15
locale	en_US.UTF-8	Jan 19 00:15
locale	es_ES.UTF-8	ene 19 00:15
+%d.%m.%y %H:%M	oricare	19.01.09 00:15
+%d.%b.%y %H:%M	C sau en_US.UTF-8	19.Jan.09 00:15
+%d.%b.%y %H:%M	es_ES.UTF-8	19.ene.09 00:15

Tabela 9.5: Display examples of time and date for the `"ls -l"` command with the **time style value**

Indicație

You can eliminate typing long option on commandline using command alias (see Secțiune 1.5.9):

```
alias ls='ls --time-style=+%d.%m.%y %H:%M'
```

9.3.5 Colorized shell echo

Shell echo to most modern terminals can be colorized using [ANSI escape code](#) (see `/usr/share/doc/xterm/ctlseqs`).

De exemplu, încercați următoarele

```
$ RED=$(printf "\x1b[31m")
$ NORMAL=$(printf "\x1b[0m")
$ REVERSE=$(printf "\x1b[7m")
$ echo "${RED}RED-TEXT${NORMAL} ${REVERSE}REVERSE-TEXT${NORMAL}"
```

9.3.6 Comenzi colorate

Colorized commands are handy for inspecting their output in the interactive environment. I include the following in my `~/ .bashrc`.

```
if [ "$TERM" != "dumb" ]; then
    eval "`dircolors -b`"
    alias ls='ls --color=always'
    alias ll='ls --color=always -l'
    alias la='ls --color=always -A'
    alias less='less -R'
    alias ls='ls --color=always'
    alias grep='grep --color=always'
    alias egrep='egrep --color=always'
    alias fgrep='fgrep --color=always'
    alias zgrep='zgrep --color=always'
else
    alias ll='ls -l'
    alias la='ls -A'
fi
```

The use of alias limits color effects to the interactive command usage. It has advantage over exporting environment variable `export GREP_OPTIONS='--color=auto'` since color can be seen under pager programs such as `less(1)`. If you wish to suppress color when piping to other programs, use `--color=auto` instead in the above example for `~/ .bashrc`.

Indicație

You can turn off these colorizing aliases in the interactive environment by invoking shell with `TERM=dumb bash`.

9.3.7 Recording the editor activities for complex repeats

You can record the editor activities for complex repeats.

For [Vim](#), as follows.

- "qa": start recording typed characters into named register "a".

- ... editor activities
- "q": end recording typed characters.
- "@a": execute the contents of register "a".

For [Emacs](#), as follows.

- "C-x (": start defining a keyboard macro.
- ... editor activities
- "C-x)": end defining a keyboard macro.
- "C-x e": execute a keyboard macro.

9.3.8 Recording the graphics image of an X application

There are few ways to record the graphics image of an X application, including an xterm display.

pachet	popcon(popularitate)	nr. de instalatii	descriere
gnome-screenshot	V:13, I:114	1115	Wayland
flameshot	V:7, I:17	3532	Wayland
gimp	V:45, I:232	32135	Wayland + X
x11-apps	V:31, I:467	2461	X
imagemagick	I:295	77	X
scrot	V:4, I:54	141	X

Tabela 9.6: List of graphics image manipulation tools

9.3.9 Recording changes in configuration files

There are specialized tools to record changes in configuration files with help of DVCS and to make system snapshots on [Btrfs](#).

pachet	popcon(popularitate)	nr. de instalatii	descriere
etckeeper	V:24, I:28	157	store configuration files and their metadata with Git (default), Mercurial , or GNU Bazaar
timeshift	V:7, I:13	4234	system restore utility using rsync or BTRFS snapshots
snapper	V:6, I:8	2426	Linux filesystem snapshot management tool

Tabela 9.7: List of packages which can record configuration history

You may also think about local script Secțiune [10.2.3](#) approach.

9.4 Monitoring, controlling, and starting program activities

Program activities can be monitored and controlled using specialized tools.

Indicație

The procs packages provide very basics of monitoring, controlling, and starting program activities. You should learn all of them.

pachet	popcon(popularitate)	descriere
coreutils	V:897, I:999 18457	nice(1): run a program with modified scheduling priority
bsdutils	V:429, I:999 335	renice(1): modify the scheduling priority of a running process
procps	V:821, I:997 2395	"/proc" filesystem utilities: ps(1), top(1), kill(1), watch(1), ...
psmisc	V:403, I:748 950	"/proc" filesystem utilities: killall(1), fuser(1), peekfd(1), pstree(1)
time	V:6, I:90 129	time(1): run a program to report system resource usages with respect to time
sysstat	V:129, I:165 1904	sar(1), iostat(1), mpstat(1), ...: system performance tools for Linux
isag	V:0, I:3 109	Interactive System Activity Grapher for sysstat
lsof	V:438, I:949 492	lsof(8): list files opened by a running process using "-p" option
strace	V:9, I:106 3253	strace(1): trace system calls and signals
ltrace	V:0, I:12 420	ltrace(1): trace library calls
xtrace	V:0, I:0 353	xtrace(1): trace communication between X11 client and server
powertop	V:29, I:226 696	power top(1): information about system power use
cron	V:903, I:996 250	run processes according to a schedule in background from cron(8) daemon
anacron	V:412, I:485 112	cron-like command scheduler for systems that don't run 24 hours a day
at	V:77, I:107 158	at(1) or batch(1): run a job at a specified time or below certain load level

Tabela 9.8: List of tools for monitoring and controlling program activities

9.4.1 Timing a process

Display time used by the process invoked by the command.

```
# time some_command >/dev/null
real    0m0.035s    # time on wall clock (elapsed real time)
user    0m0.000s    # time in user mode
sys     0m0.020s    # time in kernel mode
```

9.4.2 Prioritatea de planificare

A nice value is used to control the scheduling priority for the process.

nice value	prioritatea de planificare
19	lowest priority process (nice)
0	very high priority process for user
-20	very high priority process for root (not-nice)

Tabela 9.9: List of nice values for the scheduling priority

```
# nice -19 top # very nice
# nice --20 wodim -v -eject speed=2 dev=0,0 disk.img # very fast
```

Sometimes an extreme nice value does more harm than good to the system. Use this command carefully.

9.4.3 Comanda «ps»

The `ps(1)` command on a Debian system support both BSD and SystemV features and helps to identify the process activity statically.

style	typical command	feature
BSD	<code>ps aux</code>	afișează %CPU %MEM
System V	<code>ps -efH</code>	afișează PPID

Tabela 9.10: List of `ps` command styles

For the zombie (defunct) children process, you can kill them by the parent process ID identified in the "PPID" field. The `ps tree(1)` command display a tree of processes.

9.4.4 Comanda «top»

`top(1)` on the Debian system has rich features and helps to identify what process is acting funny dynamically.

It is an interactive full screen program. You can get its usage help press by pressing the "h"-key and terminate it by pressing the "q"-key.

9.4.5 Listing files opened by a process

You can list all files opened by a process with a process ID (PID), e.g. 1, by the following.

```
$ sudo lsof -p 1
```

PID=1 is usually the `init` program.

9.4.6 Tracing program activities

You can trace program activity with `strace(1)`, `ltrace(1)`, or `xtrace(1)` for system calls and signals, library calls, or communication between X11 client and server.

You can trace system calls of the `ls` command as the following.

```
$ sudo strace ls
```

Indicație

Use **strace-graph** script found in `/usr/share/doc/strace/examples/` to make a nice tree view

9.4.7 Identification of processes using files or sockets

You can also identify processes using files by `fuser(1)`, e.g. for `"/var/log/mail.log"` by the following.

```
$ sudo fuser -v /var/log/mail.log
                USER      PID ACCESS COMMAND
/var/log/mail.log: root      2946 F.... rsyslogd
```

You see that file `"/var/log/mail.log"` is open for writing by the `rsyslogd(8)` command.

You can also identify processes using sockets by `fuser(1)`, e.g. for `"smtp/tcp"` by the following.

```
$ sudo fuser -v smtp/tcp
USER          PID ACCESS COMMAND
smtp/tcp:      Debian-exim  3379 F.... exim4
```

Now you know your system runs `exim4(8)` to handle [TCP](#) connections to [SMTP](#) port (25).

9.4.8 Repeating a command with a constant interval

`watch(1)` executes a program repeatedly with a constant interval while showing its output in fullscreen.

```
$ watch w
```

This displays who is logged on to the system updated every 2 seconds.

9.4.9 Repeating a command looping over files

There are several ways to repeat a command looping over files matching some condition, e.g. matching glob pattern `"*.ext"`.

- Shell for-loop method (see [Secțiune 12.1.4](#)):

```
for x in *.ext; do if [ -f "$x" ]; then command "$x" ; fi; done
```

- `find(1)` and `xargs(1)` combination:

```
find . -type f -maxdepth 1 -name '*.ext' -print0 | xargs -0 -n 1 command
```

- `find(1)` with `"-exec"` option with a command:

```
find . -type f -maxdepth 1 -name '*.ext' -exec command '{}' \;
```

- `find(1)` with `"-exec"` option with a short shell script:

```
find . -type f -maxdepth 1 -name '*.ext' -exec sh -c "command '{}' && echo 'successful'" \;
```

The above examples are written to ensure proper handling of funny file names such as ones containing spaces. See [Secțiune 10.1.5](#) for more advance uses of `find(1)`.

9.4.10 Starting a program from GUI

For the [command-line interface \(CLI\)](#), the first program with the matching name found in the directories specified in the `$PATH` environment variable is executed. See [Secțiune 1.5.3](#).

For the [graphical user interface \(GUI\)](#) compliant to the [freedesktop.org](#) standards, the `*.desktop` files in the `/usr/share/applications/` directory provide necessary attributes for the GUI menu display of each program. Each package which is compliant to [Freedesktop.org](#)'s xdg menu system installs its menu data provided by `"*.desktop"` under `/usr/share/applications/`. Modern desktop environments which are compliant to [Freedesktop.org](#) standard use these data to generate their menu using the `xdg-utils` package. See `/usr/share/doc/xdg-utils/README`.

For example, the `chromium.desktop` file defines attributes for the "Chromium Web Browser" such as "Name" for the program name, "Exec" for the program execution path and arguments, "Icon" for the icon used, etc. (see the [Desktop Entry Specification](#)) as follows:

```
[Desktop Entry]
Version=1.0
Name=Chromium Web Browser
GenericName=Web Browser
Comment=Access the Internet
Comment[fr]=Explorer le Web
Exec=/usr/bin/chromium %U
Terminal=false
X-MultipleArgs=false
Type=Application
Icon=chromium
Categories=Network;WebBrowser;
MimeType=text/html;text/xml;application/xhtml+xml;x-scheme-handler/http;x-scheme-handler/ ↵
https;
StartupWMClass=Chromium
StartupNotify=true
```

This is an oversimplified description. The *.desktop files are scanned as follows.

The desktop environment sets \$XDG_DATA_HOME and \$XDG_DATA_DIR environment variables. For example, under the GNOME 3:

- \$XDG_DATA_HOME is unset. (The default value of \$HOME/.local/share is used.)
- \$XDG_DATA_DIRS is set to /usr/share/gnome:/usr/local/share/:/usr/share/.

So the base directories (see [XDG Base Directory Specification](#)) and the applications directories are as follows.

- \$HOME/.local/share/ → \$HOME/.local/share/applications/
- /usr/share/gnome/ → /usr/share/gnome/applications/
- /usr/local/share/ → /usr/local/share/applications/
- /usr/share/ → /usr/share/applications/

The *.desktop files are scanned in these applications directories in this order.

Indicație

A user custom GUI menu entry can be created by adding a *.desktop file in the \$HOME/.local/share/applications/ directory.

Indicație

The "Exec=..." line isn't parsed by the shell. Use the env(1) command if environment variables need to be set.

Indicație

Similarly, if a *.desktop file is created in the autostart directory under these base directories, the specified program in the *.desktop file is executed automatically when the desktop environment is started. See [Desktop Application Autostart Specification](#).

Indicație

Similarly, if a *.desktop file is created in the \$HOME/Desktop directory and the Desktop environment is configured to support the desktop icon launcher feature, the specified program in it is executed upon clicking the icon. Please note that the actual name of the \$HOME/Desktop directory is locale dependent. See xdg-user-dirs-update(1).

9.4.11 Customizing program to be started

Some programs start another program automatically. Here are check points for customizing this process.

- Application configuration menu:
 - GNOME3 desktop: "Settings" → "System" → "Details" → "Default Applications"
 - KDE desktop: "K" → "Control Center" → "KDE Components" → "Component Chooser"
 - Iceweasel browser: "Edit" → "Preferences" → "Applications"
 - mc(1): "/etc/mc/mc.ext"
- Environment variables such as "\$BROWSER", "\$EDITOR", "\$VISUAL", and "\$PAGER" (see [environ\(7\)](#))
- The `update-alternatives(1)` system for programs such as "editor", "view", "x-www-browser", "gnome-www-browser" and "www-browser" (see [Secțiune 1.4.7](#))
- the "~/.mailcap" and "/etc/mailcap" file contents which associate [MIME](#) type with program (see [mailcap\(5\)](#))
- The "~/.mime.types" and "/etc/mime.types" file contents which associate file name extension with [MIME](#) type (see [run-mailcap\(1\)](#))

Indicație

`update-mime(8)` updates the "/etc/mailcap" file using "/etc/mailcap.order" file (see [mailcap.order\(5\)](#)).

Indicație

The `debianutils` package provides `sensible-browser(1)`, `sensible-editor(1)`, and `sensible-pager(1)` which make sensible decisions on which editor, pager, and web browser to call, respectively. I recommend you to read these shell scripts.

Indicație

In order to run a console application such as `mutt` under GUI as your preferred application, you should create an GUI application as following and set "/usr/local/bin/mutt-term" as your preferred application to be started as described.

```
# cat /usr/local/bin/mutt-term <<EOF
#!/bin/sh
gnome-terminal -e "mutt \${@}"
EOF
# chmod 755 /usr/local/bin/mutt-term
```

9.4.12 Omorârea unui proces

Use `kill(1)` to kill (or send a signal to) a process by the process ID.

Use `killall(1)` or `pkill(1)` to do the same by the process command name and other attributes.

9.4.13 Scheduling tasks once

Run the `at(1)` command to schedule a one-time job by the following.

```
$ echo 'command -args' | at 3:40 monday
```

valoare semnal	nume semnal	acțiune	notă
0	---	no signal is sent (see <code>kill(2)</code>)	check if process is running
1	SIGHUP	terminate the process	disconnected terminal (signal hang up)
2	SIGINT	terminate the process	interrupt from keyboard (CTRL - C)
3	SIGQUIT	terminate the process and dump core	quit from keyboard (CTRL - \)
9	SIGKILL	terminate the process	unblockable kill signal
15	SIGTERM	terminate the process	blockable termination signal

Tabela 9.11: List of frequently used signals for kill command

9.4.14 Scheduling tasks regularly

Use `cron(8)` to schedule tasks regularly. See `crontab(1)` and `crontab(5)`.

You can schedule to run processes as a normal user, e.g. `foo` by creating a `crontab(5)` file as `/var/spool/cron/crontab.foo` with `"crontab -e"` command.

Here is an example of a `crontab(5)` file.

```
# use /usr/bin/sh to run commands, no matter what /etc/passwd says
SHELL=/bin/sh
# mail any output to paul, no matter whose crontab this is
MAILTO=paul
# Min Hour DayOfMonth Month DayOfWeek command (Day... are OR'ed)
# run at 00:05, every day
5 0 * * * $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# run at 14:15 on the first of every month -- output mailed to paul
15 14 1 * * $HOME/bin/monthly
# run at 22:00 on weekdays(1-5), annoy Joe. % for newline, last % for cc:
0 22 * * 1-5 mail -s "It's 10pm" joe%Joe,%%Where are your kids?%.%%
23 */2 1 2 * echo "run 23 minutes after 0am, 2am, 4am ..., on Feb 1"
5 4 * * sun echo "run at 04:05 every Sunday"
# run at 03:40 on the first Monday of each month
40 3 1-7 * * [ "$(date +%a)" == "Mon" ] && command -args
```

Indicație

For the system not running continuously, install the `anacron` package to schedule periodic commands at the specified intervals as closely as machine-uptime permits. See `anacron(8)` and `anacrontab(5)`.

Indicație

For scheduled system maintenance scripts, you can run them periodically from root account by placing such scripts in `/etc/cron.hourly/`, `/etc/cron.daily/`, `/etc/cron.weekly/`, or `/etc/cron.monthly/`. Execution timings of these scripts can be customized by `/etc/crontab` and `/etc/anacrontab`.

[Systemd](#) has low level capability to schedule programs to run without `cron` daemon. For example, `/lib/systemd/system/apt-daily.service` and `/lib/systemd/system/apt-daily.timer` set up daily apt download activities. See `systemd.timer(5)`.

9.4.15 Scheduling tasks on event

[Systemd](#) can schedule program not only on the timer event but also on the mount event. See [Section 10.2.3.3](#) and [Section 10.2.3.2](#) for examples.

9.4.16 Alt-SysRq key

Pressing `Alt-SysRq` (`PrtScr`) followed by one keys does the magic of rescuing control of the system.

See more on [Linux kernel user's and administrator's guide » Linux Magic System Request Key Hacks](#)

Indicație

From SSH terminal etc., you can use the `Alt-SysRq` feature by writing to the `/proc/sysrq-trigger`. For example, `"echo s > /proc/sysrq-trigger; echo u > /proc/sysrq-trigger"` from the root shell prompt syncs and umounts all mounted filesystems.

key following Alt-SysRq	description of action
k	kill all processes on the current virtual console (SAK)
s	sync all mounted filesystems to avoid data corruption
u	remount all mounted filesystems read-only (u mount)
r	restore the keyboard from raw mode after X crashes

Tabela 9.12: List of notable SAK command keys

The current (2021) Debian amd64 Linux kernel has `/proc/sys/kernel/sysrq=438=0b110110110`:

- 2 = 0x2 - enable control of console logging level (ON)
- 4 = 0x4 - enable control of keyboard (SAK, unraw) (ON)
- 8 = 0x8 - enable debugging dumps of processes etc. (OFF)
- 16 = 0x10 - enable sync command (ON)
- 32 = 0x20 - enable remount read-only (ON)
- 64 = 0x40 - enable signaling of processes (term, kill, oom-kill) (OFF)
- 128 = 0x80 - allow reboot/poweroff (ON)
- 256 = 0x100 - allow nicing of all RT tasks (ON)

9.5 System maintenance tips

9.5.1 Who is on the system?

You can check who is on the system by the following.

- `who(1)` shows who is logged on.
- `w(1)` shows who is logged on and what they are doing.
- `last(1)` shows listing of last logged in user.
- `lastb(1)` shows listing of last bad logged in users.

Indicație

`/var/run/utmp`, and `/var/log/wtmp` hold such user information. See `login(1)` and `utmp(5)`.

9.5.2 Warning everyone

You can send message to everyone who is logged on to the system with `wall(1)` by the following.

```
$ echo "We are shutting down in 1 hour" | wall
```

pachet	popcon(popularitate)	descriere
pciutils	V:252, I:992 279	Utilități PCI Linux: <code>lspci(8)</code>
usbutils	V:77, I:883 322	Utilități USB Linux: <code>lsusb(8)</code>
nvme-cli	V:21, I:31 2222	Utilități NVMe pentru Linux: <code>nvme(1)</code>
pcmciautils	V:4, I:7 92	Utilități PCMCIA pentru Linux: <code>pccardctl(8)</code>
scsictools	V:0, I:2 261	collection of tools for SCSI hardware management: <code>lsscsi(8)</code>
procinfo	V:0, I:6 149	system information obtained from <code>"/proc"</code> : <code>lsdev(8)</code>
lshw	V:13, I:93 971	information about hardware configuration: <code>lshw(1)</code>
discover	V:28, I:727 81	hardware identification system: <code>discover(8)</code>

Tabela 9.13: List of hardware identification tools

9.5.3 Hardware identification

For the [PCI](#)-like devices ([AGP](#), [PCI-Express](#), [CardBus](#), [ExpressCard](#), etc.), `lspci(8)` (probably with `"-nn"` option) is a good start for the hardware identification.

Alternatively, you can identify the hardware by reading contents of `"/proc/bus/pci/devices"` or browsing directory tree under `"/sys/bus/pci"` (see Secțiune [1.2.12](#)).

9.5.4 Hardware configuration

Although most of the hardware configuration on modern GUI desktop systems such as GNOME and KDE can be managed through accompanying GUI configuration tools, it is a good idea to know some basics methods to configure them.

pachet	popcon(popularitate)	descriere
console-setup	V:58, I:971 420	Linux console font and keytable utilities
x11-xserver-utils	V:306, I:537 559	Utilități server X: <code>xset(1)</code> , <code>xmodmap(1)</code>
acpid	V:60, I:96 158	daemon to manage events delivered by the Advanced Configuration and Power Interface (ACPI)
acpi	V:7, I:89 49	utility to display information on ACPI devices
sleepd	V:0, I:0 84	daemon to put a laptop to sleep during inactivity
hdparm	V:116, I:231 246	hard disk access optimization (see Secțiune 9.6.9)
smartmontools	V:227, I:265 2455	control and monitor storage systems using S.M.A.R.T.
setserial	V:3, I:5 104	collection of tools for serial port management
memtest86+	V:0, I:19 12473	collection of tools for memory hardware management
scsictools	V:0, I:2 261	collection of tools for SCSI hardware management
setcd	V:0, I:0 33	compact disc drive access optimization
big-cursor	I:0 26	larger mouse cursors for X

Tabela 9.14: List of hardware configuration tools

Here, [ACPI](#) is a newer framework for the power management system than [APM](#).

Indicație

CPU frequency scaling on modern system is governed by kernel modules such as `acpi_cpufreq`.

9.5.5 System and hardware time

The following sets system and hardware time to MM/DD hh:mm, CCYY.

```
# date MMDDhhmmCCYY
# hwclock --utc --systohc
# hwclock --show
```

Times are normally displayed in the local time on the Debian system but the hardware and system time usually use [UTC\(GMT\)](#).

If the hardware time is set to UTC, change the setting to "UTC=yes" in the "/etc/default/rcS".

The following reconfigure the timezone used by the Debian system.

```
# dpkg-reconfigure tzdata
```

If you wish to update system time via network, consider to use the [NTP](#) service with the packages such as ntp, ntpdate, and chrony.

Indicație

Under [systemd](#), use systemd-timesyncd for the network time synchronization instead. See systemd-timesyncd(8).

See the following.

- [Managing Accurate Date and Time HOWTO](#)
- [NTP Public Services Project](#)
- The ntp-doc package

Indicație

ntptrace(8) in the ntp package can trace a chain of NTP servers back to the primary source.

9.5.6 The terminal configuration

There are several components to configure character console and ncurses(3) system features.

- The "/etc/terminfo/*/*" file (terminfo(5))
- The "\$TERM" environment variable (term(7))
- setterm(1), stty(1), tic(1), și toe(1)

If the terminfo entry for xterm doesn't work with a non-Debian xterm, change your terminal type, "\$TERM", from "xterm" to one of the feature-limited versions such as "xterm-r6" when you log in to a Debian system remotely. See "/usr/share/doc/libncurses5/FAQ" for more. "dumb" is the lowest common denominator for "\$TERM".

9.5.7 Infrastructura de sunet

Device drivers for sound cards for current Linux are provided by [Advanced Linux Sound Architecture \(ALSA\)](#). ALSA provides emulation mode for previous [Open Sound System \(OSS\)](#) for compatibility.

Application softwares may be configured not only to access sound devices directly but also to access them via some standardized sound server system. Currently, PulseAudio, JACK, and PipeWire are used as sound server system. See [Debian wiki page on Sound](#) for the latest situation.

There is usually a common sound engine for each popular desktop environment. Each sound engine used by the application can choose to connect to different sound servers.

Indicație

Use "cat /dev/urandom > /dev/audio" or speaker -test(1) to test speaker (^C to stop).

Indicație

If you can not get sound, your speaker may be connected to a muted output. Modern sound system has many outputs. alsamixer(1) in the alsa-utils package is useful to configure volume and mute settings.

pachet	popcon	popularity	descriere
alsa-utils	V:333, I:470	2702	utilities for configuring and using ALSA
oss-compat	V:0, I:11	18	OSS compatibility under ALSA preventing "/dev/dsp not found" errors
pipewire	V:310, I:367	142	audio and video processing engine multimedia server - metapackage
pipewire-bin	V:319, I:367	2106	audio and video processing engine multimedia server - audio server and CLI programs
pipewire-alsa	V:164, I:230	197	audio and video processing engine multimedia server - audio server to replace ALSA
pipewire-pulse	V:273, I:330	64	audio and video processing engine multimedia server - audio server to replace PulseAudio
pulseaudio	V:174, I:205	6609	PulseAudio server
libpulse0	V:436, I:584	973	PulseAudio client library
jackd	V:2, I:16	8	JACK Audio Connection Kit. (JACK) server (low latency)
libjack0	V:2, I:9	329	JACK Audio Connection Kit. (JACK) library (low latency)
libgststreamer1.0-0	V:464, I:602	5183	GStreamer: GNOME sound engine
libphonon4qt5-4	V:33, I:73	572	Phonon: KDE sound engine

Tabela 9.15: List of sound packages

9.5.8 Disabling the screen saver

For disabling the screen saver, use following commands.

mediu	comanda
The Linux console	setterm -powersave off
The X Window (turning off screensaver)	xset s off
The X Window (disabling dpms)	xset -dpms
The X Window (GUI configuration of screen saver)	xscreensaver-command -prefs

Tabela 9.16: List of commands for disabling the screen saver

9.5.9 Disabling beep sounds

One can always unplug the PC speaker to disable beep sounds. Removing pcspkr kernel module does this for you. The following prevents the readline(3) program used by bash(1) to beep when encountering an alert character (ASCII=7).

```
$ echo "set bell-style none">> ~/.inputrc
```

9.5.10 Utilizare memorie

There are 2 resources available for you to get the memory usage situation.

