

Introducere în Linux

Un ghid la îndemână

Machtelt Garrels

Garrels.be

<tille.wants.no.spam_at_garrels.be>

Ediția 1.25

Copyright © 2002, 2003, 2004, 2005, 2006, 2007 Machtelt Garrels
20070919

Traducere în limba română: *Marian „miaumiau” Vasile*, care mulțumește [comunității UBUNTU România](#) pentru ajutorul pe care l-a primit din momentul în care a descoperit acest sistem de operare și [Grupului pentru software liber](#) pentru promovarea programelor sursă deschisă.

Introducere

1. De ce acest ghid?

Mulți oameni tot mai cred despre Linux că este dificil de învățat sau că numai experții pot înțelege cum lucrează un sistem Linux. Deși există multă documentație disponibilă gratuit, această documentație este împrăștiată pe web și, câteodată, este dificilă de înțeles, deoarece este orientată de regulă către utilizatori experimentați de UNIX sau Linux. Astăzi, mulțumită eforturilor depuse în dezvoltare, Linux-ul a câștigat popularitate atât în mediul familial, cât și la locurile noastre de muncă. Zielul acestui ghid este acela de a arăta oamenilor de toate vîrstele că Linux-ul poate fi simplu, amuzant și folositor multor scopuri.

2. Cine ar trebui să citească acest ghid?

Acest ghid a fost scris ca o trecere în revistă a sistemului de operare Linux, orientată către utilizatori noi, cărora să le servească drept îndrumător într-o explorare a Linux-ului, cu exerciții la sfârșitul fiecărui capitol. Cei care sunt mai avansați pot apela la el pentru revizuirea și împrospătarea cunoștințelor, deoarece conține noțiunile de bază, care sunt necesare în administrarea sistemelor și rețelelor. El conține multe exemple izvorăte din experiența de administrator de sistem și rețele Linux, de îndrumător și consultant a celei care a scris acest ghid. Noi sperăm că aceste exemple vă vor ajuta să înțelegeți mai bine sistemele Linux și că veți încerca aceste lucruri pe calculatoarele voastre.

Oricine dorește să capete experiență în utilizarea liniei de comandă din Linux (și, în general, UNIX) va găsi acest ghid ca fiind foarte folositor.

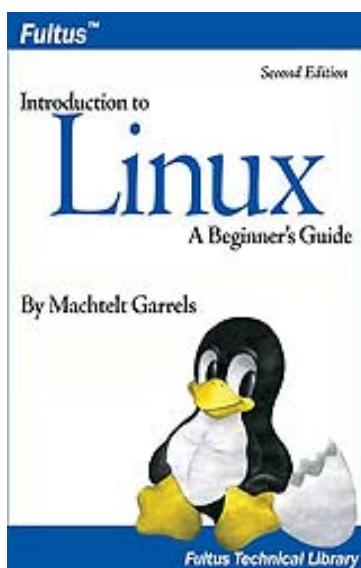
3. Versiuni noi și disponibilitate

Acest document este publicat în secțiunea Ghiduri (Guides) a Proiectului pentru documentație Linux (Linux Documentation Project) la adresa <http://www.tldp.org/guides.html>; îl puteți descărca, de asemenea, în format PDF și PostScript de la aceeași adresă.

Cea mai recentă ediție este disponibilă la <http://tille.garrels.be/training/tldp/>.

Cea de-a doua ediție a ghidului este disponibilă ca material tipărit la Fultus.com Books ca Print On Demand (tipăritură la cerere). Fultus distribuie acest ghid prin Ingram and Baker & Taylor către multe librării, printre care Amazon.com, Amazon.co.uk, BarnesAndNoble.com and Google's Froogle, portalul global pentru cumpărături, și Google Book Search.