- The kernel boot message in the `/var/log/dmesg` contains the total exact size of available memory.
- `free(1)` and `top(1)` display information on memory resources on the running system.

Here is an example.

```
# grep '\] Memory' /var/log/dmesg
[ 0.004000] Memory: 990528k/1016784k available (1975k kernel code, 25868k reserved, 931k ↵
data, 296k init)
$ free -k
              total        used        free      shared    buffers     cached
Mem:          997184       976928        20256           0        129592        171932
-/+ buffers/cache:        675404        321780
Swap:         4545576           4        4545572
```

You may be wondering "dmesg tells me a free of 990 MB, and free -k says 320 MB is free. More than 600 MB missing ...".

Do not worry about the large size of "used" and the small size of "free" in the "Mem:" line, but read the one under them (675404 and 321780 in the example above) and relax.

For my MacBook with 1GB=1048576k DRAM (video system steals some of this), I see the following.

raport	dimensiune
Total size in dmesg	1016784k = 1Go - 31792k
Free in dmesg	990528k
Total under shell	997184k
Free under shell	20256k (but effectively 321780k)

Tabela 9.17: List of memory sizes reported

9.5.11 System security and integrity check

Poor system maintenance may expose your system to external exploitation.

For system security and integrity check, you should start with the following.

- The `debsums` package, see `debsums(1)` and Secțiune 2.5.2.
- The `chkrootkit` package, see `chkrootkit(1)`.
- The `clamav` package family, see `clamscan(1)` and `freshclam(1)`.
- [Debian security FAQ](#).
- [Securing Debian Manual](#).

Here is a simple script to check for typical world writable incorrect file permissions.

```
# find / -perm 777 -a \! -type s -a \! -type l -a \! \! -type d -a -perm 1777 \)
```



Atenție

Since the `debsums` package uses [MD5](#) checksums stored locally, it can not be fully trusted as the system security audit tool against malicious attacks.

pachet	popcon(popularitete)	dimensiune	descriere
logcheck	V:5, I:6	120	daemon to mail anomalies in the system logfiles to the administrator
debsums	V:4, I:31	107	utility to verify installed package files against MD5 checksums
chkrootkit	V:9, I:15	966	rootkit detector
clamav	V:8, I:40	33105	anti-virus utility for Unix - command-line interface
tiger	V:1, I:1	7800	report system security vulnerabilities
tripwire	V:1, I:1	5050	file and directory integrity checker
john	V:1, I:8	469	active password cracking tool
aide	V:1, I:2	331	Advanced Intrusion Detection Environment - static binary
integrit	V:0, I:0	2939	file integrity verification program
crack	V:0, I:0	153	password guessing program

Tabela 9.18: List of tools for system security and integrity check

9.6 Data storage tips

Booting your system with Linux [live CDs](#) or [debian-installer CDs](#) in rescue mode makes it easy for you to reconfigure data storage on your boot device.

You may need to `umount(8)` some devices manually from the command line before operating on them if they are automatically mounted by the GUI desktop system.

9.6.1 Disk space usage

The disk space usage can be evaluated by programs provided by the `mount`, `coreutils`, and `xdu` packages:

- `mount(8)` reports all mounted filesystems (= disks).
- `df(1)` reports the disk space usage for the file system.
- `du(1)` reports the disk space usage for the directory tree.

Indicație

You can feed the output of `du(8)` to `xdu(1x)` to produce its graphical and interactive presentation with "`du -k . | xdu`", "`sudo du -k -x / | xdu`", etc.

9.6.2 Disk partition configuration

For [disk partition](#) configuration, although `fdisk(8)` has been considered standard, `parted(8)` deserves some attention. "Disk partitioning data", "partition table", "partition map", and "disk label" are all synonyms.

Older PCs use the classic [Master Boot Record \(MBR\)](#) scheme to hold [disk partitioning](#) data in the first sector, i.e., [LBA](#) sector 0 (512 bytes).

Recent PCs with [Unified Extensible Firmware Interface \(UEFI\)](#), including Intel-based Macs, use [GUID Partition Table \(GPT\)](#) scheme to hold [disk partitioning](#) data not in the first sector.

Although `fdisk(8)` has been standard for the disk partitioning tool, `parted(8)` is replacing it.



Atenție

Although `parted(8)` claims to create and to resize filesystem too, it is safer to do such things using best maintained specialized tools such as `mkfs(8)` (`mkfs.msdos(8)`, `mkfs.ext2(8)`, `mkfs.ext3(8)`, `mkfs.ext4(8)`, ...) and `resize2fs(8)`.

pachet	popcon(popularitate)	descriere
util-linux	V:900, I:999 4401	miscellaneous system utilities including fdisk(8) and cfdisk(8)
parted	V:443, I:577 126	GNU Parted disk partition resizing program
gparted	V:13, I:94 2168	GNOME partition editor based on libparted
gdisk	V:20, I:331 940	partition editor for the GPT/MBR hybrid disk
kpartx	V:17, I:29 78	program to create device mappings for partitions

Tabela 9.19: List of disk partition management packages

Notă

In order to switch between [GPT](#) and [MBR](#), you need to erase first few blocks of disk contents directly (see Secțiune 9.8.6) and use `"parted /dev/sdx mklabel gpt"` or `"parted /dev/sdx mklabel msdos"` to set it. Please note "msdos" is use here for [MBR](#).

9.6.3 Accessing partition using UUID

Although reconfiguration of your partition or activation order of removable storage media may yield different names for partitions, you can access them consistently. This is also helpful if you have multiple disks and your BIOS/UEFI doesn't give them consistent device names.

- `mount(8)` with `"-U"` option can mount a block device using [UUID](#), instead of using its file name such as `"/dev/sda3"`.
- `"/etc/fstab"` (see [fstab\(5\)](#)) can use [UUID](#).
- Boot loaders (Secțiune 3.1.2) may use [UUID](#) too.

Indicație

You can probe [UUID](#) of a block special device with `blkid(8)`.
You can also probe UUID and other information with `"lsblk -f"`.

9.6.4 LVM2

LVM2 is a [logical volume manager](#) for the Linux kernel. With LVM2, disk partitions can be created on logical volumes instead of the physical harddisks.

LVM requires the following.

- device-mapper support in the Linux kernel (default for Debian kernels)
- the userspace device-mapper support library (`libdevmapper*` package)
- the userspace LVM2 tools (`lvm2` package)

Please start learning LVM2 from the following manpages.

- `lvm(8)`: noțiuni de bază despre mecanismul LVM2 (lista tuturor comenzilor LVM2)
- `lvm.conf(5)`: fișierul de configurare pentru LVM2
- `lvs(8)`: raportează informații despre volumele logice
- `vgs(8)`: raportează informații despre grupurile de volume
- `pvs(8)`: raportează informații despre volumele fizice

9.6.5 Filesystem configuration

For [ext4](#) filesystem, the `e2fsprogs` package provides the following.

- `mkfs.ext4(8)` to create new [ext4](#) filesystem
- `fsck.ext4(8)` to check and to repair existing [ext4](#) filesystem
- `tune2fs(8)` to configure superblock of [ext4](#) filesystem
- `debugfs(8)` to debug [ext4](#) filesystem interactively. (It has `undeL` command to recover deleted files.)

The `mkfs(8)` and `fsck(8)` commands are provided by the `e2fsprogs` package as front-ends to various filesystem dependent programs (`mkfs.fstype` and `fsck.fstype`). For [ext4](#) filesystem, they are `mkfs.ext4(8)` and `fsck.ext4(8)` (they are symlinked to `mke2fs(8)` and `e2fsck(8)`).

Similar commands are available for each filesystem supported by Linux.

pachet	popcon	popularity	descriere
e2fsprogs	V:797, I:998	1549	utilities for the ext2/ext3/ext4 filesystems
btrfs-progs	V:44, I:77	5204	utilities for the Btrfs filesystem
reiserfsprogs	V:6, I:22	473	utilities for the Reiserfs filesystem
zfsutils-linux	V:32, I:32	1873	utilities for the OpenZFS filesystem
dosfstools	V:238, I:568	310	utilities for the FAT filesystem. (Microsoft: MS-DOS, Windows)
exfatprogs	V:34, I:459	352	utilities for the exFAT filesystem maintained by Samsung.
exfat-fuse	V:2, I:54	74	read/write exFAT filesystem (Microsoft) driver for FUSE.
xfsprogs	V:33, I:90	4384	utilities for the XFS filesystem. (SGI: IRIX)
ntfs-3g	V:122, I:523	1494	read/write NTFS filesystem (Microsoft: Windows NT, ...) driver for FUSE.
jfsutils	V:0, I:7	1104	utilities for the JFS filesystem. (IBM: AIX, OS/2)
reiser4progs	V:0, I:1	1367	utilities for the Reiser4 filesystem
hfsprogs	V:0, I:3	394	utilities for HFS and HFS Plus filesystem. (Apple: Mac OS)
zerofree	V:5, I:122	30	program to zero free blocks from ext2/3/4 filesystems

Tabela 9.20: List of filesystem management packages

Indicație

[Ext4](#) filesystem is the default filesystem for the Linux system and strongly recommended to use it unless you have some specific reasons not to.

[Btrfs](#) status can be found at [Debian wiki on btrfs](#) and [kernel.org wiki on btrfs](#). It is expected to be the next default filesystem after the [ext4](#) filesystem.

Some tools allow access to filesystem without Linux kernel support (see Secțiune [9.8.2](#)).

9.6.6 Filesystem creation and integrity check

The `mkfs(8)` command creates the filesystem on a Linux system. The `fsck(8)` command provides the filesystem integrity check and repair on a Linux system.

Debian now defaults to no periodic `fsck` after filesystem creation.



Atenție

It is generally not safe to run `fsck` on **mounted filesystems**.

Indicație

You can run the `fsck(8)` command safely on all filesystems including root filesystem on reboot by setting "enable_periodic_fsck" in `/etc/mke2fs.conf` and the max mount count to 0 using `tune2fs -c0 /dev/partition_name`. See `mke2fs.conf(5)` and `tune2fs(8)`.

Check files in `/var/log/fsck/` for the result of the `fsck(8)` command run from the boot script.

9.6.7 Optimization of filesystem by mount options

The basic static filesystem configuration is given by `/etc/fstab`. For example,

«file system»	«mount point»	«type»	«options»	«dump»	«pass»
proc	/proc	proc	defaults	0	0
UUID=709cbe4c-80c1-56db-8ab1-dbce3146d2f7	/	ext4	errors=remount-ro	0	1
UUID=817bae6b-45d2-5aca-4d2a-1267ab46ac23		none	swap sw	0	0
/dev/scd0	/media/cdrom0	udf,iso9660	user,noauto	0	0

Indicație

UUID (see Secțiune 9.6.3) may be used to identify a block device instead of normal block device names such as `/dev/sda1`, `/dev/sda2`, ...

Since Linux 2.6.30, the kernel defaults to the behavior provided by "relatime" option.

See `fstab(5)` and `mount(8)`.

9.6.8 Optimization of filesystem via superblock

Characteristics of a filesystem can be optimized via its superblock using the `tune2fs(8)` command.

- Execution of `sudo tune2fs -l /dev/hda1` displays the contents of the filesystem superblock on `/dev/hda1`.
- Execution of `sudo tune2fs -c 50 /dev/hda1` changes frequency of filesystem checks (`fsck` execution during boot-up) to every 50 boots on `/dev/hda1`.
- Execution of `sudo tune2fs -j /dev/hda1` adds journaling capability to the filesystem, i.e. filesystem conversion from `ext2` to `ext3` on `/dev/hda1`. (Do this on the unmounted filesystem.)
- Execution of `sudo tune2fs -O extents,uninit_bg,dir_index /dev/hda1 && fsck -pf /dev/hda1` converts it from `ext3` to `ext4` on `/dev/hda1`. (Do this on the unmounted filesystem.)

Indicație

Despite its name, `tune2fs(8)` works not only on the `ext2` filesystem but also on the `ext3` and `ext4` filesystems.

9.6.9 Optimizarea discului dur

**Avertisment**

Please check your hardware and read manpage of `hdparm(8)` before playing with hard disk configuration because this may be quite dangerous for the data integrity.

You can test disk access speed of a hard disk, e.g. `/dev/hda`, by `hdparm -tT /dev/hda`. For some hard disk connected with (E)IDE, you can speed it up with `hdparm -q -c3 -d1 -u1 -m16 /dev/hda` by enabling the "(E)IDE 32-bit I/O support", enabling the "using_dma flag", setting "interrupt-unmask flag", and setting the "multiple 16 sector I/O" (dangerous!).

You can test write cache feature of a hard disk, e.g. `/dev/sda`, by `hdparm -W /dev/sda`. You can disable its write cache feature with `hdparm -W 0 /dev/sda`.

You may be able to read badly pressed CDROMs on modern high speed CD-ROM drive by slowing it down with `setcd -x 2`.

9.6.10 Optimization of solid state drive

[Solid state drive \(SSD\)](#) is auto detected now.

Reduce unnecessary disk accesses to prevent disk wear out by mounting `tmpfs` on volatile data path in `/etc/fstab`.

9.6.11 Using SMART to predict hard disk failure

You can monitor and log your hard disk which is compliant to [SMART](#) with the `smartd(8)` daemon.

1. Enable [SMART](#) feature in [BIOS](#).
2. Install the `smartmontools` package.
3. Identify your hard disk drives by listing them with `df(1)`.
 - Let's assume a hard disk drive to be monitored as `/dev/hda`.
4. Check the output of `smartctl -a /dev/hda` to see if [SMART](#) feature is actually enabled.
 - If not, enable it by `smartctl -s on -a /dev/hda`.
5. Enable `smartd(8)` daemon to run by the following.
 - uncomment `start_smartd=yes` in the `/etc/default/smartmontools` file.
 - restart the `smartd(8)` daemon by `sudo systemctl restart smartmontools`.

Indicație

The `smartd(8)` daemon can be customized with the `/etc/smartd.conf` file including how to be notified of warnings.

9.6.12 Specify temporary storage directory via \$TMPDIR

Applications create temporary files normally under the temporary storage directory `/tmp`. If `/tmp` does not provide enough space, you can specify such temporary storage directory via the `$TMPDIR` variable for well-behaving programs.

9.6.13 Expansion of usable storage space via LVM

For partitions created on [Logical Volume Manager \(LVM\)](#) (Linux feature) at install time, they can be resized easily by concatenating extents onto them or truncating extents from them over multiple storage devices without major system reconfiguration.

9.6.14 Expansion of usable storage space by mounting another partition

If you have an empty partition (e.g., `/dev/sdx`), you can format it with `mkfs.ext4(1)` and `mount(8)` it to a directory where you need more space. (You need to copy original data contents.)

```
$ sudo mv work-dir old-dir
$ sudo mkfs.ext4 /dev/sdx
$ sudo mount -t ext4 /dev/sdx work-dir
$ sudo cp -a old-dir/* work-dir
$ sudo rm -rf old-dir
```

Indicație

You may alternatively mount an empty disk image file (see Secțiune 9.7.5) as a loop device (see Secțiune 9.7.3). The actual disk usage grows with the actual data stored.

9.6.15 Expansion of usable storage space by bind-mounting another directory

If you have an empty directory (e.g., `/path/to/emp-dir`) on another partition with usable space, you can `mount(8)` it with `--bind` option to a directory (e.g., `work-dir`) where you need more space.

```
$ sudo mount --bind /path/to/emp-dir work-dir
```

9.6.16 Expansion of usable storage space by overlay-mounting another directory

If you have usable space in another partition (e.g., `/path/to/empty` and `/path/to/work`), you can create a directory in it and stack that on to an old directory (e.g., `/path/to/old`) where you need space using the [OverlayFS](#) for Linux kernel 3.18 or newer (Debian Stretch 9.0 or newer).

```
$ sudo mount -t overlay overlay \
  -olowerdir=/path/to/old-dir,upperdir=/path/to/empty,workdir=/path/to/work
```

Here, `/path/to/empty` and `/path/to/work` should be on the RW-enabled partition to write on `/path/to/old`.

9.6.17 Expansion of usable storage space using symlink



Atenție

This is a deprecated method. Some software may not function well with "symlink to a directory". Instead, use the "mounting" approaches described in the above.

If you have an empty directory (e.g., `/path/to/emp-dir`) in another partition with usable space, you can create a symlink to the directory with `ln(8)`.

```
$ sudo mv work-dir old-dir
$ sudo mkdir -p /path/to/emp-dir
$ sudo ln -sf /path/to/emp-dir work-dir
$ sudo cp -a old-dir/* work-dir
$ sudo rm -rf old-dir
```

**Avertisment**

Do not use "symlink to a directory" for directories managed by the system such as `/opt`. Such a symlink may be overwritten when the system is upgraded.

9.7 The disk image

Here, we discuss manipulations of the disk image.

9.7.1 Making the disk image file

The disk image file, `disk.img`, of an unmounted device, e.g., the second SCSI or serial ATA drive `/dev/sdb`, can be made using `cp(1)` or `dd(1)` by the following.

```
# cp /dev/sdb disk.img
# dd if=/dev/sdb of=disk.img
```

The disk image of the traditional PC's [master boot record \(MBR\)](#) (see Secțiune 9.6.2) which reside on the first sector on the primary IDE disk can be made by using `dd(1)` by the following.

```
# dd if=/dev/hda of=mbr.img bs=512 count=1
# dd if=/dev/hda of=mbr-nopart.img bs=446 count=1
# dd if=/dev/hda of=mbr-part.img skip=446 bs=1 count=66
```

- `mbr.img`: The MBR with the partition table
- `mbr-nopart.img`: The MBR without the partition table
- `mbr-part.img`: The partition table of the MBR only

If you have an SCSI or serial ATA device as the boot disk, substitute `/dev/hda` with `/dev/sda`.

If you are making an image of a disk partition of the original disk, substitute `/dev/hda` with `/dev/hda1` etc.

9.7.2 Writing directly to the disk

The disk image file, `disk.img` can be written to an unmounted device, e.g., the second SCSI drive `/dev/sdb` with matching size, by the following.

```
# dd if=disk.img of=/dev/sdb
```

Similarly, the disk partition image file, `partition.img` can be written to an unmounted partition, e.g., the first partition of the second SCSI drive `/dev/sdb1` with matching size, by the following.

```
# dd if=partition.img of=/dev/sdb1
```

9.7.3 Mounting the disk image file

The disk image "partition.img" containing a single partition image can be mounted and unmounted by using the [loop device](#) as follows.

```
# losetup --show -f partition.img
/dev/loop0
# mkdir -p /mnt/loop0
# mount -t auto /dev/loop0 /mnt/loop0
...hack...hack...hack
# umount /dev/loop0
# losetup -d /dev/loop0
```

This can be simplified as follows.

```
# mkdir -p /mnt/loop0
# mount -t auto -o loop partition.img /mnt/loop0
...hack...hack...hack
# umount partition.img
```

Each partition of the disk image "disk.img" containing multiple partitions can be mounted by using the [loop device](#).

```
# losetup --show -f -P disk.img
/dev/loop0
# ls -l /dev/loop0*
brw-rw---- 1 root disk  7,  0 Apr  2 22:51 /dev/loop0
brw-rw---- 1 root disk 259, 12 Apr  2 22:51 /dev/loop0p1
brw-rw---- 1 root disk 259, 13 Apr  2 22:51 /dev/loop0p14
brw-rw---- 1 root disk 259, 14 Apr  2 22:51 /dev/loop0p15
# fdisk -l /dev/loop0
Disk /dev/loop0: 2 GiB, 2147483648 bytes, 4194304 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 6A1D9E28-C48C-2144-91F7-968B3CBC9BD1

Device            Start      End Sectors  Size Type
/dev/loop0p1      262144    4192255 3930112   1.9G Linux root (x86-64)
/dev/loop0p14      2048       8191    6144     3M BIOS boot
/dev/loop0p15      8192     262143   253952   124M EFI System

Partition table entries are not in disk order.
# mkdir -p /mnt/loop0p1
# mkdir -p /mnt/loop0p15
# mount -t auto /dev/loop0p1 /mnt/loop0p1
# mount -t auto /dev/loop0p15 /mnt/loop0p15
# mount |grep loop
/dev/loop0p1 on /mnt/loop0p1 type ext4 (rw,relatime)
/dev/loop0p15 on /mnt/loop0p15 type vfat (rw,relatime,fmask=0002,dmask=0002,allow_utime ↵
=0020,codepage=437,iocharset=ascii,shortname=mixed,utf8,errors=remount-ro)
...hack...hack...hack
# umount /dev/loop0p1
# umount /dev/loop0p15
# losetup -d /dev/loop0
```

Alternatively, similar effects can be done by using the [device mapper](#) devices created by `kpartx(8)` from the `kpartx` package as follows.

```
# kpartx -a -v disk.img
add map loop0p1 (253:0): 0 3930112 linear 7:0 262144
add map loop0p14 (253:1): 0 6144 linear 7:0 2048
```

```

add map loop0p15 (253:2): 0 253952 linear 7:0 8192
# fdisk -l /dev/loop0
Disk /dev/loop0: 2 GiB, 2147483648 bytes, 4194304 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 6A1D9E28-C48C-2144-91F7-968B3CBC9BD1

Device          Start      End Sectors  Size Type
/dev/loop0p1    262144    4192255 3930112   1.9G Linux root (x86-64)
/dev/loop0p14     2048       8191    6144     3M BIOS boot
/dev/loop0p15     8192    262143   253952   124M EFI System

Partition table entries are not in disk order.
# ls -l /dev/mapper/
total 0
crw----- 1 root root 10, 236 Apr  2 22:45 control
lrwxrwxrwx 1 root root      7 Apr  2 23:19 loop0p1 -> ../dm-0
lrwxrwxrwx 1 root root      7 Apr  2 23:19 loop0p14 -> ../dm-1
lrwxrwxrwx 1 root root      7 Apr  2 23:19 loop0p15 -> ../dm-2
# mkdir -p /mnt/loop0p1
# mkdir -p /mnt/loop0p15
# mount -t auto /dev/mapper/loop0p1 /mnt/loop0p1
# mount -t auto /dev/mapper/loop0p15 /mnt/loop0p15
# mount |grep loop
/dev/loop0p1 on /mnt/loop0p1 type ext4 (rw,relatime)
/dev/loop0p15 on /mnt/loop0p15 type vfat (rw,relatime,fmask=0002,dmask=0002,allow_utime ←
    =0020,codepage=437,iocharset=ascii,shortname=mixed,utf8,errors=remount-ro)
...hack...hack...hack
# umount /dev/mapper/loop0p1
# umount /dev/mapper/loop0p15
# kpartx -d disk.img

```

9.7.4 Cleaning a disk image file

A disk image file, "disk.img" can be cleaned of all removed files into clean sparse image "new.img" by the following.

```

# mkdir old; mkdir new
# mount -t auto -o loop disk.img old
# dd bs=1 count=0 if=/dev/zero of=new.img seek=5G
# mount -t auto -o loop new.img new
# cd old
# cp -a --sparse=always ./ ../new/
# cd ..
# umount new.img
# umount disk.img

```

If "disk.img" is in ext2, ext3 or ext4, you can also use zerofree(8) from the zerofree package as follows.

```

# losetup --show -f disk.img
/dev/loop0
# zerofree /dev/loop0
# cp --sparse=always disk.img new.img
# losetup -d /dev/loop0

```

9.7.5 Making the empty disk image file

The empty disk image "disk.img" which can grow up to 5GiB can be made using dd(1) as follows.

```
$ dd bs=1 count=0 if=/dev/zero of=disk.img seek=5G
```

Instead of using `dd(1)`, specialized `fallocate(8)` may be used here.

You can create an ext4 filesystem on this disk image "disk.img" using the [loop device](#) as follows.

```
# losetup --show -f disk.img
/dev/loop0
# mkfs.ext4 /dev/loop0
...hack...hack...hack
# losetup -d /dev/loop0
$ du --apparent-size -h disk.img
5.0G disk.img
$ du -h disk.img
83M disk.img
```

For "disk.img", its file size is 5.0 GiB and its actual disk usage is mere 83MiB. This discrepancy is possible since [ext4](#) can hold [sparse file](#).

Indicație

The actual disk usage of [sparse file](#) grows with data which are written to it.

Using similar operation on devices created by the [loop device](#) or the [device mapper](#) devices as Secțiune 9.7.3, you can partition this disk image "disk.img" using `parted(8)` or `fdisk(8)`, and can create filesystem on it using `mkfs.ext4(8)`, `mkswap(8)`, etc.

9.7.6 Making the ISO9660 image file

The [ISO9660](#) image file, "cd.iso", from the source directory tree at "source_directory" can be made using `genisoimage(1)` provided by [cdrkit](#) by the following.

```
# genisoimage -r -J -T -V volume_id -o cd.iso source_directory
```

Similarly, the bootable ISO9660 image file, "cdboot.iso", can be made from debian-installer like directory tree at "source_directory" by the following.

```
# genisoimage -r -o cdboot.iso -V volume_id \
  -b isolinux/isolinux.bin -c isolinux/boot.cat \
  -no-emul-boot -boot-load-size 4 -boot-info-table source_directory
```

Here [Isolinux boot loader](#) (see Secțiune 3.1.2) is used for booting.

You can calculate the md5sum value and make the ISO9660 image directly from the CD-ROM device as follows.

```
$ isoinfo -d -i /dev/cdrom
CD-ROM is in ISO 9660 format
...
Logical block size is: 2048
Volume size is: 23150592
...
# dd if=/dev/cdrom bs=2048 count=23150592 conv=notrunc,noerror | md5sum
# dd if=/dev/cdrom bs=2048 count=23150592 conv=notrunc,noerror > cd.iso
```



Avertisment

You must carefully avoid ISO9660 filesystem read ahead bug of Linux as above to get the right result.

9.7.7 Writing directly to the CD/DVD-R/RW

Indicație

DVD is only a large CD to `wodim(1)` provided by [cdrkit](#).

You can find a usable device by the following.

```
# wodim --devices
```

Then the blank CD-R is inserted to the CD drive, and the ISO9660 image file, "cd.iso" is written to this device, e.g., "/dev/hda", using `wodim(1)` by the following.

```
# wodim -v -eject dev=/dev/hda cd.iso
```

If CD-RW is used instead of CD-R, do this instead by the following.

```
# wodim -v -eject blank=fast dev=/dev/hda cd.iso
```

Indicație

If your desktop system mounts CDs automatically, unmount it by "sudo umount /dev/hda" from console before using `wodim(1)`.

9.7.8 Mounting the ISO9660 image file

If "cd.iso" contains an ISO9660 image, then the following manually mounts it to "/cdrom".

```
# mount -t iso9660 -o ro,loop cd.iso /cdrom
```

Indicație

Modern desktop system may mount removable media such as ISO9660 formatted CD automatically (see Secțiune [10.1.7](#)).

9.8 The binary data

Here, we discuss direct manipulations of the binary data on storage media.

9.8.1 Viewing and editing binary data

The most basic viewing method of binary data is to use "od -t x1" command.

Indicație

HEX is used as an acronym for [hexadecimal](#) format with [radix](#) 16. OCTAL is for [octal](#) format with [radix](#) 8. ASCII is for [American Standard Code for Information Interchange](#), i.e., normal English text code. EBCDIC is for [Extended Binary Coded Decimal Interchange Code](#) used on [IBM mainframe](#) operating systems.

pachet	popcon(popularitate)	descriere
coreutils	V:897, I:999 18457	basic package which has od(1) to dump files (HEX, ASCII, OCTAL, ...)
bsdmainutils	V:4, I:163 17	utility package which has hd(1) to dump files (HEX, ASCII, OCTAL, ...)
hexedit	V:0, I:8 70	binary editor and viewer (HEX, ASCII)
bless	V:0, I:1 924	full featured hexadecimal editor (GNOME)
okteta	V:1, I:13 1590	full featured hexadecimal editor (KDE4)
ncurses-hexedit	V:0, I:1 130	binary editor and viewer (HEX, ASCII, EBCDIC)
beav	V:0, I:0 137	binary editor and viewer (HEX, ASCII, EBCDIC, OCTAL, ...)

Tabela 9.21: List of packages which view and edit binary data

pachet	popcon(popularitate)	descriere
mtools	V:7, I:56 400	utilities for MSDOS files without mounting them
hfsutils	V:0, I:3 178	utilities for HFS and HFS+ files without mounting them

Tabela 9.22: List of packages to manipulate files without mounting disk

9.8.2 Manipulating files without mounting disk

There are tools to read and write files without mounting disk.

9.8.3 Data redundancy

Software [RAID](#) systems offered by the Linux kernel provide data redundancy in the kernel filesystem level to achieve high levels of storage reliability.

There are tools to add data redundancy to files in application program level to achieve high levels of storage reliability, too.

pachet	popcon(popularitate)	descriere
par2	V:11, I:119 292	Parity Archive Volume Set, for checking and repair of files
dvdisaster	V:0, I:1 1422	data loss/scratch/aging protection for CD/DVD media
dvbackup	V:0, I:0 413	backup tool using MiniDV camcorders (providing rsync(1))

Tabela 9.23: List of tools to add data redundancy to files

9.8.4 Data file recovery and forensic analysis

There are tools for data file recovery and forensic analysis.

Indicație

You can undelete files on the ext2 filesystem using `list_deleted_inodes` and `unde1` commands of `debugfs(8)` in the `e2fsprogs` package.

pachet	popcon(popularitate)	downloads	descriere
testdisk	V:2, I:27	1495	utilities for partition scan and disk recovery
magicrescue	V:0, I:2	258	utility to recover files by looking for magic bytes
scalpel	V:0, I:2	89	frugal, high performance file carver
myrescue	V:0, I:2	83	rescue data from damaged harddisks
extundelete	V:0, I:8	152	utility to undelete files on the ext3/4 filesystem
ext4magic	V:0, I:3	235	utility to undelete files on the ext3/4 filesystem
ext3grep	V:0, I:2	299	tool to help recover deleted files on the ext3 filesystem
scrounge-ntfs	V:0, I:1	49	data recovery program for NTFS filesystems
gzrt	V:0, I:0	33	gzip recovery toolkit
sleuthkit	V:2, I:24	1719	tools for forensics analysis. (Sleuthkit)
autopsy	V:0, I:1	1026	graphical interface to SleuthKit
foremost	V:0, I:4	102	forensics application to recover data
guymager	V:0, I:0	1047	forensic imaging tool based on Qt
dcfldd	V:0, I:3	113	enhanced version of dd for forensics and security

Tabela 9.24: List of packages for data file recovery and forensic analysis

9.8.5 Splitting a large file into small files

When a data is too big to backup as a single file, you can backup its content after splitting it into, e.g. 2000MiB chunks and merge those chunks back into the original file later.

```
$ split -b 2000m large_file
$ cat x* >large_file
```



Atenție

Please make sure you do not have any files starting with "x" to avoid name crashes.

9.8.6 Clearing file contents

In order to clear the contents of a file such as a log file, do not use `rm(1)` to delete the file and then create a new empty file, because the file may still be accessed in the interval between commands. The following is the safe way to clear the contents of the file.

```
$ :>file_to_be_cleared
```

9.8.7 Dummy files

The following commands create dummy or empty files.

```
$ dd if=/dev/zero of=5kb.file bs=1k count=5
$ dd if=/dev/urandom of=7mb.file bs=1M count=7
$ touch zero.file
$ : > alwayszero.file
```

You should find following files.

- „5kb.file” ce conține 5 Ko de zerouri.

- „7mb.file” ce conține 7 Mo de date aleatorii.
- „zero.file” may be a 0 byte file. If it existed, its mtime is updated while its content and its length are kept.
- „alwayszero.file” is always a 0 byte file. If it existed, its mtime is updated and its content is reset.

9.8.8 Erasing an entire hard disk

There are several ways to completely erase data from an entire hard disk like device, e.g., USB memory stick at `"/dev/sda"`.



Atenție

Check your USB memory stick location with `mount(8)` first before executing commands here. The device pointed by `"/dev/sda"` may be SCSI hard disk or serial-ATA hard disk where your entire system resides.

Erase all the disk content by resetting data to 0 with the following.

```
# dd if=/dev/zero of=/dev/sda
```

Erase everything by overwriting with random data as follows.

```
# dd if=/dev/urandom of=/dev/sda
```

Erase everything by overwriting with random data very efficiently as follows.

```
# shred -v -n 1 /dev/sda
```

You may alternatively use `badblocks(8)` with `-t random` option.

Since `dd(1)` is available from the shell of many bootable Linux CDs such as Debian installer CD, you can erase your installed system completely by running an erase command from such media on the system hard disk, e.g., `"/dev/hda"`, `"/dev/sda"`, etc.

9.8.9 Erasing unused area of an hard disk

Unused area on an hard disk (or USB memory stick), e.g. `"/dev/sdb1"` may still contain erased data themselves since they are only unlinked from the filesystem. These can be cleaned by overwriting them.

```
# mount -t auto /dev/sdb1 /mnt/foo
# cd /mnt/foo
# dd if=/dev/zero of=junk
dd: writing to `junk': No space left on device
...
# sync
# umount /dev/sdb1
```



Avertisment

This is usually good enough for your USB memory stick. But this is not perfect. Most parts of erased filenames and their attributes may be hidden and remain in the filesystem.

9.8.10 Undeleting deleted but still open files

Even if you have accidentally deleted a file, as long as that file is still being used by some application (read or write mode), it is possible to recover such a file.

De exemplu, încercați următoarele

```
$ echo foo > bar
$ less bar
$ ps aux | grep ' less[ ]'
bozo 4775 0.0 0.0 92200 884 pts/8 S+ 00:18 0:00 less bar
$ rm bar
$ ls -l /proc/4775/fd | grep bar
lr-x----- 1 bozo bozo 64 2008-05-09 00:19 4 -> /home/bozo/bar (deleted)
$ cat /proc/4775/fd/4 >bar
$ ls -l
-rw-r--r-- 1 bozo bozo 4 2008-05-09 00:25 bar
$ cat bar
foo
```

Execute on another terminal (when you have the `lsuf` package installed) as follows.

```
$ ls -li bar
2228329 -rw-r--r-- 1 bozo bozo 4 2008-05-11 11:02 bar
$ lsuf |grep bar|grep less
less 4775 bozo 4r REG 8,3 4 2228329 /home/bozo/bar
$ rm bar
$ lsuf |grep bar|grep less
less 4775 bozo 4r REG 8,3 4 2228329 /home/bozo/bar (deleted)
$ cat /proc/4775/fd/4 >bar
$ ls -li bar
2228302 -rw-r--r-- 1 bozo bozo 4 2008-05-11 11:05 bar
$ cat bar
foo
```

9.8.11 Searching all hardlinks

Files with hardlinks can be identified by "`ls -li`".

```
$ ls -li
total 0
2738405 -rw-r--r-- 1 root root 0 2008-09-15 20:21 bar
2738404 -rw-r--r-- 2 root root 0 2008-09-15 20:21 baz
2738404 -rw-r--r-- 2 root root 0 2008-09-15 20:21 foo
```

Both "baz" and "foo" have link counts of "2" (>1) showing them to have hardlinks. Their [inode](#) numbers are common "2738404". This means they are the same hardlinked file. If you do not happen to find all hardlinked files by chance, you can search it by the [inode](#), e.g., "2738404" as the following.

```
# find /path/to/mount/point -xdev -inum 2738404
```

9.8.12 Invisible disk space consumption

All deleted but open files consume disk space although they are not visible from normal `du(1)`. They can be listed with their size by the following.

```
# lsuf -s -X / |grep deleted
```

9.9 Data encryption tips

With physical access to your PC, anyone can easily gain root privilege and access all the files on your PC (see Secțiune 4.6.4). This means that login password system can not secure your private and sensitive data against possible theft of your PC. You must deploy data encryption technology to do it. Although [GNU privacy guard](#) (see Secțiune 10.3) can encrypt files, it takes some user efforts.

[Dm-crypt](#) facilitates automatic data encryption via native Linux kernel modules with minimal user efforts using [device-mapper](#).

pachet	popcon	popularity	descriere
cryptsetup	V:16, I:81	463	utilities for encrypted block device (dm-crypt / LUKS)
cryptmount	V:2, I:2	231	utilities for encrypted block device (dm-crypt / LUKS) with focus on mount/unmount by normal users
fscrypt	V:0, I:1	6471	utilities for Linux filesystem encryption (fscrypt)
libpam-fscrypt	V:0, I:0	5589	PAM module for Linux filesystem encryption (fscrypt)

Tabela 9.25: List of data encryption utilities



Atenție

Data encryption costs CPU time etc. Encrypted data becomes inaccessible if its password is lost. Please weigh its benefits and costs.

Notă

Entire Debian system can be installed on a encrypted disk by the [debian-installer](#) (lenny or newer) using [dm-crypt/LUKS](#) and [initramfs](#).

Indicație

See Secțiune 10.3 for user space encryption utility: [GNU Privacy Guard](#).

9.9.1 Removable disk encryption with dm-crypt/LUKS

You can encrypt contents of removable mass devices, e.g. USB memory stick on `/dev/sdx`, using [dm-crypt/LUKS](#). You simply format it as the following.

```
# fdisk /dev/sdx
... "n" "p" "1" "return" "return" "w"
# cryptsetup luksFormat /dev/sdx1
...
# cryptsetup open /dev/sdx1 secret
...
# ls -l /dev/mapper/
total 0
crw-rw---- 1 root root 10, 60 2021-10-04 18:44 control
lrwxrwxrwx 1 root root 7 2021-10-04 23:55 secret -> ../dm-0
# mkfs.vfat /dev/mapper/secret
...
# cryptsetup close secret
```

Then, it can be mounted just like normal one on to `/media/username/disk_label`, except for asking password (see Secțiune 10.1.7) under modern desktop environment using the `udisks2` package. The difference is that every data written to it is encrypted. The password entry may be automated using keyring (see Secțiune 10.3.6).

You may alternatively format media in different filesystem, e.g., `ext4` with `mkfs.ext4 /dev/mapper/sdx1`. If `btrfs` is used instead, the `udisks2-btrfs` package needs to be installed. For these filesystems, the file ownership and permissions may need to be configured.

9.9.2 Mounting encrypted disk with dm-crypt/LUKS

For example, an encrypted disk partition created with `dm-crypt/LUKS` on `/dev/sdc5` by Debian Installer can be mounted onto `/mnt` as follows:

```
$ sudo cryptsetup open /dev/sdc5 ninja --type luks
Enter passphrase for /dev/sdc5: ****
$ sudo lvm
lvm> lvscan
  inactive          '/dev/ninja-vg/root' [13.52 GiB] inherit
  inactive          '/dev/ninja-vg/swap_1' [640.00 MiB] inherit
  ACTIVE            '/dev/goofy/root' [180.00 GiB] inherit
  ACTIVE            '/dev/goofy/swap' [9.70 GiB] inherit
lvm> lvchange -a y /dev/ninja-vg/root
lvm> exit
Exiting.
$ sudo mount /dev/ninja-vg/root /mnt
```

9.10 Nucleul

Debian distributes modularized [Linux kernel](#) as packages for supported architectures.

If you are reading this documentation, you probably don't need to compile Linux kernel by yourself.

9.10.1 Parametrii nucleului

Many Linux features are configurable via kernel parameters as follows.

- Kernel parameters initialized by the bootloader (see Secțiune 3.1.2)
- Kernel parameters changed by `sysctl(8)` at runtime for ones accessible via `sysfs` (see Secțiune 1.2.12)
- Module parameters set by arguments of `modprobe(8)` when a module is activated (see Secțiune 9.7.3)

See ["The Linux kernel user's and administrator's guide » The kernel's command-line parameters"](#) for the detail.

9.10.2 Kernel headers

Most **normal programs** don't need kernel headers and in fact may break if you use them directly for compiling. They should be compiled against the headers in `/usr/include/linux` and `/usr/include/asm` provided by the `libc6-dev` package (created from the `glibc` source package) on the Debian system.

Notă

For compiling some kernel-specific programs such as the kernel modules from the external source and the automounter daemon (`amd`), you must include path to the corresponding kernel headers, e.g. `-I/usr/src/linux-particular-version/include/`, to your command line.

9.10.3 Compiling the kernel and related modules

Debian has its own method of compiling the kernel and related modules.

pachet	popcon(popularitate)	nr. desc.	descriere
build-essential	I:505	17	essential packages for building Debian packages: make, gcc, ...
bzip2	V:165, I:971	113	compress and decompress utilities for bz2 files
libncurses5-dev	I:44	6	developer's libraries and docs for ncurses
git	V:379, I:598	50172	git: distributed revision control system used by the Linux kernel
fakeroot	V:30, I:509	225	provide fakeroot environment for building package as non-root
initramfs-tools	V:422, I:988	52	tool to build an initramfs (Debian specific)
dkms	V:56, I:148	235	dynamic kernel module support (DKMS) (generic)
module-assistant	V:0, I:14	391	helper tool to make module package (Debian specific)
devscripts	V:5, I:34	2768	helper scripts for a Debian Package maintainer (Debian specific)

Tabela 9.26: List of key packages to be installed for the kernel recompilation on the Debian system

If you use `initrd` in Secțiune 3.1.2, make sure to read the related information in `initramfs-tools(8)`, `update-initramfs(8)` and `initramfs.conf(5)`.



Avertisment

Do not put symlinks to the directories in the source tree (e.g. `"/usr/src/linux*"`) from `"/usr/include/linux"` and `"/usr/include/asm"` when compiling the Linux kernel source. (Some outdated documents suggest this.)

Notă

When compiling the latest Linux kernel on the Debian stable system, the use of backported latest tools from the Debian unstable may be needed.

`module-assistant(8)` (or its short form `m-a`) helps users to build and install module package(s) easily for one or more custom kernels.

The [dynamic kernel module support \(DKMS\)](#) is a new distribution independent framework designed to allow individual kernel modules to be upgraded without changing the whole kernel. This is used for the maintenance of out-of-tree modules. This also makes it very easy to rebuild modules as you upgrade kernels.

9.10.4 Compiling the kernel source: Debian Kernel Team recommendation

For building custom kernel binary packages from the upstream kernel source, you should use the `"deb-pkg"` target provided by it.

```
$ sudo apt-get build-dep linux
$ cd /usr/src
$ wget https://mirrors.edge.kernel.org/pub/linux/kernel/v6.x/linux-version.tar.xz
$ tar --xz -xvf linux-version.tar.xz
$ cd linux-version
$ cp /boot/config-version .config
$ make menuconfig
...
$ make deb-pkg
```

Indicație

The `linux-source-version` package provides the Linux kernel source with Debian patches as `"/usr/src/linux-version.tar.bz2"`.

For building specific binary packages from the Debian kernel source package, you should use the `"binary-arch_architecture"` targets in `"debian/rules.gen"`.

```
$ sudo apt-get build-dep linux
$ apt-get source linux
$ cd linux-3.*
$ fakeroot make -f debian/rules.gen binary-arch_i386_none_686
```

See further information:

- Debian Wiki: [KernelFAQ](#)
- Debian Wiki: [DebianKernel](#)
- Debian Linux Kernel Handbook: <https://kernel-handbook.debian.net>

9.10.5 Hardware drivers and firmware

The hardware driver is the code running on the main CPUs of the target system. Most hardware drivers are available as free software now and are included in the normal Debian kernel packages in the main area.

- **GPU driver**
 - Controlor GPU Intel (main)
 - Controlor GPU AMD/ATI (main)
 - Controlor GPU NVIDIA (main pentru controlorul [nouveau](#) și non-free pentru controlorii numai-binari asigurați de producător.)

The firmware is the code or data loaded on the device attach to the target system (e.g., CPU [microcode](#), rendering code running on GPU, or [FPGA](#) / [CPLD](#) data, ...). Some firmware packages are available as free software but many firmware packages are not available as free software since they contain sourceless binary data. Installing these firmware data is essential for the device to function as expected.

- The firmware data packages containing data loaded to the volatile memory on the target device.
 - `firmware-linux-free` (main)
 - `firmware-linux-nonfree` (non-free-firmware)
 - `firmware-linux-*` (non-free-firmware)
 - `*-firmware` (non-free-firmware)
 - `intel-microcode` (non-free-firmware)
 - `amd64-microcode` (non-free-firmware)
- The firmware update program packages which update data on the non-volatile memory on the target device.
 - [fwupd](#) (main): Firmware update daemon which downloads firmware data from [Linux Vendor Firmware Service](#).
 - `gnome-firmware` (main): GTK front end for fwupd
 - `plasma-discover-backend-fwupd` (main): Qt front end for fwupd

Please note that access to non-free-firmware packages are provided by the official installation media to offer functional installation experience to the user since Debian 12 Bookworm. The non-free-firmware area is described in Secțiune [2.1.5](#).

Please also note that the firmware data downloaded by [fwupd](#) from [Linux Vendor Firmware Service](#) and loaded to the running Linux kernel may be non-free.

9.11 Virtualized system

Use of virtualized system enables us to run multiple instances of system simultaneously on a single hardware.

Indicație

See [Debian wiki on SystemVirtualization](#).

9.11.1 Virtualization and emulation tools

There are several [virtualization](#) and emulation tool platforms.

- Complete [hardware emulation](#) packages such as ones installed by the [games-emulator](#) metapackage
- Mostly CPU level emulation with some I/O device emulations such as [QEMU](#)
- Mostly CPU level virtualization with some I/O device emulations such as [Kernel-based Virtual Machine \(KVM\)](#)
- OS level container virtualization with the kernel level support such as [LXC \(Linux Containers\)](#), [Docker](#), `systemd-nspawn(1)` ...
- OS level filesystem access virtualization with the system library call override on the file path such as [chroot](#)
- OS level filesystem access virtualization with the system library call override on the file ownership such as [fakeroot](#)
- OS API emulation such as [Wine](#)
- Interpreter level virtualization with its executable selection and run-time library overrides such as [virtualenv](#) and [venv](#) for Python

The container virtualization uses Secțiune [4.7.5](#) and is the backend technology of Secțiune [7.7](#).

Here are some packages to help you to setup the virtualized system.

See Wikipedia article [Comparison of platform virtual machines](#) for detail comparison of different platform virtualization solutions.

9.11.2 Virtualization work flow

Notă

Default Debian kernels support [KVM](#) since Lenny.

Typical work flow for [virtualization](#) involves several steps.

- Create an empty filesystem (a file tree or a disk image).
 - The file tree can be created by `mkdir -p /path/to/chroot`.
 - The raw disk image file can be created with `dd(1)` (see Secțiune [9.7.1](#) and Secțiune [9.7.5](#)).
 - `qemu-img(1)` can be used to create and convert disk image files supported by [QEMU](#).
 - The raw and [VMDK](#) file format can be used as common format among virtualization tools.
 - Mount the disk image with `mount(8)` to the filesystem (optional).
 - For the raw disk image file, mount it as [loop device](#) or [device mapper](#) devices (see Secțiune [9.7.3](#)).
-

pachet	popcon(popularitate)	dimensiune	descriere
coreutils	V:897, I:999	18457	GNU core utilities which contain chroot(8)
systemd-container	V:72, I:76	2264	systemd container/nspawn tools which contain systemd-nspawn(1)
schroot	V:5, I:7	2627	specialized tool for executing Debian binary packages in chroot
sbuild	V:1, I:4	157	tool for building Debian binary packages from Debian sources
debootstrap	V:4, I:48	330	bootstrap a basic Debian system (written in sh)
cdebootstrap	V:0, I:1	114	bootstrap a Debian system (written in C)
cloud-image-utils	V:1, I:15	66	cloud image management utilities
cloud-guest-utils	V:2, I:18	71	cloud guest utilities
virt-manager	V:11, I:49	2310	Virtual Machine Manager : desktop application for managing virtual machines
libvirt-clients	V:48, I:72	1154	programs for the libvirt library
incus	V:0, I:2	21	Incus : system container and virtual machine manager (for Debian 13 "Trixie")
lxd	V:0, I:0	55606	LXD : system container and virtual machine manager (for Debian 12 "Bookworm")
podman	V:24, I:27	81828	podman : engine to run OCI-based containers in Pods
podman-docker	V:2, I:2	275	engine to run OCI-based containers in Pods - wrapper for docker
docker.io	V:45, I:47	95958	docker : Linux container runtime
games-emulator	I:0	21	games-emulator : Debian's emulators for games
bochs	V:0, I:0	8180	Bochs : IA-32 PC emulator
qemu-system	I:23	80	QEMU : full system emulation binaries
qemu-user	V:5, I:9	472779	QEMU : user mode emulation binaries
qemu-utils	V:13, I:110	12161	QEMU : utilities
qemu-system-x86	V:53, I:94	67511	KVM : full virtualization on x86 hardware with the hardware-assisted virtualization
virtualbox	V:4, I:5	154064	VirtualBox : x86 virtualization solution on i386 and amd64
gnome-boxes	V:1, I:7	6847	Boxes : Simple GNOME app to access virtual systems
xen-tools	V:0, I:1	719	tools to manage debian XEN virtual server
wine	V:13, I:57	204	Wine : Windows API Implementation (standard suite)
dosbox	V:1, I:13	2697	DOSBox : x86 emulator with Tandy/Herc/CGA/EGA/VGA/SVGA graphics, sound and DOS
lxc	V:9, I:12	1626	Linux containers user space tools
python3-venv	I:132	6	venv for creating virtual python environments (system library)
python3-virtualenv	V:8, I:43	379	virtualenv for creating isolated virtual python environments
pipx	V:6, I:42	3613	pipx for installing python applications in isolated environments

Tabela 9.27: Lista instrumentelor de virtualizare

- For disk images supported by [QEMU](#), mount them as [network block device](#) (see Secțiune [9.11.3](#)).
- Populate the target filesystem with required system data.
 - The use of programs such as `debootstrap` and `cdebootstrap` helps with this process (see Secțiune [9.11.4](#)).
 - Use installers of OSs under the full system emulation.
- Run a program under a virtualized environment.
 - [chroot](#) provides basic virtualized environment enough to compile programs, run console applications, and run daemons in it.
 - [QEMU](#) provides cross-platform CPU emulation.
 - [QEMU](#) with [KVM](#) provides full system emulation by the [hardware-assisted virtualization](#).
 - [VirtualBox](#) provides full system emulation on i386 and amd64 with or without the [hardware-assisted virtualization](#).

9.11.3 Mounting the virtual disk image file

For the raw disk image file, see Secțiune [9.7](#).

For other virtual disk image files, you can use `qemu-nbd(8)` to export them using [network block device](#) protocol and mount them using the `nbd` kernel module.

`qemu-nbd(8)` supports disk formats supported by [QEMU](#): `raw`, `qcow2`, `qcow`, `vmdk`, `vdi`, `bochs`, `cow` (user-mode Linux copy-on-write), `parallels`, `dmz`, `cloop`, `vpc`, `vfat` (virtual VFAT), and `host_device`.