Figura 1. Coperta ghidului



Acest ghid a fost tradus în hindu (<http://www.geocities.com/linuxparichay/>) de :

- Alok Kumar
- Dhananjay Sharma
- Kapil
- Puneet Goel
- Ravikant Yuyutsu

Andrea Montagner a tradus acest ghid în limba italiană. El se găsește la adresa de web <http://www.codex.altervista.org/introlinux.html>.

4. Revizuiri

Versiunea 1.26 19-09-2007

Revizuită de: MG

Comentarii de la cititori, licențiere.

Versiunea 1.25 11-05-2007

Revizuită de: MG

Comentarii de la cititori, actualizări minore, eticheta în utilizarea e-mail-ului , informații la zi despre disponibilitate (mulțumiri lui Oleg).

Versiunea 1.24 01-11-2006

Revizuită de: MG

Adăugați termeni ai index-ului, pregătire pentru cea de-a doua ediție tipărită, adăugate informații despre gpg (general public licence) și proxy-uri.

Versiunea 1.23 25-07-2006

Revizuită de: MG și FK

Actualizări și corecturi, scos appendix-ul 5 din nou, modificarea licenței pentru a permite includerea în documentația Debian.

Versiunea 1.22 06-04-2006

Revizuită de: MG

Capitolul 8 revizuit complet, capitolul 10: clarificate exemplele, adăugate informații despre ifconfig și cygwin, revizuită secțiunea despre network apps.

Versiunea 1.21 14-03-2006

Revizuită de: MG

Adăugate exerciții în capitolul 11, corectate erorile în secțiunea newline, completate comenzile din capitolul 9, corecții minore în capitolul 10.

Versiunea 1.20 06-01-2006

Revizuită de: MG

Împărțit capitolul 7: secțiunea despre audio se găsește acum în capitol separat, capitolul 11. Revizuri minore, actualizări pentru comenzi ca aptitude, mai multe despre mediile de stocare USB, telefonie Internet, corecturi de la cititori.

Versiunea 1.13 27-04-2004

Revizuită de: MG

Corecturi de ultim moment înainte de a trimite materialul către Fultus pentru tipărire. Adăugarea referințelor Fultus în secțiunea Revizuiră, revizuirea secțiunilor Convenții și Organizare. Schimbări minore în capitolele 4, 5, 6 și 8, adăugarea informațiilor despre rdesktop în capitolul 10, revizuirea glosarului, înlocuirea referințelor despre fileutils cu cele pentru coreutils, mulțumiri către translatorii pentru limba hindu.

5. Contribuții

Mulțumesc din inimă tuturor oamenilor care și-au împărtășit experiența lor. Și, în special, utilizatorilor de Linux belgieni, care au fost disponibili zi de zi și care au trimis comentariile lor foarte folositoare.

De asemenea, un gând bun către Tabatha Marshall, pentru că a făcut o revizuire completă, corectură grammaticală și de stil, și către Eugen Crosser, care a observat erorile trecute cu vederea de către noi două.

Mulțumesc și tuturor cititorilor care m-au atenționat despre topicuri ce nu se regăseau în ghid și care m-au ajutat să rectific greșelile și definitiile confuze, pentru că au găsit timp să-mi trimită remarcile lor. Aceștia sunt oamenii pentru care mă străduiesc să mențin acest ghid la zi, precum Filipus Klutiero, care a

făcut o revizuire completă în 2005 și 2006 și care m-a ajutat să includ ghidul în colecția de documentație Debian, și Alexey Eremenko, care mi-a trimis bazele de la care a plecat capitolul 11.

În 2006, Suresh Rajashekara a realizat un pachet Debian pentru această documentație.

În sfârșit, multe mulțumiri către voluntari care traduc acest ghid în limbile franceză, suedeza, farsi, hindu și altele. Este o muncă grea care nu ar trebui subestimată; vă admir curajul.

6. Feedback (Reacții și comentarii de la cititor)

Informații lipsă, legături inexistente, greșeli de tipar? Trimiteti un mail către autoare la: <tille.wants.no.spam_at_garrels.be>.