The [network block device](#) can support partitions in the same way as the [loop device](#) (see Secțiune [9.7.3](#)). You can mount the first partition of `"disk.img"` as follows.

```
# modprobe nbd max_part=16
# qemu-nbd -v -c /dev/nbd0 disk.img
...
# mkdir /mnt/part1
# mount /dev/nbd0p1 /mnt/part1
```

Indicație

You may export only the first partition of `"disk.img"` using `-P 1` option to `qemu-nbd(8)`.

9.11.4 Chroot system

If you wish to try a new Debian environment from a terminal console, I recommend you to use [chroot](#). This enables you to run console applications of Debian `unstable` and `testing` without usual risks associated and without rebooting. `chroot(8)` is the most basic way.



Atenție

Examples below assumes both parent system and chroot system share the same amd64 CPU architecture.

Although you can manually create a `chroot(8)` environment using `debootstrap(1)`, this requires non-trivial efforts.

The [sbuid](#) package to build Debian packages from source uses the chroot environment managed by the [schroot](#) package. It comes with helper script `sbuid-createchroot(1)`. Let's learn how it works by running it as follows.

```
$ sudo mkdir -p /srv/chroot
$ sudo sbuild-createchroot -v --include=eatmydata,ccache unstable /srv/chroot/unstable- ↵
amd64-sbuild http://deb.debian.org/debian
...
```

You see how `debootstrap(8)` populates system data for unstable environment under `/srv/chroot/unstable-amd64-sbuild` for a minimal build system.

You can login to this environment using `schroot(1)`.

```
$ sudo schroot -v -c chroot:unstable-amd64-sbuild
```

You see how a system shell running under unstable environment is created.

Notă

The `/usr/sbin/policy-rc.d` file which always exits with 101 prevents daemon programs to be started automatically on the Debian system. See `/usr/share/doc/init-system-helpers/README.policy-rc.d.gz`.

Notă

Some programs under chroot may require access to more files from the parent system to function than `sbuild-createchroot` provides as above. For example, `/sys`, `/etc/passwd`, `/etc/group`, `/var/run/utmp`, `/var/log/wtmp`, etc. may need to be bind-mounted or copied.

Indicație

The `sbuild` package helps to construct a chroot system and builds a package inside the chroot using `schroot` as its backend. It is an ideal system to check build-dependencies. See more on [sbuild at Debian wiki](#) and [sbuild configuration example in "Guide for Debian Maintainers"](#).

Indicație

The `systemd-nspawn(1)` command helps to run a command or OS in a light-weight container in similar ways to chroot. It is more powerful since it uses namespaces to fully virtualize the process tree, IPC, hostname, domain name and, optionally, networking and user databases. See [systemd-nspawn](#).

9.11.5 Multiple desktop systems

If you wish to try a new GUI Desktop environment of any OS, I recommend you to use [QEMU](#) or [KVM](#) on a Debian stable system to run multiple desktop systems safely using [virtualization](#). These enable you to run any desktop applications including ones of Debian unstable and testing without usual risks associated with them and without rebooting.

Since pure [QEMU](#) is very slow, it is recommended to accelerate it with [KVM](#) when the host system supports it.

[Virtual Machine Manager](#) also known as `virt-manager` is a convenient GUI tool for managing KVM virtual machines via [libvirt](#).

The virtual disk image `virtdisk.qcow2` containing a Debian system for [QEMU](#) can be created using [debian-installer: Small CDs](#) as follows.

```
$ wget https://cdimage.debian.org/debian-cd/5.0.3/amd64/iso-cd/debian-503-amd64-netinst.iso
$ qemu-img create -f qcow2 virtdisk.qcow2 5G
$ qemu -hda virtdisk.qcow2 -cdrom debian-503-amd64-netinst.iso -boot d -m 256
...
```

Indicație

Running other GNU/Linux distributions such as [Ubuntu](#) and [Fedora](#) under [virtualization](#) is a great way to learn configuration tips. Other proprietary OSs may be run nicely under this GNU/Linux [virtualization](#), too.

See more tips at [Debian wiki: SystemVirtualization](#).

Capitolul 10

Data management

Tools and tips for managing binary and text data on the Debian system are described.

10.1 Sharing, copying, and archiving

**Avertisment**

The uncoordinated write access to actively accessed devices and files from multiple processes must not be done to avoid the [race condition](#). [File locking](#) mechanisms using `flock(1)` may be used to avoid it.

The security of the data and its controlled sharing have several aspects.

- The creation of data archive
- The remote storage access
- The duplication
- The tracking of the modification history
- The facilitation of data sharing
- The prevention of unauthorized file access
- The detection of unauthorized file modification

These can be realized by using some combination of tools.

- Archive and compression tools
 - Copy and synchronization tools
 - Network filesystems
 - Removable storage media
 - The secure shell
 - The authentication system
 - Version control system tools
 - Hash and cryptographic encryption tools
-

10.1.1 Archive and compression tools

Here is a summary of archive and compression tools available on the Debian system.

**Avertisment**

Do not set the "\$TAPE" variable unless you know what to expect. It changes tar(1) behavior.

- The gzipped tar(1) archive uses the file extension ".tgz" or ".tar.gz".
- The xz-compressed tar(1) archive uses the file extension ".txz" or ".tar.xz".
- Popular compression method in FOSS tools such as tar(1) has been moving as follows: gzip → bzip2 → xz
- cp(1), scp(1) and tar(1) may have some limitation for special files. cpio(1) is most versatile.
- cpio(1) is designed to be used with find(1) and other commands and suitable for creating backup scripts since the file selection part of the script can be tested independently.
- Internal structure of Libreoffice data files are ".jar" file which can be opened also by unzip.
- The de-facto cross platform archive tool is zip. Use it as "zip -rX" to attain the maximum compatibility. Use also the "-s" option, if the maximum file size matters.

10.1.2 Copy and synchronization tools

Here is a summary of simple copy and backup tools available on the Debian system.

Copying files with rsync(8) offers richer features than others.

- delta-transfer algorithm that sends only the differences between the source files and the existing files in the destination
- quick check algorithm (by default) that looks for files that have changed in size or in last-modified time
- "--exclude" and "--exclude-from" options similar to tar(1)
- "a trailing slash on the source directory" syntax that avoids creating an additional directory level at the destination.

Indicație

Version control system (VCS) tools in Tabel [10.14](#) can function as the multi-way copy and synchronization tools.

10.1.3 Idioms for the archive

Here are several ways to archive and unarchive the entire content of the directory "./source" using different tools.

GNU tar(1):

```
$ tar -cvJf archive.tar.xz ./source
$ tar -xvJf archive.tar.xz
```

Alternatively, by the following.

```
$ find ./source -xdev -print0 | tar -cvJf archive.tar.xz --null -T -
```

cpio(1):

```
$ find ./source -xdev -print0 | cpio -ov --null > archive.cpio; xz archive.cpio
$ zcat archive.cpio.xz | cpio -i
```

pachet	popcon(popularitate)	amărime	extensie	comanda	comentariu
tar	V:895, I:999	3085	.tar	tar(1)	the standard archiver (de facto standard)
cpio	V:321, I:998	1201	.cpio	cpio(1)	Arhivator în stil Unix System V, se utilizează cu find(1)
binutils	V:170, I:643	1118	.ar	ar(1)	archiver for the creation of static libraries
fastjar	V:1, I:10	183	.jar	fastjar(1)	archiver for Java (zip like)
pax	V:6, I:10	167	.pax	pax(1)	new POSIX standard archiver, compromise between tar and cpio
gzip	V:890, I:999	256	.gz	gzip(1), zcat(1), ...	GNU LZ77 compression utility (de facto standard)
bzip2	V:165, I:971	113	.bz2	bzip2(1), bzip2cat(1), ...	Burrows-Wheeler block-sorting compression utility with higher compression ratio than gzip(1) (slower than gzip with similar syntax)
lzma	V:0, I:12	349	.lzma	lzma(1)	LZMA compression utility with higher compression ratio than gzip(1) (deprecated)
xz-utils	V:311, I:980	1475	.xz	xz(1), xzdec(1), ...	XZ compression utility with higher compression ratio than bzip2(1) (slower than gzip but faster than bzip2; replacement for LZMA compression utility)
zstd	V:132, I:760	2312	.zstd	zstd(1), zstdcat(1), ...	Zstandard fast lossless compression utility
p7zip	V:8, I:259	8	.7z	7zr(1), p7zip(1)	7-Zip file archiver with high compression ratio (LZMA compression)
p7zip-full	V:28, I:280	12	.7z	7z(1), 7za(1)	7-Zip file archiver with high compression ratio (LZMA compression and others)
lzop	V:12, I:139	164	.lzo	lzop(1)	LZO compression utility with higher compression and decompression speed than gzip(1) (lower compression ratio than gzip with similar syntax)
zip	V:48, I:372	627	.zip	zip(1)	InfoZIP : DOS archive and compression tool
unzip	V:107, I:762	387	.zip	unzip(1)	InfoZIP : DOS unarchive and decompression tool

Tabela 10.1: List of archive and compression tools

pachet	popcon(popularity)	limite de instrument	funcție
coreutils	V:897, I:999	18457	GNU cp
openssh-client	V:902, I:996	5131	scp
openssh-server	V:752, I:808	3501	sshd
rsync	V:201, I:545	814	
unison	V:2, I:13	14	

Tabela 10.2: List of copy and synchronization tools

10.1.4 Idioms for the copy

Here are several ways to copy the entire content of the directory `"/source"` using different tools.

- Local copy: `"/source"` directory → `"/dest"` directory
- Remote copy: `"/source"` directory at local host → `"/dest"` directory at `"user@host.dom"` host

`rsync(8)`:

```
# cd ./source; rsync -aHAXSv . /dest
# cd ./source; rsync -aHAXSv . user@host.dom:/dest
```

You can alternatively use "a trailing slash on the source directory" syntax.

```
# rsync -aHAXSv ./source/ /dest
# rsync -aHAXSv ./source/ user@host.dom:/dest
```

Alternatively, by the following.

```
# cd ./source; find . -print0 | rsync -aHAXSv0 --files-from=- . /dest
# cd ./source; find . -print0 | rsync -aHAXSv0 --files-from=- . user@host.dom:/dest
```

GNU `cp(1)` și openSSH `scp(1)`:

```
# cd ./source; cp -a . /dest
# cd ./source; scp -pr . user@host.dom:/dest
```

GNU `tar(1)`:

```
# (cd ./source && tar cf - . ) | (cd /dest && tar xvpf - )
# (cd ./source && tar cf - . ) | ssh user@host.dom '(cd /dest && tar xvpf - )'
```

`cpio(1)`:

```
# cd ./source; find . -print0 | cpio -pvdm --null --sparse /dest
```

You can substitute `"/"` with `"foo"` for all examples containing `"/"` to copy files from `"/source/foo"` directory to `"/dest/foo"` directory.

You can substitute `"/"` with the absolute path `"/path/to/source/foo"` for all examples containing `"/"` to drop `"cd ./source;"`. These copy files to different locations depending on tools used as follows.

- `"/dest/foo"`: `rsync(8)`, GNU `cp(1)`, and `scp(1)`
- `"/dest/path/to/source/foo"`: GNU `tar(1)`, and `cpio(1)`

Indicație

`rsync(8)` and GNU `cp(1)` have option `"-u"` to skip files that are newer on the receiver.

10.1.5 Idioms for the selection of files

`find(1)` is used to select files for archive and copy commands (see Secțiune 10.1.3 and Secțiune 10.1.4) or for `xargs(1)` (see Secțiune 9.4.9). This can be enhanced by using its command arguments.

Basic syntax of `find(1)` can be summarized as the following.

- Its conditional arguments are evaluated from left to right.
- This evaluation stops once its outcome is determined.
- "Logical **OR**" (specified by `"-o"` between conditionals) has lower precedence than "logical **AND**" (specified by `"-a"` or nothing between conditionals).
- "Logical **NOT**" (specified by `"!"` before a conditional) has higher precedence than "logical **AND**".
- `"-prune"` always returns logical **TRUE** and, if it is a directory, searching of file is stopped beyond this point.
- `"-name"` matches the base of the filename with shell glob (see Secțiune 1.5.6) but it also matches its initial `"."` with metacharacters such as `"*"` and `"?"`. (New [POSIX](#) feature)
- `"-regex"` matches the full path with emacs style **BRE** (see Secțiune 1.6.2) as default.
- `"-size"` matches the file based on the file size (value preceded with `"+"` for larger, preceded with `"-"` for smaller)
- `"-newer"` matches the file newer than the one specified in its argument.
- `"-print0"` always returns logical **TRUE** and print the full filename ([null terminated](#)) on the standard output.

`find(1)` is often used with an idiomatic style as the following.

```
# find /path/to \  
-xdev -regextype posix-extended \  
-type f -regex ".*\.cpio|.*~" -prune -o \  
-type d -regex ".*\/\.git" -prune -o \  
-type f -size +99M -prune -o \  
-type f -newer /path/to/timestamp -print0
```

This means to do following actions.

1. Search all files starting from `"/path/to"`
2. Globally limit its search within its starting filesystem and uses **ERE** (see Secțiune 1.6.2) instead
3. Exclude files matching regex of `".*\.cpio"` or `".*~"` from search by stop processing
4. Exclude directories matching regex of `".*\/\.git"` from search by stop processing
5. Exclude files larger than 99 Megabytes (units of 1048576 bytes) from search by stop processing
6. Print filenames which satisfy above search conditions and are newer than `"/path/to/timestamp"`

Please note the idiomatic use of `"-prune -o"` to exclude files in the above example.

Notă

For non-Debian [Unix-like](#) system, some options may not be supported by `find(1)`. In such a case, please consider to adjust matching methods and replace `"-print0"` with `"-print"`. You may need to adjust related commands too.

10.1.6 Archive media

When choosing [computer data storage media](#) for important data archive, you should be careful about their limitations. For small personal data backup, I use CD-R and DVD-R by the brand name company and store in a cool, shaded, dry, clean environment. (Tape archive media seem to be popular for professional use.)

Notă

[A fire-resistant safe](#) are meant for paper documents. Most of the computer data storage media have less temperature tolerance than paper. I usually rely on multiple secure encrypted copies stored in multiple secure locations.

Optimistic storage life of archive media seen on the net (mostly from vendor info).

- 100+ years : Acid free paper with ink
- 100 years : Optical storage (CD/DVD, CD/DVD-R)
- 30 years : Magnetic storage (tape, floppy)
- 20 years : Phase change optical storage (CD-RW)

These do not count on the mechanical failures due to handling etc.

Optimistic write cycle of archive media seen on the net (mostly from vendor info).

- 250,000+ cycles : Harddisk drive
- 10,000+ cycles : Flash memory
- 1,000 cycles : CD/DVD-RW
- 1 cycles : CD/DVD-R, paper

**Atenție**

Figures of storage life and write cycle here should not be used for decisions on any critical data storage. Please consult the specific product information provided by the manufacture.

Indicație

Since CD/DVD-R and paper have only 1 write cycle, they inherently prevent accidental data loss by overwriting. This is advantage!

Indicație

If you need fast and frequent backup of large amount of data, a hard disk on a remote host linked by a fast network connection, may be the only realistic option.

Indicație

If you use re-writable media for your backups, use of filesystem such as [btrfs](#) or [zfs](#) which supports read-only snapshots may be a good idea.

10.1.7 Removable storage device

Removable storage devices may be any one of the following.

- [USB flash drive](#)
- [Hard disk drive](#)
- [Optical disc drive](#)
- Digital camera
- Digital music player

They may be connected via any one of the following.

- [USB](#)
- [IEEE 1394 / FireWire](#)
- [PC Card](#)

Modern desktop environments such as GNOME and KDE can mount these removable devices automatically without a matching `/etc/fstab` entry.

- `udisks2` package provides a daemon and associated utilities to mount and unmount these devices.
- [D-bus](#) creates events to initiate automatic processes.
- [PolicyKit](#) provides required privileges.

Indicație

Automounted devices may have the `uhelper=` mount option which is used by `umount(8)`.

Indicație

Automounting under modern desktop environment happens only when those removable media devices are not listed in `/etc/fstab`.

Mount point under modern desktop environment is chosen as `/media/username/disk_label` which can be customized by the following.

- `mlabel(1)` for FAT filesystem
- `genisoimage(1)` with `-V` option for ISO9660 filesystem
- `tune2fs(1)` with `-L` option for ext2/ext3/ext4 filesystem

Indicație

The choice of encoding may need to be provided as mount option (see Secțiune [8.1.3](#)).

Indicație

The use of the GUI menu to unmount a filesystem may remove its dynamically generated device node such as `/dev/sdc`. If you wish to keep its device node, unmount it with the `umount(8)` command from the shell prompt.

numele sistemului de fișiere	scenariul de utilizare tipic
FAT12	cross platform sharing of data on the floppy disk (<32MiB)
FAT16	cross platform sharing of data on the small hard disk like device (<2GiB)
FAT32	cross platform sharing of data on the large hard disk like device (<8TiB, supported by newer than MS Windows95 OSR2)
exFAT	cross platform sharing of data on the large hard disk like device (<512TiB, supported by WindowsXP, Mac OS X Snow Leopard 10.6.5, and Linux kernel since 5.4 release)
NTFS	cross platform sharing of data on the large hard disk like device (supported natively on MS Windows NT and later version, and supported by NTFS-3G via FUSE on Linux)
ISO9660	cross platform sharing of static data on CD-R and DVD+/-R
UDF	incremental data writing on CD-R and DVD+/-R (new)
MINIX	space efficient unix file data storage on the floppy disk
ext2	sharing of data on the hard disk like device with older Linux systems
ext3	sharing of data on the hard disk like device with older Linux systems
ext4	sharing of data on the hard disk like device with current Linux systems
btrfs	sharing of data on the hard disk like device with current Linux systems with read-only snapshots

Tabela 10.3: List of filesystem choices for removable storage devices with typical usage scenarios

10.1.8 Filesystem choice for sharing data

When sharing data with other system via removable storage device, you should format it with common [filesystem](#) supported by both systems. Here is a list of filesystem choices.

Indicație

See Secțiune [9.9.1](#) for cross platform sharing of data using device level encryption.

The FAT filesystem is supported by almost all modern operating systems and is quite useful for the data exchange purpose via removable hard disk like media.

When formatting removable hard disk like devices for cross platform sharing of data with the FAT filesystem, the following should be safe choices.

- Partitioning them with `fdisk(8)`, `cfdisk(8)` or `parted(8)` (see Secțiune [9.6.2](#)) into a single primary partition and to mark it as the following.
 - Type "6" for FAT16 for media smaller than 2GB.
 - Type "c" for FAT32 (LBA) for larger media.
- Formatting the primary partition with `mkfs.vfat(8)` with the following.
 - Just its device name, e.g. `"/dev/sda1"` for FAT16
 - The explicit option and its device name, e.g. `"-F 32 /dev/sda1"` for FAT32

When using the FAT or ISO9660 filesystems for sharing data, the following should be the safe considerations.

- Archiving files into an archive file first using `tar(1)`, or `cpio(1)` to retain the long filename, the symbolic link, the original Unix file permission and the owner information.
- Splitting the archive file into less than 2 GiB chunks with the `split(1)` command to protect it from the file size limitation.
- Encrypting the archive file to secure its contents from the unauthorized access.

Notă

For FAT filesystems by its design, the maximum file size is $(2^{32} - 1)$ bytes = (4GiB - 1 byte). For some applications on the older 32 bit OS, the maximum file size was even smaller $(2^{31} - 1)$ bytes = (2GiB - 1 byte). Debian does not suffer the latter problem.

Notă

Microsoft itself does not recommend to use FAT for drives or partitions of over 200 MB. Microsoft highlights its short comings such as inefficient disk space usage in their "[Overview of FAT, HPFS, and NTFS File Systems](#)". Of course, we should normally use the ext4 filesystem for Linux.

Indicație

For more on filesystems and accessing filesystems, please read "[Filesystems HOWTO](#)".

10.1.9 Sharing data via network

When sharing data with other system via network, you should use common service. Here are some hints.

network service	description of typical usage scenario
SMB/CIFS network mounted filesystem with Samba	sharing files via "Microsoft Windows Network", see <code>smb.conf(5)</code> and The Official Samba 3.x.x HOWTO and Reference Guide or the <code>samba-doc</code> package
NFS network mounted filesystem with the Linux kernel	sharing files via "Unix/Linux Network", see <code>exports(5)</code> and Linux NFS-HOWTO
HTTP service	sharing file between the web server/client
HTTPS service	sharing file between the web server/client with encrypted Secure Sockets Layer (SSL) or Transport Layer Security (TLS)
FTP service	sharing file between the FTP server/client

Tabela 10.4: List of the network service to chose with the typical usage scenario

Although these filesystems mounted over network and file transfer methods over network are quite convenient for sharing data, these may be insecure. Their network connection must be secured by the following.

- Encrypt it with [SSL/TLS](#)
- Tunnel it via [SSH](#)
- Tunnel it via [VPN](#)
- Limit it behind the secure firewall

See also Secțiune [6.5](#) and Secțiune [6.6](#).

10.2 Backup and recovery

We all know that computers fail sometime or human errors cause system and data damages. Backup and recovery operations are the essential part of successful system administration. All possible failure modes hit you some day.

Indicație

Keep your backup system simple and backup your system often. Having backup data is more important than how technically good your backup method is.

10.2.1 Backup and recovery policy

There are 3 key factors which determine actual backup and recovery policy.

1. Knowing what to backup and recover.

- Data files directly created by you: data in "~/"
- Data files created by applications used by you: data in "/var/" (except "/var/cache/", "/var/run/", and "/var/tmp/")
- System configuration files: data in "/etc/"
- Local programs: data in "/usr/local/" or "/opt/"
- System installation information: a memo in plain text on key steps (partition, ...)
- Proven set of data: confirmed by experimental recovery operations in advance
 - Cron job as a user process: files in "/var/spool/cron/crontabs" directory and restart cron(8). See Secțiune 9.4.14 for cron(8) and crontab(1).
 - Systemd timer jobs as user processes: files in "~/.config/systemd/user" directory. See systemd.timer(5) and systemd.service(5).
 - Autostart jobs as user processes: files in "~/.config/autostart" directory. See [Desktop Application Autostart Specification](#).

2. Knowing how to backup and recover.

- Secure storage of data: protection from overwrite and system failure
- Frequent backup: scheduled backup
- Redundant backup: data mirroring
- Fool proof process: easy single command backup

3. Assessing risks and costs involved.

- Risk of data when lost
 - Data should be at least on different disk partitions preferably on different disks and machines to withstand the filesystem corruption. Important data are best stored on a read-only filesystem. [1](#)
- Risk of data when breached
 - Sensitive identity data such as "/etc/ssh/ssh_host_*_key", "~/.gnupg/*", "~/.ssh/*", "~/.local/share/*", "/etc/passwd", "/etc/shadow", "popularity-contest.conf", "/etc/ppp/pap-secrets", and "/etc/exim4/passwd.client" should be backed up as encrypted. [2](#) (See Secțiune 9.9.)
 - Never hard code system login password nor decryption passphrase in any script even on any trusted system. (See Secțiune 10.3.6.)
- Failure mode and their possibility
 - Hardware (especially HDD) will break
 - Filesystem may be corrupted and data in it may be lost
 - Remote storage system can't be trusted for security breaches
 - Weak password protection can be easily compromised
 - File permission system may be compromised
- Required resources for backup: human, hardware, software, ...
 - Automatic scheduled backup with cron job or systemd timer job

¹A write-once media such as CD/DVD-R can prevent overwrite accidents. (See Secțiune 9.8 for how to write to the storage media from the shell commandline. GNOME desktop GUI environment gives you easy access via menu: "Places → CD/DVD Creator".)

²Some of these data can not be regenerated by entering the same input string to the system.

Indicație

You can recover debconf configuration data with "debconf-set-selections debconf-selections" and dpkg selection data with "dpkg --set-selection <dpkg-selections.list".

Notă

Do not back up the pseudo-filesystem contents found on /proc, /sys, /tmp, and /run (see Secțiune 1.2.12 and Secțiune 1.2.13). Unless you know exactly what you are doing, they are huge useless data.

Notă

You may wish to stop some application daemons such as MTA (see Secțiune 6.2.4) while backing up data.

10.2.2 Backup utility suites

Here is a select list of notable backup utility suites available on the Debian system.

pachet	popcon	popularity	descriere
bacula-common	V:6, I:7	2501	Bacula : network backup, recovery and verification - common support files
bacula-client	V:0, I:2	199	Bacula : network backup, recovery and verification - client meta-package
bacula-console	V:0, I:2	112	Bacula : network backup, recovery and verification - text console
bacula-server	I:0	199	Bacula : network backup, recovery and verification - server meta-package
amanda-common	V:0, I:2	9851	Amanda : Advanced Maryland Automatic Network Disk Archiver (Libs)
amanda-client	V:0, I:2	1099	Amanda : Advanced Maryland Automatic Network Disk Archiver (Client)
amanda-server	V:0, I:0	1093	Amanda : Advanced Maryland Automatic Network Disk Archiver (Server)
backuppc	V:1, I:1	3088	BackupPC is a high-performance, enterprise-grade system for backing up PCs (disk based)
duplicity	V:6, I:50	2649	(remote) incremental backup
deja-dup	V:30, I:45	5031	GUI frontend for duplicity
borgbackup	V:12, I:27	3477	(remote) deduplicating backup
borgmatic	V:3, I:4	946	borgbackup helper
rdiff-backup	V:2, I:7	1207	(remote) incremental backup
restic	V:4, I:9	24708	(remote) incremental backup
backupninja	V:2, I:2	360	lightweight, extensible meta-backup system
slbackup	V:0, I:0	147	(remote) incremental backup
backup-manager	V:0, I:0	573	command-line backup tool
backup2l	V:0, I:0	110	low-maintenance backup/restore tool for mountable media (disk based)

Tabela 10.5: List of backup suite utilities

Backup tools have their specialized focuses.

- [Mondo Rescue](#) is a backup system to facilitate restoration of complete system quickly from backup CD/DVD etc. without going through normal system installation processes.

- [Bacula](#), [Amanda](#), and [BackupPC](#) are full featured backup suite utilities which are focused on regular backups over network.
- [Duplicity](#), and [Borg](#) are simpler backup utilities for typical workstations.

10.2.3 Backup tips

For a personal workstation, full featured backup suite utilities designed for the server environment may not serve well. At the same time, existing backup utilities for workstations may have some shortcomings.

Here are some tips to make backup easier with minimal user efforts. These techniques may be used with any backup utilities.

For demonstration purpose, let's assume the primary user and group name to be penguin and create a backup and snapshot script example `"/usr/local/bin/bkss.sh"` as:

```
#!/bin/sh -e
SRC="$1" # source data path
DSTFS="$2" # backup destination filesystem path
DSTSV="$3" # backup destination subvolume name
DSTSS="${DSTFS}/${DSTSV}-snapshot" # snapshot destination path
if [ "$(stat -f -c %T "$DSTFS")" != "btrfs" ]; then
    echo "E: $DSTFS needs to be formatted to btrfs" >&2 ; exit 1
fi
MSGID=$(notify-send -p "bkup.sh $DSTSV" "in progress ...")
if [ ! -d "$DSTFS/$DSTSV" ]; then
    btrfs subvolume create "$DSTFS/$DSTSV"
    mkdir -p "$DSTSS"
fi
rsync -aHxS --delete --mkpath "${SRC}/" "${DSTFS}/${DSTSV}"
btrfs subvolume snapshot -r "${DSTFS}/${DSTSV}" "${DSTSS}/${date -u --iso=min}"
notify-send -r "$MSGID" "bkup.sh $DSTSV" "finished!"
```

Here, only the basic tool `rsync(1)` is used to facilitate system backup and the storage space is efficiently used by [Btrfs](#).

Indicație

FYI: This author uses his own similar shell script ["bss: Btrfs Subvolume Snapshot Utility"](#) for his workstation.

10.2.3.1 GUI backup

Here is an example to setup the single GUI click backup.

- Prepare a USB storage device to be used for backup.
 - Format a USB storage device with one partition in btrfs with its label name as "BKUP". This can be encrypted (see [Secțiune 9.9.1](#)).
 - Plug this in to your system. The desktop system should automatically mount it as `"/media/penguin/BKUP"`.
 - Execute `"sudo chown penguin:penguin /media/penguin/BKUP"` to make it writable by the user.
- Create `"~/local/share/applications/BKUP.desktop"` following techniques written in [Secțiune 9.4.10](#) as:

```
[Desktop Entry]
Name=bkss
Comment=Backup and snapshot of ~/Documents
Exec=/usr/local/bin/bkss.sh /home/penguin/Documents /media/penguin/BKUP Documents
Type=Application
```

For each GUI click, your data is backed up from `"~/Documents"` to a USB storage device and a read-only snapshot is created.

10.2.3.2 Mount event triggered backup

Here is an example to setup for the automatic backup triggered by the mount event.

- Prepare a USB storage device to be used for backup as in Secțiune [10.2.3.1](#).
- Create a systemd service unit file "`~/ .config/systemd/user/back-BKUP.service`" as:

```
[Unit]
Description=USB Disk backup
Requires=media-%u-BKUP.mount
After=media-%u-BKUP.mount

[Service]
ExecStart=/usr/local/bin/bkss.sh %h/Documents /media/%u/BKUP Documents
StandardOutput=append:%h/.cache/systemd-snap.log
StandardError=append:%h/.cache/systemd-snap.log

[Install]
WantedBy=media-%u-BKUP.mount
```

- Enable this systemd unit configuration with the following:

```
$ systemctl --user enable bkup-BKUP.service
```

For each mount event, your data is backed up from "`~/Documents`" to a USB storage device and a read-only snapshot is created.

Here, names of systemd mount units that systemd currently has in memory can be asked to the service manager of the calling user with "`systemctl --user list-units --type=mount`".

10.2.3.3 Timer event triggered backup

Here is an example to setup for the automatic backup triggered by the timer event.

- Prepare a USB storage device to be used for backup as in Secțiune [10.2.3.1](#).
- Create a systemd timer unit file "`~/ .config/systemd/user/snap-Documents.timer`" as:

```
[Unit]
Description=Run btrfs subvolume snapshot on timer
Documentation=man:btrfs(1)

[Timer]
OnStartupSec=30
OnUnitInactiveSec=900

[Install]
WantedBy=timers.target
```

- Create a systemd service unit file "`~/ .config/systemd/user/snap-Documents.service`" as:

```
[Unit]
Description=Run btrfs subvolume snapshot
Documentation=man:btrfs(1)

[Service]
Type=oneshot
Nice=15
```

```
ExecStart=/usr/local/bin/bkss.sh %h/Documents /media/%u/BKUP Documents
IOSchedulingClass=idle
CPUSchedulingPolicy=idle
StandardOutput=append:%h/.cache/systemd-snap.log
StandardError=append:%h/.cache/systemd-snap.log
```

- Enable this systemd unit configuration with the following:

```
$ systemctl --user enable snap-Documents.timer
```

For each timer event, your data is backed up from "~/Documents" to a USB storage device and a read-only snapshot is created.

Here, names of systemd timer user units that systemd currently has in memory can be asked to the service manager of the calling user with "systemctl --user list-units --type=timer".

For the modern desktop system, this systemd approach can offer more fine grained control than the traditional Unix ones using at(1), cron(8), or anacron(8).

10.3 Data security infrastructure

The data security infrastructure is provided by the combination of data encryption tool, message digest tool, and signature tool.

pachet	popcon(popularitete)	dimensiune	comanda	descriere
gnupg	V:352, I:878	468	gpg(1)	GNU Privacy Guard - OpenPGP encryption and signing tool
gpgv	V:289, I:964	559	gpgv(1)	GNU Privacy Guard - signature verification tool
paperkey	V:1, I:14	58	paperkey(1)	extract just the secret information out of OpenPGP secret keys
cryptsetup	V:16, I:81	463	cryptsetup(8) ...	Utilities for dm-crypt block device encryption supporting LUKS
coreutils	V:897, I:999	18457	md5sum(1)	compute and check MD5 message digest
coreutils	V:897, I:999	18457	sha1sum(1)	compute and check SHA1 message digest
openssl	V:843, I:996	2493	openssl(1ssl)	compute message digest with "openssl dgst" (OpenSSL)
libsecret-tools	V:0, I:9	49	secret-tool(1)	store and retrieve passwords (CLI)
seahorse	V:78, I:270	7971	seahorse(1)	key management tool (GNOME)

Tabela 10.6: List of data security infrastructure tools

See Secțiune 9.9 on [dm-crypt](#) and [fscrypt](#) which implement automatic data encryption infrastructure via Linux kernel modules.

10.3.1 Key management for GnuPG

Here are [GNU Privacy Guard](#) commands for the basic key management.

Here is the meaning of the trust code.

The following uploads my key "1DD8D791" to the popular keyserver "hkp://keys.gnupg.net".

comanda	descriere
<code>gpg --gen-key</code>	generează o cheie nouă
<code>gpg --gen-revoke my_user_ID</code>	generate revoke key for my_user_ID
<code>gpg --edit-key user_ID</code>	edit key interactively, "help" for help
<code>gpg -o fișier --export</code>	export all keys to file
<code>gpg --import fișier</code>	import all keys from file
<code>gpg --send-keys ID-utilizator</code>	send key of user_ID to keyserver
<code>gpg --recv-keys ID-utilizator</code>	recv. key of user_ID from keyserver
<code>gpg --list-keys ID-utilizator</code>	list keys of user_ID
<code>gpg --list-sigs ID-utilizator</code>	list sig. of user_ID
<code>gpg --check-sigs ID-utilizator</code>	check sig. of user_ID
<code>gpg --fingerprint ID-utilizator</code>	check fingerprint of user_ID
<code>gpg --refresh-keys</code>	update local keyring

Tabela 10.7: List of GNU Privacy Guard commands for the key management

cod	description of trust
-	no owner trust assigned / not yet calculated
e	trust calculation failed
q	not enough information for calculation
n	never trust this key
m	marginally trusted
f	fully trusted
u	ultimately trusted

Tabela 10.8: List of the meaning of the trust code

```
$ gpg --keyserver hkp://keys.gnupg.net --send-keys 1DD8D791
```

A good default keyserver set up in "`~/ .gnupg/gpg.conf`" (or old location "`~/ .gnupg/options`") contains the following.

```
keyserver hkp://keys.gnupg.net
```

The following obtains unknown keys from the keyserver.

```
$ gpg --list-sigs --with-colons | grep '^sig.*\[User ID not found\]' |\
  cut -d ':' -f 5 | sort | uniq | xargs gpg --recv-keys
```

There was a bug in [OpenPGP Public Key Server](#) (pre version 0.9.6) which corrupted key with more than 2 sub-keys. The newer gnupg (>1.2.1-2) package can handle these corrupted subkeys. See `gpg(1)` under "`--repair-pks-subkey-bu`" option.

10.3.2 Using GnuPG on files

Here are examples for using [GNU Privacy Guard](#) commands on files.

10.3.3 Using GnuPG with Mutt

Add the following to "`~/ .muttrc`" to keep a slow GnuPG from automatically starting, while allowing it to be used by typing "S" at the index menu.

```
macro index S ":toggle pgp_verify_sig\n"
set pgp_verify_sig=no
```

comanda	descriere
<code>gpg -a -s fișier</code>	sign file into ASCII armored file.asc
<code>gpg --armor --sign fișier</code>	, ,
<code>gpg --clearsign fișier</code>	clear-sign message
<code>gpg --clearsign file mail foo@example.org</code>	mail a clear-signed message to foo@example.org
<code>gpg --clearsign --not-dash-escaped patchfile</code>	clear-sign patchfile
<code>gpg --verify fișier</code>	verify clear-signed file
<code>gpg -o file.sig -b file</code>	create detached signature
<code>gpg -o file.sig --detach-sign file</code>	, ,
<code>gpg --verify file.sig file</code>	verify file with file.sig
<code>gpg -o crypt_file.gpg -r name -e file</code>	public-key encryption intended for name from file to binary crypt_file.gpg
<code>gpg -o crypt_file.gpg --recipient name --encrypt file</code>	, ,
<code>gpg -o crypt_file.asc -a -r name -e file</code>	public-key encryption intended for name from file to ASCII armored crypt_file.asc
<code>gpg -o crypt_file.gpg -c file</code>	symmetric encryption from file to crypt_file.gpg
<code>gpg -o crypt_file.gpg --symmetric file</code>	, ,
<code>gpg -o crypt_file.asc -a -c file</code>	symmetric encryption intended for name from file to ASCII armored crypt_file.asc
<code>gpg -o file -d crypt_file.gpg -r name</code>	decriptare
<code>gpg -o file --decrypt crypt_file.gpg</code>	, ,

Tabela 10.9: List of GNU Privacy Guard commands on files

10.3.4 Using GnuPG with Vim

The gnupg plugin let you run GnuPG transparently for files with extension ".pgp", ".asc", and ".pgp".³

```
$ sudo aptitude install vim-scripts
$ echo "packadd! gnupg" >> ~/.vim/vimrc
```

10.3.5 Suma MD5

md5sum(1) provides utility to make a digest file using the method in [rfc1321](#) and verifying each file with it.

```
$ md5sum foo bar >baz.md5
$ cat baz.md5
d3b07384d113edec49eaa6238ad5ff00  foo
c157a79031e1c40f85931829bc5fc552  bar
$ md5sum -c baz.md5
foo: OK
bar: OK
```

Notă

The computation for the [MD5](#) sum is less CPU intensive than the one for the cryptographic signature by [GNU Privacy Guard \(GnuPG\)](#). Usually, only the top level digest file is cryptographically signed to ensure data integrity.

10.3.6 Password keyring

On GNOME system, the GUI tool seahorse(1) manages passwords and stores them securely in the keyring ~/.local/share/secret-tool(1) can store password to the keyring from the command line.

Let's store passphrase used for LUKS/dm-crypt encrypted disk image

```
$ secret-tool store --label='LUKS passphrase for disk.img' LUKS my_disk.img
Password: *****
```

This stored password can be retrieved and fed to other programs, e.g., cryptsetup(8).

```
$ secret-tool lookup LUKS my_disk.img | \
  cryptsetup open disk.img disk_img --type luks --keyring -
$ sudo mount /dev/mapper/disk_img /mnt
```

Indicație

Whenever you need to provide password in a script, use secret-tool and avoid directly hardcoding the passphrase in it.

10.4 Source code merge tools

There are many merge tools for the source code. Following commands caught my eyes.

³If you use "~/.vimrc" instead of "~/.vim/vimrc", please substitute accordingly.

pachet	popcon(popularity)	limite	comanda	descriere
patch	V:77, I:716	242	patch(1)	apply a diff file to an original
vim	V:85, I:353	4077	vimdiff(1)	compare 2 files side by side in vim
imediff	V:0, I:0	348	imediff(1)	interactive full screen 2/3-way merge tool
meld	V:5, I:25	3546	meld(1)	compare and merge files (GTK)
wiggle	V:0, I:0	175	wiggle(1)	apply rejected patches
diffutils	V:879, I:997	1768	diff(1)	compare files line by line
diffutils	V:879, I:997	1768	diff3(1)	compare and merges three files line by line
quilt	V:1, I:19	885	quilt(1)	manage series of patches
wdiff	V:5, I:43	651	wdiff(1)	display word differences between text files
diffstat	V:10, I:106	79	diffstat(1)	produce a histogram of changes by the diff
patchutils	V:12, I:104	242	combinediff(1)	create a cumulative patch from two incremental patches
patchutils	V:12, I:104	242	dehtmldiff(1)	extract a diff from an HTML page
patchutils	V:12, I:104	242	filterdiff(1)	extract or excludes diffs from a diff file
patchutils	V:12, I:104	242	fixcvsdiff(1)	fix diff files created by CVS that patch(1) mis-interprets
patchutils	V:12, I:104	242	flipdiff(1)	exchange the order of two patches
patchutils	V:12, I:104	242	grepdiff(1)	show which files are modified by a patch matching a regex
patchutils	V:12, I:104	242	interdiff(1)	show differences between two unified diff files
patchutils	V:12, I:104	242	lsdiff(1)	show which files are modified by a patch
patchutils	V:12, I:104	242	recountdiff(1)	recompute counts and offsets in unified context diffs
patchutils	V:12, I:104	242	rediff(1)	fix offsets and counts of a hand-edited diff
patchutils	V:12, I:104	242	splitdiff(1)	separate out incremental patches
patchutils	V:12, I:104	242	unwrapdiff(1)	demangle patches that have been word-wrapped
dirdiff	V:0, I:1	167	dirdiff(1)	display differences and merge changes between directory trees
docdiff	V:0, I:0	554	docdiff(1)	compare two files word by word / char by char
makepatch	V:0, I:0	99	makepatch(1)	generate extended patch files
makepatch	V:0, I:0	99	applypatch(1)	apply extended patch files

Tabela 10.10: List of source code merge tools

10.4.1 Extracting differences for source files

The following procedures extract differences between two source files and create unified diff files "file.patch0" or "file.patch1" depending on the file location.

```
$ diff -u file.old file.new > file.patch0
$ diff -u old/file new/file > file.patch1
```

10.4.2 Merging updates for source files

The diff file (alternatively called patch file) is used to send a program update. The receiving party applies this update to another file by the following.

```
$ patch -p0 file < file.patch0
$ patch -p1 file < file.patch1
```

10.4.3 Interactive merge

If you have two versions of a source code, you can perform 2-way merge interactively using `imediff(1)` by the following.

```
$ imediff -o file.merged file.old file.new
```

If you have three versions of a source code, you can perform 3-way merge interactively using `imediff(1)` by the following.

```
$ imediff -o file.merged file.yours file.base file.theirs
```

10.5 Git

Git is the tool of choice these days for the [version control system \(VCS\)](#) since Git can do everything for both local and remote source code management.

Debian provides free Git services via [Debian Salsa service](#). Its documentation can be found at <https://wiki.debian.org/Salsa>.

Here are some Git related packages.

10.5.1 Configuration of Git client

You may wish to set several global configuration in "`~/.gitconfig`" such as your name and email address used by Git by the following.

```
$ git config --global user.name "Name Surname"
$ git config --global user.email yourname@example.com
```

You may also customize the Git default behavior by the following.

```
$ git config --global init.defaultBranch main
$ git config --global pull.rebase true
$ git config --global push.default current
```

If you are too used to CVS or Subversion commands, you may wish to set several command aliases by the following.

pachet	popcon(popularity)	limita	comanda	descriere
git	V:379, I:598	50172	git(7)	Git, the fast, scalable, distributed revision control system
gitk	V:3, I:30	2003	gitk(1)	GUI Git repository browser with history
git-gui	V:1, I:19	2508	git-gui(1)	GUI for Git (No history)
git-email	V:0, I:11	1187	git-send-email(1)	collection of patches as email from the Git
git-buildpackage	V:1, I:8	2030	git-buildpackage(1)	package the Debian packaging with the Git
dgit	V:0, I:1	646	dgit(1)	git interoperability with the Debian archive
imediff	V:0, I:0	348	git-ime(1)	interactive git commit split helper tool
stgit	V:0, I:0	604	stg(1)	quilt on top of git (Python)
git-doc	I:12	14238	N/D	official documentation for Git
gitmagic	I:0	721	N/D	"Git Magic", easier to understand guide for Git

Tabela 10.11: List of git related packages and commands

```
$ git config --global alias.ci "commit -a"
$ git config --global alias.co checkout
```

You can check your global configuration by the following.

```
$ git config --global --list
```

10.5.2 Basic Git commands

Git operation involves several data.

- The working tree which holds user facing files and to which you make changes.
 - The changes to be recorded must be explicitly selected and staged to the index. This is `git add` and `git rm` commands.
- The index which holds staged files.
 - Staged files will be committed to the local repository upon the subsequent request. This is `git commit` command.
- The local repository which holds committed files.
 - Git records the linked history of the committed data and organizes them as branches in the repository.
 - The local repository can send data to the remote repository by `git push` command.
 - The local repository can receive data from the remote repository by `git fetch` and `git pull` commands.
 - * The `git pull` command performs `git merge` or `git rebase` command after `git fetch` command.
 - * Here, `git merge` combines two separate branches of history at the end to a point. (This is default of `git pull` without customization and may be good for upstream people who publish branch to many people.)
 - * Here, `git rebase` creates one single branch of sequential history of the remote branch one followed by the local branch one. (This is `pull.rebase true` customization case and may be good for rest of us.)
- The remote repository which holds committed files.
 - The communication to the remote repository uses secure communication protocols such as SSH or HTTPS.

The working tree is files outside of the `.git/` directory. Files inside of the `.git/` directory hold the index, the local repository data, and some git configuration text files.

Here is an overview of main Git commands.

Comanda git	funcție
git init	create the (local) repository
git clone URL	clone the remote repository to a local repository with the working tree
git pull origin main	update the local main branch by the remote repository origin
git add .	add file(s) in the working tree to the index for pre-existing files in index only
git add -A .	add file(s) in the working tree to the index for all files including removals
git rm nume-fișier	remove file(s) from the working tree and the index
git commit	commit staged changes in the index to the local repository
git commit -a	add all changes in the working tree to the index and commit them to the local repository (add + commit)
git push -u origin branch_name	update the remote repository origin by the local branch_name branch (initial invocation)
git push origin branch_name	update the remote repository origin by the local branch_name branch (subsequent invocation)
git diff treeish1 treeish2	show difference between <i>treeish1</i> commit and <i>treeish2</i> commit
gitk	GUI display of VCS repository branch history tree

Tabela 10.12: Main Git commands

10.5.3 Git tips

Here are some Git tips.



Avertisment

Do not use the tag string with spaces in it even if some tools such as `gitk(1)` allow you to use it. It may choke some other `git` commands.



Atenție

If a local branch which has been pushed to remote repository is rebased or squashed, pushing this branch has risks and requires `--force` option. This is usually not an acceptable for `main` branch but may be acceptable for a topic branch before merging to `main` branch.



Atenție

Invoking a `git` subcommand directly as `"git-xyz"` from the command line has been deprecated since early 2006.

Indicație

If there is a executable file `git-foo` in the path specified by `$PATH`, entering `"git foo"` without hyphen to the command line invokes this `git-foo`. This is a feature of the `git` command.

10.5.4 Git references

See the following.

Git command line	funcție
<code>gitk --all</code>	see complete Git history and operate on them such as resetting HEAD to another commit, cheery-picking patches, creating tags and branches ...
<code>git stash</code>	get the clean working tree without loosing data
<code>git remote -v</code>	check settings for remote
<code>git branch -vv</code>	check settings for branch
<code>git status</code>	show working tree status
<code>git config -l</code>	list git settings
<code>git reset --hard HEAD; git clean -x -d -f</code>	revert all working tree changes and clean them up completely
<code>git rm --cached nume-fișier</code>	revert staged index changed by <code>git add filename</code>
<code>git reflog</code>	get reference log (useful for recovering commits from the removed branch)
<code>git branch new_branch_name HEAD@{6}</code>	create a new branch from reflog information
<code>git remote add new_remote URL</code>	add a new_remote remote repository pointed by URL
<code>git remote rename origin upstream</code>	rename the remote repository name from origin to upstream
<code>git branch -u upstream/branch_name</code>	set the remote tracking to the remote repository upstream and its branch name branch_name.
<code>git remote set-url origin https://foo/bar.git</code>	change URL of origin
<code>git remote set-url --push upstream DISABLED</code>	disable push to upstream (Edit .git/config to re-enable)
<code>git remote update upstream</code>	fetch updates of all remote branches in the upstream repository
<code>git fetch upstream foo:upstream-foo</code>	create a local (possibly orphan) upstream-foo branch as a copy of foo branch in the upstream repository
<code>git checkout -b topic_branch ; git push -u topic_branch origin</code>	make a new topic_branch and push it to origin
<code>git branch -m oldname newname</code>	rename local branch name
<code>git push -d origin branch_to_be_removed</code>	remove remote branch (new method)
<code>git push origin :branch_to_be_removed</code>	remove remote branch (old method)
<code>git checkout --orphan unconnected</code>	create a new unconnected branch
<code>git rebase -i origin/main</code>	reorder/drop/squish commits from origin/main to clean branch history
<code>git reset HEAD^; git commit --amend</code>	squash last 2 commits into one
<code>git checkout topic_branch ; git merge --squash topic_branch</code>	squash entire topic_branch into a commit
<code>git fetch --unshallow --update-head-ok origin '+refs/heads/*:refs/heads/*'</code>	convert a shallow clone to the full clone of all branches
<code>git ime</code>	split the last commit into a series of file-by-file smaller commits etc. (imediff package required)
<code>git repack -a -d; git prune</code>	repack the local repository into single pack (this may limit chance of lost data recovery from erased branch etc.)

Tabela 10.13: Git tips

- [pagina de manual: git\(1\)](/usr/share/doc/git-doc/git.html) (/usr/share/doc/git-doc/git.html)
- [Manualul utilizatorului git](/usr/share/doc/git-doc/user-manual.html) (/usr/share/doc/git-doc/user-manual.html)
- [A tutorial introduction to git](/usr/share/doc/git-doc/gittutorial.html) (/usr/share/doc/git-doc/gittutorial.html)
- [A tutorial introduction to git: part two](/usr/share/doc/git-doc/gittutorial-2.html) (/usr/share/doc/git-doc/gittutorial-2.html)
- [Everyday GIT With 20 Commands Or So](/usr/share/doc/git-doc/giteveryday.html) (/usr/share/doc/git-doc/giteveryday.html)
- [Git Magic](/usr/share/doc/gitmagic/html/index.html) (/usr/share/doc/gitmagic/html/index.html)

10.5.5 Other version control systems

The [version control systems \(VCS\)](#) is sometimes known as the revision control system (RCS), or the software configuration management (SCM).

Here is a summary of the notable other non-Git VCS on the Debian system.

pachet	popcon	popularity	instrument	VCS type	comentariu
mercurial	V:3, I:27	1330	Mercurial	distributed	DVCS in Python and some C
darcs	V:0, I:3	48209	Darcs	distributed	DVCS with smart algebra of patches (slow)
bazaar	I:5	28	GNU Bazaar	distributed	DVCS influenced by t l a written in Python (historic)
tla	V:0, I:0	1017	GNU arch	distributed	DVCS mainly by Tom Lord (historic)
subversion	V:10, I:61	4850	Subversion	remote	"CVS done right", newer standard remote VCS (historic)
cvs	V:3, I:28	4835	CVS	remote	previous standard remote VCS (historic)
tkcvs	V:0, I:0	34	CVS, ...	remote	GUI display of VCS (CVS, Subversion, RCS) repository tree
rcs	V:1, I:10	578	RCS	local	" Unix SCCS done right" (historic)
cssc	V:0, I:0	2044	CSSC	local	clone of the Unix SCCS (historic)

Tabela 10.14: List of other version control system tools

Capitolul 11

Data conversion

Tools and tips for converting data formats on the Debian system are described.

Standard based tools are in very good shape but support for proprietary data formats are limited.