Nu uitați să verificați mai întâi ultima versiune disponibilă, la adresa de Internet <http://tille.garrels.be/training/tldp/>.

7. Informații despre copyright

```
* Copyright (c) 2002-2007, Machtelt Garrels
* All rights reserved.
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are met:
*
*      * Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
*      * Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
*      * Neither the name of the author, Machtelt Garrels, nor the
* names of its contributors may be used to endorse or promote products
* derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS "AS IS" AND ANY
* EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
* WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE AUTHOR AND CONTRIBUTORS BE LIABLE FOR ANY
* DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

Logo-urile, mărcile și simbolurile utilizate în acest ghid sunt proprietatea respectivilor deținători.

8. Ce vă trebuie?

Aveți nevoie de un calculator și un mediu care să conțină o distribuție de Linux. Cea mai mare parte a acestui ghid este valabilă tuturor distribuțiilor de Linux – și UNIX, în general. La acest moment, nu există alte cerințe.

Capitolul CUM SĂ instalăm conține informații folosite care descrevă obținerea sistemului de operare Linux și instalarea acestuia în calculator. Cerințele de hardware și coexistența cu alte sisteme de operare vor fi, de asemenea, discutate.

Imaginile CD-urilor pot fi descărcate de la <http://www.linux-iso.com/> și din alte locuri, prezentate în Appendix A.

O alternativă interesantă pentru cei care nu vor să facă pasul către o instalare a Linux-ului pe calculatorul lor este cea a distribuțiilor de Linux care pot rula și de pe CD, precum Knoppix (<http://www.knoppix.com/>) sau Ubuntu^{n.t.} (<http://www.ubuntu.com/>).

9. Convenții utilizate în acest document

Următoarele convenții folosite sunt utilizate în acest text:

Tabelul 1. Convenții utilizate în acest ghid

„Text citat”	Considerații de la cititor, evidențierea unui răspuns al calculatorului.
captură din Terminal	Captură de text introdus sau afișat în Terminal, evidențiat, de regulă, de un fond gri deschis.
comandă	Numele unei comenzi care poate fi introdusă în linia de comandă.
VARIABILĂ	Numele unei variabile sau al unui indicator ce conține o variabilă, precum în \$VARNAME
opțiune	Opțiune a unei comenzi, precum în: „opțiunea -a a comenzi ls”.
argument	Argumentul unei comenzi, precum în „citește man ls ”.
prompt	Promptul utilizatorului, urmat, de regulă, de o comandă pe care ați introdus-o în Terminal, precum în hilda@home> ls -l
comandă opțiune argumente	Sintaxa unei comenzi, scrisă ca linie separată.
nume_fișier	Numele unui fișier sau director, de exemplu „Mergeti la directorul /usr/bin”.
Tastă	Taste care trebuie apăsate, precum în „tastați Q pentru renunțare”.
Buton	Buton grafic care trebuie apăsat, precum butonul OK.
Meniu→Alege	Alegere făcută dintr-un meniu grafic, de exemplu: „Selectați Ajutor →Despre Mozilla din browser-ul (navigatorul) dumneavoastră”.
<i>Terminologie</i>	Termen sau concept important: „Kernelul Linux este inima sistemului.”
\	Acest caracter, într-o captură din Terminal sau sintaxă de comandă, indică o linie neterminată. Cu alte cuvinte, dacă vedeti o comandă mai lungă, care este prezentată în mai multe linii, \ înseamnă „Nu tastați încă Enter”.

Sunt utilizate și următoarele imagini:



Aceasta este o notă. Contine informații adiționale sau anumite remarcări.



Aceasta este o atenționare. Înseamnă să fiți atenți.