11.1 Text data conversion tools

Following packages for the text data conversion caught my eyes.

pachet	popcon(popularity)	limita (limit)	keyword	descriere
libc6	V:932, I:999	5678	charset	text encoding converter between locales by <code>iconv(1)</code> (fundamental)
recode	V:1, I:14	528	charset+eol	text encoding converter between locales (versatile, more aliases and features)
konwert	V:1, I:44	137	charset	text encoding converter between locales (fancy)
nkf	V:0, I:8	359	charset	character set translator for Japanese
tcs	V:0, I:0	518	charset	character set translator
unaccent	V:0, I:0	34	charset	replace accented letters by their unaccented equivalent
tofromdos	V:0, I:13	50	eol	text format converter between DOS and Unix: <code>fromdos(1)</code> and <code>todos(1)</code>
macutils	V:0, I:0	319	eol	text format converter between Macintosh and Unix: <code>frommac(1)</code> and <code>tomac(1)</code>

Tabela 11.1: List of text data conversion tools

11.1.1 Converting a text file with `iconv`

Indicație

`iconv(1)` is provided as a part of the `libc6` package and it is always available on practically all Unix-like systems to convert the encoding of characters.

You can convert encodings of a text file with `iconv(1)` by the following.

```
$ iconv -f encoding1 -t encoding2 input.txt >output.txt
```

encoding value	utilizare
ASCII	American Standard Code for Information Interchange , 7 bit code w/o accented characters
UTF-8	current multilingual standard for all modern OSs
ISO-8859-1	old standard for western European languages, ASCII + accented characters
ISO-8859-2	old standard for eastern European languages, ASCII + accented characters
ISO-8859-15	old standard for western European languages, ISO-8859-1 with euro sign
CP850	code page 850, Microsoft DOS characters with graphics for western European languages, ISO-8859-1 variant
CP932	code page 932, Microsoft Windows style Shift-JIS variant for Japanese
CP936	code page 936, Microsoft Windows style GB2312 , GBK or GB18030 variant for Simplified Chinese
CP949	code page 949, Microsoft Windows style EUC-KR or Unified Hangul Code variant for Korean
CP950	code page 950, Microsoft Windows style Big5 variant for Traditional Chinese
CP1251	code page 1251, Microsoft Windows style encoding for the Cyrillic alphabet
CP1252	code page 1252, Microsoft Windows style ISO-8859-15 variant for western European languages
KOI8-R	old Russian UNIX standard for the Cyrillic alphabet
ISO-2022-JP	standard encoding for Japanese email which uses only 7 bit codes
eucJP	old Japanese UNIX standard 8 bit code and completely different from Shift-JIS
Shift-JIS	JIS X 0208 Appendix 1 standard for Japanese (see CP932)

Tabela 11.2: List of encoding values and their usage

Encoding values are case insensitive and ignore "-" and "_" for matching. Supported encodings can be checked by the "iconv -l" command.

Notă

Some encodings are only supported for the data conversion and are not used as locale values (Secțiune 8.1).

For character sets which fit in single byte such as [ASCII](#) and [ISO-8859](#) character sets, the [character encoding](#) means almost the same thing as the character set.

For character sets with many characters such as [JIS X 0213](#) for Japanese or [Universal Character Set \(UCS, Unicode, ISO-10646-1\)](#) for practically all languages, there are many encoding schemes to fit them into the sequence of the byte data.

- [EUC](#) and [ISO/IEC 2022 \(also known as JIS X 0202\)](#) for Japanese
- [UTF-8](#), [UTF-16/UCS-2](#) and [UTF-32/UCS-4](#) for Unicode

For these, there are clear differentiations between the character set and the character encoding.

The [code page](#) is used as the synonym to the character encoding tables for some vendor specific ones.

Notă

Please note most encoding systems share the same code with ASCII for the 7 bit characters. But there are some exceptions. If you are converting old Japanese C programs and URLs data from the casually-called shift-JIS encoding format to UTF-8 format, use "CP932" as the encoding name instead of "shift - JIS" to get the expected results: 0x5C → "\" and 0x7E → "~". Otherwise, these are converted to wrong characters.

Indicație

recode(1) may be used too and offers more than the combined functionality of iconv(1), fromdos(1), todos(1), frommac(1), and tomac(1). For more, see "info recode".

11.1.2 Checking file to be UTF-8 with iconv

You can check if a text file is encoded in UTF-8 with iconv(1) by the following.

```
$ iconv -f utf8 -t utf8 input.txt >/dev/null || echo "non-UTF-8 found"
```

Indicație

Use "- -verbose" option in the above example to find the first non-UTF-8 character.

11.1.3 Converting file names with iconv

Here is an example script to convert encoding of file names from ones created under older OS to modern UTF-8 ones in a single directory.

```
#!/bin/sh
ENCDN=iso-8859-1
for x in *;
do
  mv "$x" "$(echo "$x" | iconv -f $ENCDN -t utf-8)"
done
```

The "\$ENCDN" variable specifies the original encoding used for file names under older OS as in Tabel [11.2](#).

For more complicated case, please mount a filesystem (e.g. a partition on a disk drive) containing such file names with proper encoding as the mount(8) option (see Secțiune [8.1.3](#)) and copy its entire contents to another filesystem mounted as UTF-8 with "cp -a" command.

11.1.4 EOL conversion

The text file format, specifically the end-of-line (EOL) code, is dependent on the platform.

platforma	EOL code	control	zecimal	hexazecimal
Debian (unix)	LF	^J	10	0A
MSDOS și Windows	CR-LF	^M^J	13 10	0D 0A
Apple's Macintosh	CR	^M	13	0D

Tabela 11.3: List of EOL styles for different platforms

The EOL format conversion programs, fromdos(1), todos(1), frommac(1), and tomac(1), are quite handy. recode(1) is also useful.

Notă

Some data on the Debian system, such as the wiki page data for the python-moinmoin package, use MSDOS style CR-LF as the EOL code. So the above rule is just a general rule.

Notă

Most editors (eg. vim, emacs, gedit, ...) can handle files in MSDOS style EOL transparently.

Indicație

The use of "sed -e '/\r\$/!s/\$/\r/'" instead of todos(1) is better when you want to unify the EOL style to the MSDOS style from the mixed MSDOS and Unix style. (e.g., after merging 2 MSDOS style files with diff3(1).) This is because todos adds CR to all lines.

11.1.5 TAB conversion

There are few popular specialized programs to convert the tab codes.

funcție	bsdmainutils	coreutils
expand tab to spaces	"col -x"	expand
unexpand tab from spaces	"col -h"	unexpand

Tabela 11.4: List of TAB conversion commands from bsdmainutils and coreutils packages

indent(1) from the indent package completely reformats whitespaces in the C program.

Editor programs such as vim and emacs can be used for TAB conversion, too. For example with vim, you can expand TAB with ":set expandtab" and ":%retab" command sequence. You can revert this with ":set noexpandtab" and ":%retab!" command sequence.

11.1.6 Editors with auto-conversion

Intelligent modern editors such as the `vim` program are quite smart and copes well with any encoding systems and any file formats. You should use these editors under the UTF-8 locale in the UTF-8 capable console for the best compatibility.

An old western European Unix text file, "`u-file.txt`", stored in the `latin1` (`iso-8859-1`) encoding can be edited simply with `vim` by the following.

```
$ vim u-file.txt
```

This is possible since the auto detection mechanism of the file encoding in `vim` assumes the UTF-8 encoding first and, if it fails, assumes it to be `latin1`.

An old Polish Unix text file, "`pu-file.txt`", stored in the `latin2` (`iso-8859-2`) encoding can be edited with `vim` by the following.

```
$ vim '+e ++enc=latin2 pu-file.txt'
```

An old Japanese unix text file, "`ju-file.txt`", stored in the `eucJP` encoding can be edited with `vim` by the following.

```
$ vim '+e ++enc=eucJP ju-file.txt'
```

An old Japanese MS-Windows text file, "`jw-file.txt`", stored in the so called `shift-JIS` encoding (more precisely: `CP932`) can be edited with `vim` by the following.

```
$ vim '+e ++enc=CP932 ++ff=dos jw-file.txt'
```

When a file is opened with "`++enc`" and "`++ff`" options, "`:w`" in the Vim command line stores it in the original format and overwrite the original file. You can also specify the saving format and the file name in the Vim command line, e.g., "`:w ++enc=utf8 new.txt`".

Please refer to the `mbyte.txt` "multi-byte text support" in `vim` on-line help and Tabel [11.2](#) for locale values used with "`++enc`".

The `emacs` family of programs can perform the equivalent functions.

11.1.7 Plain text extraction

The following reads a web page into a text file. This is very useful when copying configurations off the Web or applying basic Unix text tools such as `grep(1)` on the web page.

```
$ w3m -dump https://www.remote-site.com/help-info.html >textfile
```

Similarly, you can extract plain text data from other formats using the following.

11.1.8 Highlighting and formatting plain text data

You can highlight and format plain text data by the following.

11.2 XML data

The [Extensible Markup Language \(XML\)](#) is a markup language for documents containing structured information.

See introductory information at [XML.COM](#).

pachet	popcon(popularitete)	dimensiune	keyword	funcție
w3m	V:11, I:151	2853	html → text	HTML to text converter with the "w3m -dump" command
html2text	V:4, I:71	298	html → text	advanced HTML to text converter (ISO 8859-1)
lynx	V:27, I:450	1972	html → text	HTML to text converter with the "lynx -dump" command
elinks	V:2, I:17	1755	html → text	HTML to text converter with the "elinks -dump" command
links	V:2, I:22	2321	html → text	HTML to text converter with the "links -dump" command
links2	V:0, I:11	5466	html → text	HTML to text converter with the "links2 -dump" command
catdoc	V:15, I:176	682	MSWord → text	Convert MSWord files to plain text or TeX
antiword	V:0, I:7	587	MSWord → text	convert MSWord files to plain text or ps
unhtml	V:0, I:0	40	html → text	remove the markup tags from an HTML file
odt2txt	V:1, I:27	60	odt → text	converter from OpenDocument Text to text

Tabela 11.5: List of tools to extract plain text data

pachet	popcon(popularitete)	dimensiune	keyword	descriere
vim-runtime	V:17, I:373	37551	highlight	Vim MACRO to convert source code to HTML with " :source \$VIMRUNTIME/syntax/html.vim"
cxref	V:0, I:0	1191	c → html	converter for the C program to latex and HTML (C language)
src2tex	V:0, I:0	1799	highlight	convert many source codes to TeX (C language)
source-highlight	V:0, I:3	2127	highlight	convert many source codes to HTML, XHTML, LaTeX, Texinfo, ANSI color escape sequences and DocBook files with highlight (C++)
highlight	V:0, I:3	1412	highlight	convert many source codes to HTML, XHTML, RTF, LaTeX, TeX or XSL-FO files with highlight (C++)
grc	V:0, I:6	208	text → culoare	generic colouriser for everything (Python)
pandoc	V:10, I:48	193946	text → orice	general markup converter (Haskell)
python3-docutils	V:13, I:53	2009	text → orice	ReStructured Text document formatter to XML (Python)
markdown	V:0, I:7	56	text → html	Markdown text document formatter to (X)HTML (Perl)
asciidoc	V:0, I:5	101	text → orice	AsciiDoc text document formatter to XML/HTML (Ruby)
python3-sphinx	V:7, I:27	2996	text → orice	ReStructured Text based document publication system (Python)
hugo	V:0, I:5	61446	text → html	Markdown based static site publication system (Go)

Tabela 11.6: List of tools to highlight plain text data

- ["What is XML?"](#)
- ["What Is XSLT?"](#)
- ["What Is XSL-FO?"](#)
- ["What Is XLink?"](#)

11.2.1 Basic hints for XML

XML text looks somewhat like [HTML](#). It enables us to manage multiple formats of output for a document. One easy XML system is the docbook-xsl package, which is used here.

Each XML file starts with standard XML declaration as the following.

```
<?xml version="1.0" encoding="UTF-8"?>
```

The basic syntax for one XML element is marked up as the following.

```
<name attribute="value">content</name>
```

XML element with empty content is marked up in the following short form.

```
<name attribute="value" />
```

The "attribute="value"" in the above examples are optional.

The comment section in XML is marked up as the following.

```
<!-- comment -->
```

Other than adding markups, XML requires minor conversion to the content using predefined entities for following characters.

predefined entity	character to be converted into
"	" : quote
'	' : apostrophe
<	< : less-than
>	> : greater-than
&	& : ampersand

Tabela 11.7: List of predefined entities for XML



Atenție

"<" or "&" can not be used in attributes or elements.

Notă

When SGML style user defined entities, e.g. "&some-tag;", are used, the first definition wins over others. The entity definition is expressed in "<!ENTITY some-tag "entity value">".

Notă

As long as the XML markup are done consistently with certain set of the tag name (either some data as content or attribute value), conversion to another XML is trivial task using [Extensible Stylesheet Language Transformations \(XSLT\)](#).

11.2.2 XML processing

There are many tools available to process XML files such as [the Extensible Stylesheet Language \(XSL\)](#).

Basically, once you create well formed XML file, you can convert it to any format using [Extensible Stylesheet Language Transformations \(XSLT\)](#).

The [Extensible Stylesheet Language for Formatting Objects \(XSL-FO\)](#) is supposed to be solution for formatting. The fop package is new to the Debian main archive due to its dependence to the [Java programming language](#). So the LaTeX code is usually generated from XML using XSLT and the LaTeX system is used to create printable file such as DVI, PostScript, and PDF.

pachet	popcon(popularity)	limitate	keyword	descriere
docbook-xml	I:420	2126	xml	XML document type definition (DTD) for DocBook
docbook-xsl	V:14, I:152	14823	xml/xslt	XSL stylesheets for processing DocBook XML to various output formats with XSLT
xsltproc	V:16, I:77	83	xslt	XSLT command line processor (XML → XML, HTML, plain text, etc.)
xmlto	V:0, I:10	124	xml/xslt	XML-to-any converter with XSLT
fop	V:0, I:9	281	xml/xsl-fo	convert Docbook XML files to PDF
dblatex	V:1, I:6	4636	xml/xslt	convert Docbook files to DVI, PostScript, PDF documents with XSLT
dbtoepub	V:0, I:0	37	xml/xslt	DocBook XML to .epub converter

Tabela 11.8: List of XML tools

Since XML is subset of [Standard Generalized Markup Language \(SGML\)](#), it can be processed by the extensive tools available for SGML, such as [Document Style Semantics and Specification Language \(DSSSL\)](#).

pachet	popcon(popularity)	limitate	keyword	descriere
openjade	V:1, I:23	1066	dsssl	ISO/IEC 10179:1996 standard DSSSL processor (latest)
docbook-dsssl	V:0, I:9	2594	xml/dsssl	DSSSL stylesheets for processing DocBook XML to various output formats with DSSSL
docbook-utils	V:0, I:6	287	xml/dsssl	utilities for DocBook files including conversion to other formats (HTML, RTF, PS, man, PDF) with docbook2* commands with DSSSL

Tabela 11.9: List of DSSSL tools

Indicație

[GNOME](#)'s `ye1p` is sometimes handy to read [DocBook](#) XML files directly since it renders decently on X.

11.2.3 The XML data extraction

You can extract HTML or XML data from other formats using followings.

11.2.4 The XML data lint

For non-XML HTML files, you can convert them to XHTML which is an instance of well formed XML. XHTML can be processed by XML tools.

Syntax of XML files and goodness of URLs found in them may be checked.

Once proper XML is generated, you can use XSLT technology to extract data based on the mark-up context etc.

pachet	popcon(popularitate)	dimensiune	keyword	descriere
man2html	V:0, I:1	142	manpage → html	converter from manpage to HTML (CGI support)
doclifter	V:0, I:0	473	troff → xml	converter from troff to DocBook XML
texi2html	V:0, I:3	1847	texi → html	converter from Texinfo to HTML
info2www	V:0, I:1	74	info → html	converter from GNU info to HTML (CGI support)
wv	V:0, I:2	733	MSWord → orice	document converter from Microsoft Word to HTML, LaTeX, etc.
unrtf	V:0, I:3	159	rtf → html	document converter from RTF to HTML, etc
wp2x	V:0, I:0	200	WordPerfect → orice	WordPerfect 5.0 and 5.1 files to TeX, LaTeX, troff, GML and HTML

Tabela 11.10: List of XML data extraction tools

pachet	popcon(popularitate)	dimensiune	funcție	descriere
libxml2-utils	V:62, I:217	205	xml ↔ html ↔ xhtml	command line XML tool with xmllint(1) (syntax check, reformat, lint, ...)
tidy	V:1, I:8	79	xml ↔ html ↔ xhtml	HTML syntax checker and reformatter
weblint-perl	V:0, I:1	32	lint	syntax and minimal style checker for HTML
linklint	V:0, I:0	343	link check	fast link checker and web site maintenance tool

Tabela 11.11: List of XML pretty print tools

11.3 Type setting

The Unix [troff](#) program originally developed by AT&T can be used for simple typesetting. It is usually used to create manpages.

[TeX](#) created by Donald Knuth is a very powerful type setting tool and is the de facto standard. [LaTeX](#) originally written by Leslie Lamport enables a high-level access to the power of TeX.

pachet	popcon(popularitate)	dimensiune	keyword	descriere
texlive	I:30	56	(La)TeX	TeX system for typesetting, previewing and printing
groff	V:1, I:27	20577	troff	GNU troff text-formatting system

Tabela 11.12: List of type setting tools

11.3.1 roff typesetting

Traditionally, [roff](#) is the main Unix text processing system. See [roff\(7\)](#), [groff\(7\)](#), [groff\(1\)](#), [grotty\(1\)](#), [troff\(1\)](#), [groff_mdoc\(7\)](#), [groff_man\(7\)](#), [groff_ms\(7\)](#), [groff_me\(7\)](#), [groff_mm\(7\)](#), and "info groff".

You can read or print a good tutorial and reference on "-me" [macro](#) in "/usr/share/doc/groff/" by installing the groff package.

Indicație

"groff -Tascii -me -" produces plain text output with [ANSI escape code](#). If you wish to get manpage like output with many "^H" and "_", use "GROFF_NO_SGR=1 groff -Tascii -me -" instead.

Indicație

To remove "^H" and "_" from a text file generated by groff, filter it by "col -b -x".

11.3.2 TeX/LaTeX

The [TeX Live](#) software distribution offers a complete TeX system. The `texlive` metapackage provides a decent selection of the [TeX Live](#) packages which should suffice for the most common tasks.

There are many references available for [TeX](#) and [LaTeX](#).

- [The teTeX HOWTO: The Linux-teTeX Local Guide](#)
- `tex(1)`
- `latex(1)`
- `texdoc(1)`
- `texdoctk(1)`
- "The TeXbook", by Donald E. Knuth, (Addison-Wesley)
- "LaTeX - A Document Preparation System", by Leslie Lamport, (Addison-Wesley)
- "The LaTeX Companion", by Goossens, Mittelbach, Samarin, (Addison-Wesley)

This is the most powerful typesetting environment. Many [SGML](#) processors use this as their back end text processor. [Lyx](#) provided by the `lyx` package and [GNU TeXmacs](#) provided by the `texmacs` package offer nice [WYSIWYG](#) editing environment for [LaTeX](#) while many use [Emacs](#) and [Vim](#) as the choice for the source editor.

There are many online resources available.

- The TEX Live Guide - TEX Live 2007 ("`/usr/share/doc/texlive-doc-base/english/texlive-en/live.html`") (`texlive-doc-base` package)
- [A Simple Guide to Latex/Lyx](#)
- [Word Processing Using LaTeX](#)

When documents become bigger, sometimes TeX may cause errors. You must increase pool size in "`/etc/texmf/texmf.d`" (or more appropriately edit "`/etc/texmf/texmf.d/95NonPath`" and run `update-texmf(8)`) to fix this.

Notă

The TeX source of "The TeXbook" is available at www.ctan.org/tex-archive/site/texbook.tex. This file contains most of the required macros. I heard that you can process this document with `tex(1)` after commenting lines 7 to 10 and adding "`\input manmac \proofmodefalse`". It's strongly recommended to buy this book (and all other books from Donald E. Knuth) instead of using the online version but the source is a great example of TeX input!

11.3.3 Pretty print a manual page

You can print a manual page in PostScript nicely by one of the following commands.

```
$ man -Tps some_manpage | lpr
```

11.3.4 Creating a manual page

Although writing a manual page (`manpage`) in the plain [troff](#) format is possible, there are few helper packages to create it.

pachet	popcon(popularity)	dimensiune	keyword	descriere
docbook-to-man	V:0, I:6	189	SGML → manpage	converter from DocBook SGML into roff man macros
help2man	V:0, I:6	542	text → manpage	automatic manpage generator from --help
info2man	V:0, I:0	134	info → manpage	converter from GNU info to POD or man pages
txt2man	V:0, I:0	112	text → manpage	convert flat ASCII text to man page format

Tabela 11.13: List of packages to help creating the manpage

11.4 Printable data

Printable data is expressed in the [PostScript](#) format on the Debian system. [Common Unix Printing System \(CUPS\)](#) uses Ghostscript as its rasterizer backend program for non-PostScript printers.

Printable data may also be expressed in the [PDF](#) format on the recent Debian system.

PDF files can displayed and its form entries may be filled using GUI viewer tools such as [Evince](#) and [Okular](#) (see Secțiune 7.4); and modern browsers such as [Chromium](#).

PDF files can be edited using some graphics tools such as [LibreOffice](#), [Scribus](#), and [Inkscape](#) (see Secțiune 11.6).

Indicație

You can read a PDF file with [GIMP](#) and convert it into [PNG](#) format using higher than 300 dpi resolution. This may be used as a background image for [LibreOffice](#) to produce a desirable altered printout with minimum efforts.

11.4.1 Ghostscript

The core of printable data manipulation is the [Ghostscript PostScript \(PS\)](#) interpreter which generates raster image.

pachet	popcon(popularity)	dimensiune	descriere
ghostscript	V:149, I:578	183	The GPL Ghostscript PostScript/PDF interpreter
ghostscript-x	I:19	87	GPL Ghostscript PostScript/PDF interpreter - X display support
libpoppler147	V:149, I:250	4891	PDF rendering library forked from the xpdf PDF viewer
libpoppler-glib8t64	V:134, I:246	550	PDF rendering library (GLib-based shared library)
poppler-data	V:161, I:599	13086	CMaps for PDF rendering library (for CJK support: Adobe-*)

Tabela 11.14: List of Ghostscript PostScript interpreters

Indicație

"gs -h" can display the configuration of Ghostscript.

11.4.2 Merge two PS or PDF files

You can merge two [PostScript \(PS\)](#) or [Portable Document Format \(PDF\)](#) files using gs(1) of Ghostscript.

```
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pswrite -sOutputFile=bla.ps -f foo1.ps foo2.ps
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -sOutputFile=bla.pdf -f foo1.pdf foo2.pdf
```

Notă

The [PDF](#), which is a widely used cross-platform printable data format, is essentially the compressed [PS](#) format with few additional features and extensions.

Indicație

For command line, `psmerge(1)` and other commands from the `psutils` package are useful for manipulating PostScript documents. `pdftk(1)` from the `pdftk` package is useful for manipulating PDF documents, too.

11.4.3 Printable data utilities

The following packages for the printable data utilities caught my eyes.

pachet	popcon(popularity)	dimensiune	keyword	descriere
poppler-utils	V:131, I:485	760	pdf → ps,text,...	PDF utilities: <code>pdftops</code> , <code>pdfinfo</code> , <code>pdfimages</code> , <code>pdftotext</code> , <code>pdffonts</code>
psutils	V:3, I:55	34	ps → ps	PostScript document conversion tools
poster	V:0, I:1	58	ps → ps	create large posters out of PostScript pages
enscript	V:0, I:11	2138	text → ps,html, rtf	convert ASCII text to PostScript, HTML, RTF or Pretty-Print
a2ps	V:0, I:7	4083	text → ps	'Anything to PostScript' converter and pretty-printer
pdftk	I:27	28	pdf → pdf	PDF document conversion tool: <code>pdftk</code>
html2ps	V:0, I:2	256	html → ps	converter from HTML to PostScript
gnuhtml2latex	V:0, I:0	26	html → latex	converter from html to latex
latex2rtf	V:0, I:2	495	latex → rtf	convert documents from LaTeX to RTF which can be read by MS Word
ps2eps	V:1, I:35	95	ps → eps	converter from PostScript to EPS (Encapsulated PostScript)
e2ps	V:0, I:0	104	text → ps	Text to PostScript converter with Japanese encoding support
impose+	V:0, I:1	118	ps → ps	PostScript utilities
trueprint	V:0, I:0	148	text → ps	pretty print many source codes (C, C++, Java, Pascal, Perl, Pike, Sh, and Verilog) to PostScript. (C language)
pdf2svg	V:0, I:3	33	pdf → svg	converter from PDF to Scalable vector graphics format
pdftoipe	V:0, I:0	74	pdf → ipe	converter from PDF to IPE's XML format

Tabela 11.15: List of printable data utilities

11.4.4 Imprimarea cu CUPS

Both `lp(1)` and `lpr(1)` commands offered by [Common Unix Printing System \(CUPS\)](#) provides options for customized printing the printable data.

You can print 3 copies of a file collated using one of the following commands.

```
$ lp -n 3 -o Collate=True filename
```

```
$ lpr -#3 -o Collate=True filename
```


You can further customize printer operation by using printer option such as "-o number-up=2", "-o page-set=even", "-o page-set=odd", "-o scaling=200", "-o natural-scaling=200", etc., documented at [Command-Line Printing and Options](#).

11.5 The mail data conversion

The following packages for the mail data conversion caught my eyes.

pachet	popcon(popularity)	limitare	keyword	descriere
sharutils	V:2, I:31	1436	mail	shar(1), unshar(1), uuencode(1), uudecode(1)
mpack	V:0, I:9	109	MIME	encoding and decoding of MIME messages: mpack(1) and munpack(1)
tnef	V:0, I:4	103	ms-tnef	unpacking MIME attachments of type "application/ms-tnef" which is a Microsoft only format
uudeview	V:0, I:2	105	mail	encoder and decoder for the following formats: uuencode , xxencode , BASE64 , quoted printable , and BinHex

Tabela 11.16: List of packages to help mail data conversion

Indicație

The [Internet Message Access Protocol](#) version 4 (IMAP4) server may be used to move mails out from proprietary mail systems if the mail client software can be configured to use IMAP4 server too.

11.5.1 Mail data basics

Mail ([SMTP](#)) data should be limited to series of 7 bit data. So binary data and 8 bit text data are encoded into 7 bit format with the [Multipurpose Internet Mail Extensions \(MIME\)](#) and the selection of the charset (see Tabel 11.2).

The standard mail storage format is mbox formatted according to [RFC2822 \(updated RFC822\)](#). See mbox(5) (provided by the `mutt` package).

For European languages, "Content-Transfer-Encoding: quoted-printable" with the ISO-8859-1 charset is usually used for mail since there are not much 8 bit characters. If European text is encoded in UTF-8, "Content-Transfer-Encoding: quoted-printable" is likely to be used since it is mostly 7 bit data.

For Japanese, traditionally "Content-Type: text/plain; charset=ISO-2022-JP" is usually used for mail to keep text in 7 bits. But older Microsoft systems may send mail data in Shift-JIS without proper declaration. If Japanese text is encoded in UTF-8, [Base64](#) is likely to be used since it contains many 8 bit data. The situation of other Asian languages is similar.

Notă

If your non-Unix mail data is accessible by a non-Debian client software which can talk to the IMAP4 server, you may be able to move them out by running your own IMAP4 server.

Notă

If you use other mail storage formats, moving them to mbox format is the good first step. The versatile client program such as `mutt(1)` may be handy for this.

You can split mailbox contents to each message using `procmail(1)` and `formail(1)`.

Each mail message can be unpacked using `munpack(1)` from the `mpack` package (or other specialized tools) to obtain the MIME encoded contents.

11.6 Graphic data tools

Although GUI programs such as `gimp(1)` are very powerful, command line tools such as `imagemagick(1)` are quite useful for automating image manipulation via scripts.

The de facto image file format of the digital camera is the [Exchangeable Image File Format](#) (EXIF) which is the [JPEG](#) image file format with additional metadata tags. It can hold information such as date, time, and camera settings.

The [Lempel-Ziv-Welch \(LZW\) lossless data compression](#) patent has been expired. [Graphics Interchange Format \(GIF\)](#) utilities which use the LZW compression method are now freely available on the Debian system.

Indicație

Any digital camera or scanner with removable recording media works with Linux through [USB storage](#) readers since it follows the [Design rule for Camera Filesystem](#) and uses [FAT](#) filesystem. See Secțiune [10.1.7](#).

11.6.1 Graphic data tools (metapackage)

The following metapackages are good starting points for searching graphics data tools using `aptitude(8)`. "[Packages overview for Debian PhotoTools Maintainers](#)" can be another starting point.

pachet	popcon(popularity)	limite(limit)	keyword	descriere
education-graphics	1.0	31	svg, jpeg, ...	metapackage for teaching graphics and pictural art.
open-font-design-toolkit	1.0	9	ttf, ps, ...	metapackage for open font design

Tabela 11.17: List of graphics data tools (metapackage)

Indicație

Search more image tools using regex `"~Gworks-with::image"` in `aptitude(8)` (see Secțiune [2.2.6](#)).

11.6.2 Graphic data tools (GUI)

The following packages for the GUI graphics data conversion, editing, and organization tools caught my eyes.

11.6.3 Graphic data tools (CLI)

The following packages for the CLI graphics data conversion, editing, and organization tools caught my eyes.

pachet	popcon(popularity)	limita	keyword	descriere
gimp	V:45, I:232	32135	image(bitmap)	GNU Image Manipulation Program
xsane	V:10, I:135	1512	image(bitmap)	GTK-based X11 frontend for SANE (Scanner Access Now Easy)
scribus	V:1, I:14	32052	ps/pdf/SVG/...	Scribus DTP editor
libreoffice-draw	V:92, I:434	10985	image(vector)	LibreOffice office suite - drawing
inkscape	V:12, I:86	113183	image(vector)	SVG (Scalable Vector Graphics) editor
dia	V:1, I:19	4086	image(vector)	diagram editor (Gtk)
xfig	V:0, I:9	8511	image(vector)	Facility for Interactive Generation of figures under X11
gocr	V:0, I:4	549	image → text	free OCR software
eog	V:35, I:176	10310	image(Exif)	Eye of GNOME graphics viewer program
gthumb	V:3, I:13	5152	image(Exif)	image viewer and browser (GNOME)
geeqie	V:3, I:12	2982	image(Exif)	image viewer using GTK
shotwell	V:15, I:255	6334	image(Exif)	digital photo organizer (GNOME)
gwenview	V:39, I:119	5988	image(Exif)	image viewer (KDE)
kamera	I:118	982	image(Exif)	digital camera support for KDE applications
digikam	V:1, I:9	302	image(Exif)	digital photo management application for KDE
darktable	V:4, I:12	35873	image(Exif)	virtual lighttable and darkroom for photographers
hugin	V:0, I:6	6489	image(Exif)	panorama photo stitcher
librecad	V:1, I:15	9100	DXF, ...	2D CAD data editor
freecad	I:20	110	DXF, ...	3D CAD data editor
blender	V:3, I:24	92911	blend, TIFF, VRML, ...	3D content editor for animation etc
mm3d	V:0, I:0	4123	ms3d, obj, dxf, ...	OpenGL based 3D model editor
fontforge	V:0, I:6	4058	ttf, ps, ...	font editor for PS, TrueType and OpenType fonts
xgridfit	V:0, I:0	878	ttf	program for gridfitting and hinting TrueType fonts

Tabela 11.18: List of graphics data tools (GUI)

pachet	popcon(popular) contribute	limitate	keyword	descriere
imagemagick	V:295	77	image(bitmap)	image manipulation programs
graphicsmagick	V:1, I:9	5816	image(bitmap)	image manipulation programs (fork of imagemagick)
netpbm	V:27, I:305	8433	image(bitmap)	graphics conversion tools
libheif-examples	V:0, I:3	412	heif → jpeg(bitmap)	convert High Efficiency Image File Format (HEIF) to JPEG, PNG, or Y4M formats with heif-converter(1) command
icoutils	V:4, I:36	221	png ↔ ico(bitmap)	convert MS Windows icons and cursors to and from PNG formats (favicon.ico)
pstoedit	V:1, I:43	1076	ps/pdf → image(vector)	PostScript and PDF files to editable vector graphics converter (SVG)
libwmf-bin	V:4, I:92	151	Windows/image(vector)	Windows metafile (vector graphics data) conversion tools
fig2sxd	V:0, I:0	151	fig → sxd(vector)	convert XFig files to OpenOffice.org Draw format
unpaper	V:1, I:17	417	image → image	post-processing tool for scanned pages for OCR
tesseract-ocr	V:7, I:33	2279	image → text	free OCR software based on the HP's commercial OCR engine
tesseract-ocr-eng	V:7, I:33	4032	image → text	OCR engine data: tesseract-ocr language files for English text
ocrad	V:0, I:2	599	image → text	free OCR software
exif	V:2, I:53	335	image(Exif)	command-line utility to show EXIF information in JPEG files
exiv2	V:1, I:21	427	image(Exif)	EXIF/IPTC metadata manipulation tool
exiftran	V:0, I:12	81	image(Exif)	transform digital camera jpeg images
exiftags	V:0, I:3	309	image(Exif)	utility to read Exif tags from a digital camera JPEG file
exifprobe	V:0, I:2	502	image(Exif)	read metadata from digital pictures
dcraw	V:0, I:8	428	image(Raw) → png	decode raw digital camera images
findimagedupes	V:0, I:1	76	image → fingerprint	find visually similar or duplicate images
ale	V:0, I:0	818	image → image	merge images to increase fidelity or create mosaics
imageindex	V:0, I:1	143	image(Exif) → html	generate static HTML galleries from images
outguess	V:0, I:1	230	jpeg.png	universal Steganographic tool
jpegoptim	V:0, I:6	59	jpeg	optimize JPEG files
optipng	V:2, I:44	187	png	optimize PNG files, lossless compression
pngquant	V:1, I:10	62	png	optimize PNG files, lossy compression

Tabela 11.19: List of graphics data tools (CLI)

pachet	popcon(popular) contribute	limitate	keyword	descriere
alien	V:1, I:14	150	rpm/tgz → deb	converter for the foreign package into the Debian package
freepwing	V:0, I:0	447	EB → EPWING	converter from "Electric Book" (popular in Japan) to a single JIS X 4081 format (a subset of the EPWING V1)
calibre	V:7, I:27	65174	orice → EPUB	e-book converter and library management

Tabela 11.20: List of miscellaneous data conversion tools

11.7 Miscellaneous data conversion

There are many other programs for converting data. Following packages caught my eyes using regex "~Guse : : converting in aptitude(8) (see Secțiune [2.2.6](#)).

You can also extract data from RPM format with the following.

```
$ rpm2cpio file.src.rpm | cpio --extract
```

Capitolul 12

Programare

I provide some pointers for people to learn programming on the Debian system enough to trace the packaged source code. Here are notable packages and corresponding documentation packages for programming.

Online references are available by typing "man name" after installing manpages and manpages-dev packages. Online references for the GNU tools are available by typing "info program_name" after installing the pertinent documentation packages. You may need to include the contrib and non-free archives in addition to the main archive since some GFDL documentations are not considered to be DFSG compliant.

Please consider to use version control system tools. See Secțiune [10.5](#).



Avertisment

Do not use "test" as the name of an executable test file. "test" is a shell builtin.



Atenție

You should install software programs directly compiled from source into "/usr/local" or "/opt" to avoid collision with system programs.

Indicație

[Code examples of creating "Song 99 Bottles of Beer"](#) should give you good ideas of practically all the programming languages.

12.1 The shell script

The [shell script](#) is a text file with the execution bit set and contains the commands in the following format.

```
#!/bin/sh
... command lines
```

The first line specifies the shell interpreter which read and execute this file contents.

Reading shell scripts is the **best** way to understand how a Unix-like system works. Here, I give some pointers and reminders for shell programming. See "Shell Mistakes" (<https://www.greenend.org.uk/rjk/2001/04/shell.html>) to learn from mistakes.

Unlike shell interactive mode (see Secțiune [1.5](#) and Secțiune [1.6](#)), shell scripts frequently use parameters, conditionals, and loops.

12.1.1 POSIX shell compatibility

Many system scripts may be interpreted by any one of [POSIX](#) shells (see Tabel [1.13](#)).

- The default non-interactive POSIX shell `/usr/bin/sh` is a symlink pointing to `/usr/bin/dash` and used by many system programs.
- The default interactive POSIX shell is `/usr/bin/bash`.

Avoid writing a shell script with **bashisms** or **zshisms** to make it portable among all POSIX shells. You can check it using `checkbashisms(1)`.

Good: POSIX	Avoid: bashism
<code>if ["\$foo" = "\$bar"] ; then ...</code>	<code>if ["\$foo" == "\$bar"] ; then ...</code>
<code>diff -u file.c.orig file.c</code>	<code>diff -u file.c{.orig,}</code>
<code>mkdir /foobar /foobaz</code>	<code>mkdir /foo{bar,baz}</code>
<code>funcname() { ... }</code>	<code>function funcname() { ... }</code>
octal format: <code>"\377"</code>	hexadecimal format: <code>"\xff"</code>

Tabela 12.1: List of typical bashisms

The `"echo"` command must be used with following cares since its implementation differs among shell builtin and external commands.

- Avoid using any command options except `"-n"`.
- Avoid using escape sequences in the string since their handling varies.

Notă

Although `"-n"` option is **not** really POSIX syntax, it is generally accepted.

Indicație

Use the `"printf"` command instead of the `"echo"` command if you need to embed escape sequences in the output string.

12.1.2 Shell parameters

Special shell parameters are frequently used in the shell script.

Basic **parameter expansions** to remember are as follows.

Here, the colon `":"` in all of these operators is actually optional.

- **with** `":"` = operator test for **exist** and **not null**
 - **without** `":"` = operator test for **exist** only
-

parametru shell	valoare
\$0	name of the shell or shell script
\$1	first (1st) shell argument
\$9	ninth (9th) shell argument
\$#	number of positional parameters
"\$*"	"\$1 \$2 \$3 \$4 ... "
"\$@"	"\$1" "\$2" "\$3" "\$4" ...
\$?	exit status of the most recent command
\$\$	PID of this shell script
\$_	PID of most recently started background job

Tabela 12.2: List of shell parameters

parameter expression form	value if var is set	value if var is not set
\${var:-string}	"\$var"	"string"
\${var:+string}	"string"	„null"
\${var:=string}	"\$var"	"string" (and run "var=string")
\${var:?string}	"\$var"	echo "string" to stderr (and exit with error)

Tabela 12.3: List of shell parameter expansions

parameter substitution form	rezultatul
\${var%suffix}	remove smallest suffix pattern
\${var%%suffix}	remove largest suffix pattern
\${var#prefix}	remove smallest prefix pattern
\${var##prefix}	remove largest prefix pattern

Tabela 12.4: List of key shell parameter substitutions

12.1.3 Condiționale shell

Each command returns an **exit status** which can be used for conditional expressions.

- Success: 0 ("True")
- Error: non 0 ("False")

Notă

"0" in the shell conditional context means "True", while "0" in the C conditional context means "False".

Notă

"[" is the equivalent of the `test` command, which evaluates its arguments up to "]" as a conditional expression.

Basic **conditional idioms** to remember are the following.

- `"command && if_success_run_this_command_too || true"`
- `"command || if_not_success_run_this_command_too || true"`
- A multi-line script snippet as the following

```
if [ conditional_expression ]; then
    if_success_run_this_command
else
    if_not_success_run_this_command
fi
```

Here trailing `"|| true"` was needed to ensure this shell script does not exit at this line accidentally when shell is invoked with `"-e"` flag.

equation	condition to return logical true
<code>-e fișier</code>	<i>file</i> exists
<code>-d fișier</code>	<i>file</i> exists and is a directory
<code>-f fișier</code>	<i>file</i> exists and is a regular file
<code>-w fișier</code>	<i>file</i> exists and is writable
<code>-x fișier</code>	<i>file</i> exists and is executable
<code>fișier1 -nt fișier2</code>	<i>file1</i> is newer than <i>file2</i> (modification)
<code>fișier1 -ot fișier2</code>	<i>file1</i> is older than <i>file2</i> (modification)
<code>fișier1 -ef fișier2</code>	<i>file1</i> and <i>file2</i> are on the same device and the same inode number

Tabela 12.5: List of file comparison operators in the conditional expression

Arithmetic integer comparison operators in the conditional expression are `"-eq"`, `"-ne"`, `"-lt"`, `"-le"`, `"-gt"`, and `"-ge"`.

12.1.4 Bucle shell

There are several loop idioms to use in POSIX shell.

- `"for x in foo1 foo2 ... ; do command ; done"` loops by assigning items from the list `"foo1 foo2 ..."` to variable `"x"` and executing `"command"`.
-

equation	condition to return logical true
<code>-z <i>str</i></code>	the length of <i>str</i> is zero
<code>-n <i>str</i></code>	the length of <i>str</i> is non-zero
<code><i>str1</i> = <i>str2</i></code>	<i>str1</i> and <i>str2</i> are equal
<code><i>str1</i> != <i>str2</i></code>	<i>str1</i> and <i>str2</i> are not equal
<code><i>str1</i> < <i>str2</i></code>	<i>str1</i> sorts before <i>str2</i> (locale dependent)
<code><i>str1</i> > <i>str2</i></code>	<i>str1</i> sorts after <i>str2</i> (locale dependent)

Tabela 12.6: List of string comparison operators in the conditional expression

- "while condition ; do command ; done" repeats "command" while "condition" is true.
- "until condition ; do command ; done" repeats "command" while "condition" is not true.
- "break" enables to exit from the loop.
- "continue" enables to resume the next iteration of the loop.

Indicație

The C-language like numeric iteration can be realized by using `seq(1)` as the "foo1 foo2 ..." generator.

Indicație

See Secțiune [9.4.9](#).

12.1.5 Shell environment variables

Some popular environment variables for the normal shell command prompt may not be available under the execution environment of your script.

- For "\$USER", use "\$(id -un)"
- For "\$UID", use "\$(id -u)"
- For "\$HOME", use "\$(getent passwd "\$(id -u)" | cut -d ':' -f 6)" (this works also on Secțiune [4.5.2](#))

12.1.6 The shell command-line processing sequence

The shell processes a script roughly as the following sequence.

- The shell reads a line.
- The shell groups a part of the line as **one token** if it is within "\"" or '... '.
- The shell splits other part of a line into **tokens** by the following.
 - Spații-albe: *spațiu tabulator linie-nouă*
 - Metacaractere: < > | ; & ()
- The shell checks the **reserved word** for each token to adjust its behavior if not within "\"" or '... '.
 - **reserved word**: if then elif else fi for in while unless do done case esac
- The shell expands **alias** if not within "\"" or '... '.

- The shell expands **tilde** if not within `"..."` or `'...'`.
 - `"~"` → current user's home directory
 - `"~user"` → *user*'s home directory
- The shell expands **parameter** to its value if not within `'...'`.
 - **parameter**: `"$PARAMETER"` or `"${PARAMETER}"`
- The shell expands **command substitution** if not within `'...'`.
 - `"$(command)"` → the output of `"command"`
 - `"` command `"` → the output of `"command"`
- The shell expands **pathname glob** to matching file names if not within `"..."` or `'...'`.
 - `*` → orice caractere
 - `?` → un caracter
 - `[...]` → any one of the characters in `"..."`
- The shell looks up **command** from the following and execute it.
 - **function** definition
 - **builtin** command
 - **executable file** in `"$PATH"`
- The shell goes to the next line and repeats this process again from the top of this sequence.

Single quotes within double quotes have no effect.

Executing `"set -x"` in the shell or invoking the shell with `"-x"` option make the shell to print all of commands executed. This is quite handy for debugging.

12.1.7 Utility programs for shell script

In order to make your shell program as portable as possible across Debian systems, it is a good idea to limit utility programs to ones provided by **essential** packages.

- `"aptitude search ~E"` lists **essential** packages.
- `"dpkg -L package_name |grep '/man/man.*/'"` lists manpages for commands offered by *package_name* package.

pachet	popcon(popularitete)	dimensiune	descriere
dash	V:910, I:998	207	small and fast POSIX-compliant shell for sh
coreutils	V:897, I:999	18457	GNU core utilities
grep	V:765, I:999	1297	GNU grep, egrep și fgrep
sed	V:801, I:999	987	GNU sed
mawk	V:458, I:997	296	small and fast awk
debianutils	V:922, I:998	225	miscellaneous utilities specific to Debian
bsdutils	V:429, I:999	335	basic utilities from 4.4BSD-Lite
bsdextrautils	V:725, I:841	361	extra utilities from 4.4BSD-Lite
moreutils	V:16, I:38	231	additional Unix utilities

Tabela 12.7: List of packages containing small utility programs for shell scripts

Indicație

Although `moreutils` may not exist outside of Debian, it offers interesting small programs. Most notable one is `sponge(8)` which is quite useful when you wish to overwrite original file.

Consultați Secțiune 1.6 pentru exemple.

12.2 Scripting in interpreted languages

pachet	popcon	popularity	documentație
dash	V:910, I:998	207	sh : small and fast POSIX-compliant shell for <code>sh</code>
bash	V:875, I:999	7273	sh : "info bash" provided by <code>bash-doc</code>
mawk	V:458, I:997	296	AWK : small and fast <code>awk</code>
gawk	V:255, I:315	3289	AWK : "info gawk" provided by <code>gawk-doc</code>
perl	V:673, I:990	841	Perl : <code>perl(1)</code> and html pages provided by <code>perl-doc</code> and <code>perl-doc-html</code>
libterm-readline-gnu-perl	V:2, I:28	439	Perl extension for the GNU ReadLine/History Library: <code>perlsh(1)</code>
libreply-perl	I:0	171	REPL for Perl: <code>reply(1)</code>
libdevel-repl-perl	V:0, I:0	237	REPL for Perl: <code>re.pl(1)</code>
python3	V:714, I:969	82	Python : <code>python3(1)</code> and html pages provided by <code>python3-doc</code>
tcl	V:23, I:188	20	Tcl : <code>tcl(3)</code> and detail manual pages provided by <code>tcl-doc</code>
tk	V:18, I:182	20	Tk : <code>tk(3)</code> and detail manual pages provided by <code>tk-doc</code>
ruby	V:67, I:171	32	Ruby : <code>ruby(1)</code> , <code>erb(1)</code> , <code>irb(1)</code> , <code>rdoc(1)</code> , <code>ri(1)</code>

Tabela 12.8: List of interpreter related packages

When you wish to automate a task on Debian, you should script it with an interpreted language first. The guide line for the choice of the interpreted language is:

- Use `dash`, if the task is a simple one which combines CLI programs with a shell program.
- Use `python3`, if the task isn't a simple one and you are writing it from scratch.
- Use `perl`, `tcl`, `ruby`, ... if there is an existing code using one of these languages on Debian which needs to be touched up to do the task.

If the resulting code is too slow, you can rewrite only the critical portion for the execution speed in a compiled language and call it from the interpreted language.

12.2.1 Debugging interpreted language codes

Most interpreters offer basic syntax check and code tracing functionalities.

- "**dash -n** *script.sh*" - Syntax check of a Shell script
- "**dash -x** *script.sh*" - Trace a Shell script
- "**python -m py_compile** *script.py*" - Syntax check of a Python script
- "**python -mtrace --trace** *script.py*" - Trace a Python script

- “**perl -l ../libpath -c script.pl**” - Syntax check of a Perl script
- “**perl -d:Trace script.pl**” - Trace a Perl script

For testing code for dash, try Secțiune 9.1.4 which accommodates bash-like interactive environment.

For testing code for perl, try REPL environment for Perl which accommodates Python-like **REPL (=READ + EVAL + PRINT + LOOP)** environment for Perl.

12.2.2 GUI program with the shell script

The shell script can be improved to create an attractive GUI program. The trick is to use one of so-called dialog programs instead of dull interaction using echo and read commands.

pachet	popcon(popularitatem)	descriere
x11-utils	V:217, I:565 651	xmessage(1): display a message or query in a window (X)
whiptail	V:284, I:996 62	displays user-friendly dialog boxes from shell scripts (newt)
dialog	V:9, I:85 493	displays user-friendly dialog boxes from shell scripts (ncurses)
zenity	V:77, I:355 190	display graphical dialog boxes from shell scripts (GTK)
ssft	V:0, I:0 75	Shell Scripts Frontend Tool (wrapper for zenity, kdialog, and dialog with gettext)
gettext	V:51, I:234 7165	"/usr/bin/gettext.sh": translate message

Tabela 12.9: List of dialog programs

Here is an example of GUI program to demonstrate how easy it is just with a shell script.

This script uses zenity to select a file (default /etc/motd) and display it.

GUI launcher for this script can be created following Secțiune 9.4.10.

```
#!/bin/sh -e
# Copyright (C) 2021 Osamu Aoki <osamu@debian.org>, Public Domain
# vim:set sw=2 sts=2 et:
DATA_FILE=$(zenity --file-selection --filename="/etc/motd" --title="Select a file to check ↵
") || \
( echo "E: File selection error" >&2 ; exit 1 )
# Check size of archive
if ( file -ib "$DATA_FILE" | grep -qe '^text/' ) ; then
    zenity --info --title="Check file: $DATA_FILE" --width 640 --height 400 \
        --text="$(head -n 20 "$DATA_FILE")"
else
    zenity --info --title="Check file: $DATA_FILE" --width 640 --height 400 \
        --text="The data is MIME=$(file -ib "$DATA_FILE")"
fi
```

This kind of approach to GUI program with the shell script is useful only for simple choice cases. If you are to write any program with complexities, please consider writing it on more capable platform.

12.2.3 Custom actions for GUI filer

GUI filer programs can be extended to perform some popular actions on selected files using additional extension packages. They can also made to perform very specific custom actions by adding your specific scripts.

- For GNOME, see [NautilusScriptsHowto](#).
- For KDE, see [Creating Dolphin Service Menus](#).
- For Xfce, see [Thunar - Custom Actions](#) and <https://help.ubuntu.com/community/ThunarCustomActions>.
- For LXDE, see [Custom Actions](#).

12.2.4 Perl short script madness

In order to process data, `sh` needs to spawn sub-process running `cut`, `grep`, `sed`, etc., and is slow. On the other hand, `perl` has internal capabilities to process data, and is fast. So many system maintenance scripts on Debian use `perl`.

Let's think following one-liner AWK script snippet and its equivalents in Perl.

```
awk '($2=="1957") { print $3 }' |
```

This is equivalent to any one of the following lines.

```
perl -ne '@f=split; if ($f[1] eq "1957") { print "$f[2]\n"}' |
```

```
perl -ne 'if ((@f=split)[1] eq "1957") { print "$f[2]\n"}' |
```

```
perl -ne '@f=split; print $f[2] if ( $f[1]==1957 )' |
```

```
perl -lane 'print $F[2] if $F[1] eq "1957"' |
```

```
perl -lane 'print$F[2]if$F[1]eq+1957' |
```

The last one is a riddle. It took advantage of following Perl features.