Aceasta este o avertizare. Înseamnă să fiți foarte atenți.



Sfaturi.

10. Organizarea acestui document

Acest ghid este parte a Proiectului de documentație pentru Linux și se dorește a fi punctul de plecare în studierea celorlalte materiale pe care le puteți găsi în cadrul acestui proiect. Ca atare, vă oferă cunoștințele de bază necesare oricărei persoane care dorește să înceapă să lucreze cu un sistem Linux, încercând, în același timp, să evite a inventa din nou apa caldă. De aceea, așteptați-vă ca acest ghid să nu fie complet, ci plin de legături către surse adiționale de informații despre sistemul dumneavoastră, pe Internet și către documentația specifică sistemului dumneavoastră.

Primul capitol este o introducere în subiectul Linux; în următoarele două se discută comenzi de bază. În capitolele 4 și 5 se discută despre subiecte ceva mai avansate, dar, totuși, de bază. De capitolul 6 aveți nevoie pentru a continua cu restul, întrucât se discută despre editarea fișierelor, o abilitate de care aveți nevoie pentru a face trecerea de la nou în Linux, către utilizator de Linux. Capitolele următoare tratează subiecte mai avansate, de care aveți nevoie în utilizarea de zi cu zi a Linux-ului.

Toate capitolele au exerciții care să testează pregătirea în vederea abordării următorului nivel.

- [Capitolul 1](#): Ce este Linux-ul, cum a apărut el, avantaje și dezavantaje, ce va aduce viitorul pentru Linux, cine ar trebui să-l folosească, instalarea în calculatorul dumneavoastră.
- [Capitolul 2](#): Să începem, autentificarea la sistem, comenzi de bază, unde să căutăm ajutor.
- [Capitolul 3](#): Sistemul de fișiere, fișiere și directoare importante, lucru cu fișierele și directoarele, protejarea datelor.
- [Capitolul 4](#): Înțelegerea proceselor și lucrul cu acestea, procedurile de pornire și închidere, amânarea sarcinilor, sarcini repetitive.
- [Capitolul 5](#): Ce este o intrare standard, o ieșire și o eroare și cum sunt folosite aceste elemente din prompterul de comandă.
- [Capitolul 6](#): De ce ar trebui să învățați să lucrați cu un editor de text, discutarea celor mai comune editoare de text.
- [Capitolul 7](#): Configurarea mediului dumneavoastră, grafic, text sau audio, setări pentru utilizatorii de Linux pentru care engleza nu este limba maternă, sfaturi pentru adăugarea altor programe.
- [Capitolul 8](#): Convertirea fișierelor în format pentru tipărire, scoaterea acestora la imprimantă, sfaturi pentru rezolvarea problemelor legate de tipărire.
- [Capitolul 9](#): Pregătirea datelor pentru copiile de rezervă, discutarea diverselor unele, copii de rezervă la distanță.
- [Capitolul 10](#): Trecerea în revistă a unelelor de rețea ale Linux-ului și a aplicațiilor pentru utilizatori, cu o scurtă discuție axată pe programele tip servicii (service daemon) și rețea securizată.
- [Capitolul 11](#): Sunet și video, inclusiv și Voice over IP și înregistrarea sunetului.
- [Appendix A](#): Ce cărți să citiți și ce saituri să vizitați după terminarea acestei cărți.
- [Appendix B](#): O comparație.
- [Appendix C](#): Dacă vă blocați, acest tabel poate fi de ajutor. De asemenea, un bun argument atunci când șeful insistă ca DUMNEAVOASTRĂ să folosiți shell-ul SĂU favorit.

Capitolul 1. Ce este Linux-ul?

Vom începe cu o trecere în revistă a modului în care Linux-ul a devenit sistemul de operare aşa cum îl cunoaştem astăzi. Vom discuta despre trecut și despre dezvoltarea acestui sistem pe viitor și ne vom apela asupra avantajelor și dezavantajelor lui. Vom vorbi despre distribuții, despre Sursa deschisă în general și vom încerca să explicăm pe scurt ce este GNU.