- The whitespace is optional.
- The automatic conversion exists from number to the string.
- Perl execution tricks via command line options: `perlrun(1)`
- Perl special variables: `perlvar(1)`

This flexibility is the strength of Perl. At the same time, this allows us to create cryptic and tangled codes. So be careful.

12.3 Coding in compiled languages

Here, Secțiune [12.3.3](#) and Secțiune [12.3.4](#) are included to indicate how compiler-like program can be written in C language by compiling higher level description into C language.

12.3.1 C

You can set up proper environment to compile programs written in the [C programming language](#) by the following.

```
# apt-get install glibc-doc manpages-dev libc6-dev gcc build-essential
```

The `libc6-dev` package, i.e., GNU C Library, provides [C standard library](#) which is collection of header files and library routines used by the C programming language.

See references for C as the following.

- "info libc" (C library function reference)
- `gcc(1)` și «info gcc»
- `each_C_library_function_name(3)`
- Kernighan & Ritchie, "The C Programming Language", 2nd edition (Prentice Hall)

pachet	popcon	(populație)	descriere
gcc	V:134, I:565	36	GNU C compiler
libc6-dev	V:266, I:585	13574	GNU C Library: Development Libraries and Header Files
g++	V:53, I:526	13	GNU C++ compiler
libstdc++-14-dev	V:27, I:203	24507	GNU Standard C++ Library v3 (development files)
cpp	V:323, I:725	18	GNU C preprocessor
gettext	V:51, I:234	7165	GNU Internationalization utilities
glade	V:0, I:3	1613	GTK User Interface Builder
valac	V:0, I:3	532	C# like language for the GObject system
flex	V:6, I:70	1239	LEX-compatible fast lexical analyzer generator
bison	V:6, I:75	3118	YACC-compatible parser generator
susv2	I:0	16	fetch "The Single UNIX Specifications v2"
susv3	I:0	16	fetch "The Single UNIX Specifications v3"
susv4	I:0	16	fetch "The Single UNIX Specifications v4"
golang	I:21	12	Go programming language compiler
rustc	V:4, I:18	13748	Rust systems programming language
gfortran	V:4, I:53	15	GNU Fortran 95 compiler
fpc	I:2	104	Free Pascal

Tabela 12.10: List of compiler related packages

12.3.2 Simple C program (gcc)

A simple example "example.c" can be compiled with a library "libm" into an executable "run_example" by the following.

```
$ cat > example.c << EOF
#include <stdio.h>
#include <math.h>
#include <string.h>

int main(int argc, char **argv, char **envp){
    double x;
    char y[11];
    x=sqrt(argc+7.5);
    strncpy(y, argv[0], 10); /* prevent buffer overflow */
    y[10] = '\0'; /* fill to make sure string ends with '\0' */
    printf("%5i, %5.3f, %10s, %10s\n", argc, x, y, argv[1]);
    return 0;
}
EOF
$ gcc -Wall -g -o run_example example.c -lm
$ ./run_example
    1, 2.915, ./run_exam,      (null)
$ ./run_example 1234567890qwerty
    2, 3.082, ./run_exam, 1234567890qwerty
```

Here, "-lm" is needed to link library "/usr/lib/libm.so" from the libc6 package for sqrt(3). The actual library is in "/lib/" with filename "libm.so.6", which is a symlink to "libm-2.7.so".

Look at the last parameter in the output text. There are more than 10 characters even though "%10s" is specified.

The use of pointer memory operation functions without boundary checks, such as sprintf(3) and strcpy(3), is deprecated to prevent buffer overflow exploits that leverage the above overrun effects. Instead, use snprintf(3) and strncpy(3).

12.3.3 Flex — a better Lex

Flex is a [Lex](#)-compatible fast [lexical analyzer](#) generator.

Tutorial for `flex(1)` can be found in `"info flex"`.

Many simple examples can be found under `"/usr/share/doc/flex/examples/"`. [1](#)

12.3.4 Bison — a better Yacc

Several packages provide a [Yacc](#)-compatible lookahead [LR parser](#) or [LALR parser](#) generator in Debian.

pachet	popcon	popularity	descriere
bison	V:6, I:75	3118	GNU LALR parser generator
byacc	V:0, I:3	263	Berkeley LALR parser generator
btyacc	V:0, I:0	251	backtracking parser generator based on byacc

Tabela 12.11: List of Yacc-compatible LALR parser generators

Tutorial for `bison(1)` can be found in `"info bison"`.

You need to provide your own `"main()"` and `"yyerror()"`. `"main()"` calls `"yyparse()"` which calls `"yylex()"`, usually created with Flex.

Here is an example to create a simple terminal calculator program.

Let's create `example.y`:

```
/* calculator source for bison */
%{
#include <stdio.h>
extern int yylex(void);
extern int yyerror(char *);
%}

/* declare tokens */
%token NUMBER
%token OP_ADD OP_SUB OP_MUL OP_RGT OP_LFT OP_EQU

%%
calc:
| calc exp OP_EQU    { printf("Y: RESULT = %d\n", $2); }
;

exp: factor
| exp OP_ADD factor  { $$ = $1 + $3; }
| exp OP_SUB factor  { $$ = $1 - $3; }
;

factor: term
| factor OP_MUL term { $$ = $1 * $3; }
;

term: NUMBER
| OP_LFT exp OP_RGT  { $$ = $2; }
;
%%
```

¹Some [tweaks](#) may be required to get them work under the current system.


```
int main(int argc, char **argv)
{
    yyparse();
}

int yyerror(char *s)
{
    fprintf(stderr, "error: '%s'\n", s);
}
```

Let's create, `example.l`:

```
/* calculator source for flex */
%{
#include "example.tab.h"
%}

%%
[0-9]+ { printf("L: NUMBER = %s\n", yytext); yylval = atoi(yytext); return NUMBER; }
"+"    { printf("L: OP_ADD\n"); return OP_ADD; }
"-"    { printf("L: OP_SUB\n"); return OP_SUB; }
"*"    { printf("L: OP_MUL\n"); return OP_MUL; }
"("    { printf("L: OP_LFT\n"); return OP_LFT; }
")"    { printf("L: OP_RGT\n"); return OP_RGT; }
"="    { printf("L: OP_EQU\n"); return OP_EQU; }
"exit" { printf("L: exit\n"); return YYEOF; } /* YYEOF = 0 */
.      { /* ignore all other */ }
%%
```

Then execute as follows from the shell prompt to try this:

```
$ bison -d example.y
$ flex example.l
$ gcc -lfl example.tab.c lex.yy.c -o example
$ ./example
1 + 2 * ( 3 + 1 ) =
L: NUMBER = 1
L: OP_ADD
L: NUMBER = 2
L: OP_MUL
L: OP_LFT
L: NUMBER = 3
L: OP_ADD
L: NUMBER = 1
L: OP_RGT
L: OP_EQU
Y: RESULT = 9

exit
L: exit
```

12.4 Static code analysis tools

[Lint](#) like tools can help automatic [static code analysis](#).

[Indent](#) like tools can help human code reviews by reformatting source codes consistently.

[Ctags](#) like tools can help human code reviews by generating an index (or tag) file of names found in source codes.

Indicație

Configuring your favorite editor (emacs or vim) to use asynchronous lint engine plugins helps your code writing. These plugins are getting very powerful by taking advantage of [Language Server Protocol](#). Since they are moving fast, using their upstream code instead of Debian package may be a good option.

pachet	popcon	(popularity)	descriere
vim-ale	I:0	2833	Asynchronous Lint Engine for Vim 8 and NeoVim
vim-syntastic	I:2	1379	Syntax checking hacks for vim
elpa-flycheck	V:0, I:1	826	modern on-the-fly syntax checking for Emacs
elpa-relint	V:0, I:0	150	Emacs Lisp regexp mistake finder
cppcheck-gui	V:0, I:1	7682	tool for static C/C++ code analysis (GUI)
shellcheck	V:2, I:15	35220	lint tool for shell scripts
pyflakes3	V:2, I:15	20	passive checker of Python 3 programs
pylint	V:4, I:20	2089	Python code static checker
perl	V:673, I:990	841	interpreter with internal static code checker: B::Lint(3perl)
rubocop	V:0, I:1	3247	Ruby static code analyzer
clang-tidy	V:1, I:12	22	clang-based C++ linter tool
splint	V:0, I:1	2328	tool for statically checking C programs for bugs
flawfinder	V:0, I:0	205	tool to examine C/C++ source code and looks for security weaknesses
black	V:4, I:16	10138	uncompromising Python code formatter
perltidy	V:0, I:3	3086	Perl script indenter and reformatter
indent	V:0, I:5	438	C language source code formatting program
astyle	V:0, I:2	769	Source code indenter for C, C++, Objective-C, C#, and Java
bcpp	V:0, I:0	114	C(++) beautifier
xmlindent	V:0, I:0	52	XML stream reformatter
global	V:0, I:1	1923	Source code search and browse tools
exuberant-ctags	V:1, I:14	341	build tag file indexes of source code definitions
universal-ctags	V:1, I:12	4238	build tag file indexes of source code definitions

Tabela 12.12: List of tools for static code analysis

12.5 Depanare

Debug is important part of programming activities. Knowing how to debug programs makes you a good Debian user who can produce meaningful bug reports.

pachet	popcon	(popularity)	documentație
gdb	V:74, I:154	12478	"info gdb" provided by gdb-doc
ddd	V:0, I:5	4210	"info ddd" provided by ddd-doc

Tabela 12.13: List of debug packages

12.5.1 Basic gdb execution

Primary [debugger](#) on Debian is gdb(1) which enables you to inspect a program while it executes.

Let's install gdb and related programs by the following.

```
# apt-get install gdb gdb-doc build-essential devscripts
```

Good tutorial of gdb can be found:

- “info gdb”
- “Debugging with GDB” in `/usr/share/doc/gdb-doc/html/gdb/index.html`
- [“tutorial on the web”](#)

Here is a simple example of using `gdb(1)` on a “program” compiled with the “-g” option to produce debugging information.

```
$ gdb program
(gdb) b 1           # set break point at line 1
(gdb) run args      # run program with args
(gdb) next          # next line
...
(gdb) step          # step forward
...
(gdb) p parm        # print parm
...
(gdb) p parm=12     # set value to 12
...
(gdb) quit
```

Indicație

Many `gdb(1)` commands can be abbreviated. Tab expansion works as in the shell.

12.5.2 Debugging the Debian package

Since all installed binaries should be stripped on the Debian system by default, most debugging symbols are removed in the normal package. In order to debug Debian packages with `gdb(1)`, *-dbgsym packages need to be installed (e.g. `coreutils-dbgsym` in the case of `coreutils`). The source packages generate *-dbgsym packages automatically along with normal binary packages and those debug packages are placed separately in [debian-debug](#) archive. Please refer to [articles on Debian Wiki](#) for more information.

If a package to be debugged does not provide its *-dbgsym package, you need to install it after rebuilding it by the following.

```
$ mkdir /path/new ; cd /path/new
$ sudo apt-get update
$ sudo apt-get dist-upgrade
$ sudo apt-get install fakeroot devscripts build-essential
$ apt-get source package_name
$ cd package_name*
$ sudo apt-get build-dep ./
```

Fix bugs if needed.

Bump package version to one which does not collide with official Debian versions, e.g. one appended with “+debug1” when recompiling existing package version, or one appended with “~pre1” when compiling unreleased package version by the following.

```
$ dch -i
```

Compile and install packages with debug symbols by the following.

```
$ export DEB_BUILD_OPTIONS="nostrip noopt"
$ debuild
$ cd ..
$ sudo debi package_name*.changes
```

You need to check build scripts of the package and ensure to use "CFLAGS=-g -Wall" for compiling binaries.

12.5.3 Obtaining backtrace

When you encounter program crash, reporting bug report with cut-and-pasted backtrace information is a good idea. The backtrace can be obtained by `gdb(1)` using one of the following approaches:

- Crash-in-GDB approach:
 - Rulați programul din GDB.
 - Crash the program.
 - Tastați „bt” în promptul GDB.
- Crash-first approach:
 - Update the **“/etc/security/limits.conf”** file to include the following:

```
* soft core unlimited
```

- Type `“ulimit -c unlimited”` to the shell prompt.
- Run the program from this shell prompt.
- Crash the program to produce a [core dump](#) file.
- Load the [core dump](#) file to GDB as `“gdb gdb ./program_binary core”`.
- Tastați „bt” în promptul GDB.

For infinite loop or frozen keyboard situation, you can force to crash the program by pressing `Ctrl-\` or `Ctrl-C` or executing `“kill -ABRT PID”`. (See Secțiune [9.4.12](#))

Indicație

Often, you see a backtrace where one or more of the top lines are in `“malloc()”` or `“g_malloc()”`. When this happens, chances are your backtrace isn't very useful. The easiest way to find some useful information is to set the environment variable `“$MALLOCCHECK_”` to a value of 2 (`malloc(3)`). You can do this while running `gdb` by doing the following.

```
$ MALLOCCHECK_=2 gdb hello
```

12.5.4 Advanced gdb commands

12.5.5 Check dependency on libraries

Use `ldd(1)` to find out a program's dependency on libraries by the followings.

```
$ ldd /usr/bin/ls
    librt.so.1 => /lib/librt.so.1 (0x4001e000)
    libc.so.6 => /lib/libc.so.6 (0x40030000)
    libpthread.so.0 => /lib/libpthread.so.0 (0x40153000)
    /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

For `ls(1)` to work in a `“chroot”`ed environment, the above libraries must be available in your `“chroot”`ed environment. See Secțiune [9.4.6](#).

comanda	description for command objectives
(gdb) thread apply all bt	get a backtrace for all threads for multi-threaded program
(gdb) bt full	get parameters came on the stack of function calls
(gdb) thread apply all bt full	get a backtrace and parameters as the combination of the preceding options
(gdb) thread apply all bt full 10	get a backtrace and parameters for top 10 calls to cut off irrelevant output
(gdb) set logging on	write log of gdb output to a file (the default is "gdb.txt")

Tabela 12.14: List of advanced gdb commands

12.5.6 Dynamic call tracing tools

There are several dynamic call tracing tools available in Debian. See Secțiune 9.4.

12.5.7 Debugging X Errors

If a GNOME program preview1 has received an X error, you should see a message as follows.

The program 'preview1' received an X Window System error.

If this is the case, you can try running the program with "- -sync", and break on the "gtk_x_error" function in order to obtain a backtrace.

12.5.8 Memory leak detection tools

There are several memory leak detection tools available in Debian.

pachet	popcon	popularity	descriere
libc6-dev	V:266, I:585	13574	mt race(1): malloc debugging functionality in glibc
valgrind	V:5, I:35	87847	memory debugger and profiler
electric-fence	V:0, I:2	69	ma lloc(3) debugger
libdmalloc5	V:0, I:0	380	debug memory allocation library
duma	V:0, I:0	297	library to detect buffer overruns and under-runs in C and C++ programs
leaktracer	V:0, I:0	56	memory-leak tracer for C++ programs

Tabela 12.15: List of memory leak detection tools

12.5.9 Disassemble binary

You can disassemble binary code with objdump(1) by the following.

```
$ objdump -m i386 -b binary -D /usr/lib/grub/x86_64-pc/stage1
```

Notă

gdb(1) may be used to disassemble code interactively.

12.6 Instrumentele de construcție

pachet	popcon	popularity	documentație
make	V:126, I:566	1762	"info make" provided by make-doc
autoconf	V:28, I:210	2197	"info autoconf" provided by autoconf-doc
automake	V:28, I:210	1933	"info automake" provided by automake1.10-doc
libtool	V:24, I:193	1245	"info libtool" provided by libtool-doc
cmake	V:18, I:119	41613	cmake(1) cross-platform, open-source make system
ninja-build	V:7, I:51	456	ninja(1) small build system closest in spirit to Make
meson	V:6, I:28	4183	meson(1) high productivity build system on top of ninja
xutils-dev	V:0, I:7	1495	imake(1), xmkmf(1), etc.

Tabela 12.16: List of build tool packages

12.6.1 Make

Make is a utility to maintain groups of programs. Upon execution of `make(1)`, make read the rule file, "Makefile", and updates a target if it depends on prerequisite files that have been modified since the target was last modified, or if the target does not exist. The execution of these updates may occur concurrently.

The rule file syntax is the following.

```
target: [ prerequisites ... ]
[TAB] command1
[TAB] -command2 # ignore errors
[TAB] @command3 # suppress echoing
```

Here "[TAB]" is a TAB code. Each line is interpreted by the shell after make variable substitution. Use "\" at the end of a line to continue the script. Use "\$\$" to enter "\$" for environment values for a shell script.

Implicit rules for the target and prerequisites can be written, for example, by the following.

```
%.o: %.c header.h
```

Here, the target contains the character "%" (exactly one of them). The "%" can match any nonempty substring in the actual target filenames. The prerequisites likewise use "%" to show how their names relate to the actual target name.

automatic variable	valoare
\$@	target
\$<	first prerequisite
\$?	all newer prerequisites
\$^	all prerequisites
\$*	"%" matched stem in the target pattern

Tabela 12.17: List of make automatic variables

expandarea variabilei	descriere
foo1 := bar	one-time expansion
foo2 = bar	recursive expansion
foo3 += bar	adăugare

Tabela 12.18: List of make variable expansions

Run "`make -p -f/dev/null`" to see automatic internal rules.

12.6.2 Autotools

Autotools is a suite of programming tools designed to assist in making source code packages portable to many **Unix-like** systems.

- **Autoconf** is a tool to produce a shell script "configure" from "configure.ac".
 - "configure" is used later to produce "Makefile" from "Makefile.in" template.
- **Automake** is a tool to produce "Makefile.in" from "Makefile.am".
- **Libtool** is a shell script to address the software portability problem when compiling shared libraries from source code.

12.6.2.1 Compile and install a program



Avertisment

Do not overwrite system files with your compiled programs when installing them.

Debian does not touch files in "/usr/local/" or "/opt". So if you compile a program from source, install it into "/usr/local/" so it does not interfere with Debian.

```
$ cd src
$ ./configure --prefix=/usr/local
$ make # this compiles program
$ sudo make install # this installs the files in the system
```

12.6.2.2 Uninstall program

If you have the original source and if it uses autoconf(1)/automake(1) and if you can remember how you configured it, execute as follows to uninstall the program.

```
$ ./configure all-of-the-options-you-gave-it
$ sudo make uninstall
```

Alternatively, if you are absolutely sure that the install process puts files only under "/usr/local/" and there is nothing important there, you can erase all its contents by the following.

```
# find /usr/local -type f -print0 | xargs -0 rm -f
```

If you are not sure where files are installed, you should consider using checkinstall(8) from the checkinstall package, which provides a clean path for the uninstall. It now supports to create a Debian package with "-D" option.

12.6.3 Meson

The software build system has been evolving:

- **Autotools** on the top of **Make** has been the de facto standard for the portable build infrastructure since 1990s. This is extremely slow.
 - **CMake** initially released in 2000 improved speed significantly but was originally built on the top of inherently slow **Make**. (Now **Ninja** can be its backend.)
-

- [Ninja](#) initially released in 2012 is meant to replace Make for the further improved build speed and is designed to have its input files generated by a higher-level build system.
- [Meson](#) initially released in 2013 is the new popular and fast higher-level build system which uses [Ninja](#) as its backend.

See documents found at "[The Meson Build system](#)" and "[The Ninja build system](#)".

12.7 Web

Basic interactive dynamic web pages can be made as follows.

- Queries are presented to the browser user using [HTML](#) forms.
- Filling and clicking on the form entries sends one of the following [URL](#) string with encoded parameters from the browser to the web server.
 - "https://www.foo.dom/cgi-bin/program.pl?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3"
 - "https://www.foo.dom/cgi-bin/program.py?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3"
 - "https://www.foo.dom/program.php?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3"
- "%nn" in URL is replaced with a character with hexadecimal nn value.
- The environment variable is set as: "QUERY_STRING="VAR1=VAL1 VAR2=VAL2 VAR3=VAL3"".
- [CGI](#) program (any one of "program.*") on the web server executes itself with the environment variable "\$QUERY_STRING".
- stdout of CGI program is sent to the web browser and is presented as an interactive dynamic web page.

For security reasons it is better not to hand craft new hacks for parsing CGI parameters. There are established modules for them in Perl and Python. [PHP](#) comes with these functionalities. When client data storage is needed, [HTTP cookies](#) are used. When client side data processing is needed, [Javascript](#) is frequently used.

For more, see the [Common Gateway Interface](#), [The Apache Software Foundation](#), and [JavaScript](#).

Searching "CGI tutorial" on Google by typing encoded URL <https://www.google.com/search?hl=en&ie=UTF-8&q=CGI+tutorial> directly to the browser address is a good way to see the CGI script in action on the Google server.

12.8 The source code translation

There are programs to convert source codes.

pachet	popcon(popularity contest)	limita de popularitate	keyword	descriere
perl	V:673, I:990	841	AWK → PERL	convert source codes from AWK to PERL: a2p(1)
f2c	V:0, I:2	443	FORTTRAN → C	convert source codes from FORTRAN 77 to C/C++: f2c(1)
intel2gas	V:0, I:0	178	intel → gas	converter from NASM (Intel format) to the GNU Assembler (GAS)

Tabela 12.19: List of source code translation tools

12.9 Making Debian package

If you want to make a Debian package, read followings.

- Cap. 2 to understand the basic package system
- Secțiune 2.7.13 to understand basic porting process
- Secțiune 9.11.4 to understand basic chroot techniques
- `debuild(1)`, și `sbuid(1)`
- Secțiune 12.5.2 for recompiling for debugging
- [Guide for Debian Maintainers](#) (the `debmake-doc` package)
- [Debian Developer's Reference](#) (the `developers-reference` package)
- [Debian Policy Manual](#) (the `debian-policy` package)

There are packages such as `debmake`, `dh-make`, `dh-make-perl`, etc., which help packaging.

Anexa A

Appendix

Here are backgrounds of this document.

A.1 The Debian maze

The Linux system is a very powerful computing platform for a networked computer. However, learning how to use all its capabilities is not easy. Setting up the LPR printer queue with a non-PostScript printer was a good example of stumble points. (There are no issues anymore since newer installations use the new CUPS system.)

There is a complete, detailed map called the "SOURCE CODE". This is very accurate but very hard to understand. There are also references called HOWTO and mini-HOWTO. They are easier to understand but tend to give too much detail and lose the big picture. I sometimes have a problem finding the right section in a long HOWTO when I need a few commands to invoke.

I hope this "Debian Reference (version 2.136)" () provides a good starting direction for people in the Debian maze.

A.2 Copyright history

The Debian Reference was initiated by me, Osamu Aoki <osamu at debian dot org>, as a personal system administration memo. Many contents came from the knowledge I gained from [the debian-user mailing list](#) and other Debian resources.

Following a suggestion from Josip Rodin, who was very active with the [Debian Documentation Project \(DDP\)](#), "Debian Reference (version 1, 2001-2007)" was created as a part of DDP documents.

After 6 years, I realized that the original "Debian Reference (version 1)" was outdated and started to rewrite many contents. New "Debian Reference (version 2)" is released in 2008.

I have updated "Debian Reference (version 2)" to address new topics (Systemd, Wayland, IMAP, PipeWire, Linux kernel 5.10) and removed outdated topics (SysV init, CVS, Subversion, SSH protocol 1, Linux kernels before 2.5). References to Jessie 8 (2015-2020) release situation or older are mostly removed.

This "Debian Reference (version 2.136)" () covers mostly Trixie (=stable) and Forky (=testing) Debian releases.

The tutorial contents can trace its origin and its inspiration in followings.

- "[Linux User's Guide](#)" by Larry Greenfield (December 1996)
 - obsoleted by "Debian Tutorial"
- "Debian Tutorial" by Havoc Pennington. (11 December, 1998)

- partially written by Oliver Elphick, Ole Tetlie, James Treacy, Craig Sawyer, and Ivan E. Moore II
- obsoleted by "Debian GNU/Linux: Guide to Installation and Usage"
- "[Debian GNU/Linux: Guide to Installation and Usage](#)" by John Goerzen and Ossama Othman (1999)
 - obsoleted by "Debian Reference (version 1)"

The package and archive description can trace some of their origin and their inspiration in following.

- "[Debian FAQ](#)" (March 2002 version, when this was maintained by Josip Rodin)

The other contents can trace some of their origin and their inspiration in following.

- "Debian Reference (version 1)" by Osamu Aoki (2001–2007)
 - obsoleted by the newer "Debian Reference (version 2)" in 2008.

The previous "Debian Reference (version 1)" was created with many contributors.

- the major contents contribution on network configuration topics by Thomas Hood
- significant contents contribution on X and VCS related topics by Brian Nelson
- the help on the build scripts and many content corrections by Jens Seidel
- extensive proofreading by David Sewell
- many contributions by the translators, contributors, and bug reporters

Many manual pages and info pages on the Debian system as well as upstream web pages and [Wikipedia](#) documents were used as the primary references to write this document. To the extent Osamu Aoki considered within the [fair use](#), many parts of them, especially command definitions, were used as phrase pieces after careful editorial efforts to fit them into the style and the objective of this document.

The gdb debugger description was expanded using [Debian wiki contents on backtrace](#) with consent by Ari Pollak, Loïc Minier, and Dafydd Harries.

Contents of the current "Debian Reference (version 2.136)" () are mostly my own work except as mentioned above. These has been updated by the contributors too.

The author, Osamu Aoki, thanks all those who helped make this document possible.

A.3 Formatul documentului

The source of the English original document is currently written in [DocBook](#) XML files. This Docbook XML source are converted to HTML, plain text, PostScript, and PDF. (Some formats may be skipped for distribution.)