Acest capitol răspunde la întrebări precum:

- ◆ Ce este Linux-ul?
- ◆ Unde și cum a luat naștere Linux-ul?
- ◆ Nu este Linux-ul acel sistem în care totul este făcut în modul text?
- ◆ Are Linux-ul viitor, sau este doar o fantezie?
- ◆ Care sunt avantajele utilizării Linux-ului?
- ◆ Care sunt dezavantajele?
- ◆ Câte feluri de Linux există și cum aleg ceea ce mi se potrivește mie?
- ◆ Ce sunt mișcările Sursa deschisă și GNU?

1.1. Istoric

1.1.1. UNIX

Pentru a înțelege popularitatea Linux-ului, trebuie să călătorim înapoi în timp, cu aproape 30 de ani...

Imaginați-vă calculatoare atât de mari precum casele sau chiar stadioanele. În timp ce doar mărimea acestora ridică probleme substanțiale, exista ceva care făcea ca totul să fie și mai rău: fiecare calculator avea un sistem de operare diferit. Programele erau întotdeauna dedicate unui scop anume, iar un program pentru un anume sistem nu rula pe un altul. A fi capabil să lucrezi cu un sistem nu însemna că, automat, poți lucra cu un altul. Era dificil, atât pentru utilizatori, cât și pentru administratorii de sistem.

Calculatoarele erau deosebit de scumpe, iar sacrificii tot trebuiau făcute și după achiziționarea acestora, doar pentru a face utilizatorii să înțeleagă modul de lucru al acestor mașini. Costul total pe unitatea de calcul era enorm.

Din punct de vedere tehnologic, lumea nu era atât de avansată, aşa că mărimea a mai pus probleme pentru încă o decadă. În 1969, o echipă de dezvoltatori de la laboratoarele Bell Lab a început să lucreze la o soluție pentru problema programelor, pentru a rezolva aspectele de compatibilitate. Ei au dezvoltat un nou sistem de operare, care era:

1. Simplu și elegant.
2. Scris în limbajul de programare C în locul limbajului de asamblare.
3. Capabil să recicleze codul.

Dezvoltatorii de la Bell Labs au numit acest proiect „UNIX”.

Capabilitatea de reciclare a codului era foarte importantă. Până atunci, toate sistemele de operare comerciale care se găseau erau scrise într-un cod dezvoltat pentru un anume sistem. UNIX-ul, pe de altă parte, avea nevoie doar de o porțiune mică a aceluia cod, care este numită acum kernel. Acest kernel este singura porțiune de cod care trebuie adaptată pentru un anume sistem și formează baza unui sistem UNIX. Sistemul de operare și celelalte funcții au fost construite în jurul acestui kernel și scrise într-un limbaj de programare mai evoluat, C.

Acest limbaj a fost dezvoltat în mod special pentru realizarea sistemului UNIX. Utilizând noua tehnică, a fost mult mai ușor să se dezvolte un sistem de operare care să ruleze pe echipamente diferite.

Comerçanții de programe s-au adaptat repede, deoarece puteau acum să vândă de zeci de ori mai mult, aproape fără efort. Situații noi și ciudate au apărut: imaginați-vă de exemplu calculatoare de la diferiți producători care comunică în aceeași rețea, sau utilizatori care lucrează pe mașini diferite fără a avea nevoie de instruire. UNIX-ul a ajutat mult pe utilizatori în privința compatibilității cu calculatoare diferite.

Pe parcursul următoarelor două decenii, dezvoltarea UNIX-ului a continuat. Mai multe lucruri au

devenit posibile, iar din ce în ce mai mulți producători de echipamente și programe au adăugat suport pentru UNIX în produsele lor.

UNIX-ul se găsea inițial numai în medii dedicate, alcătuite din servere și minicalculatoare (luată nota de faptul că PC-ul este un „micro” calculator). Trebuia să lucrezi într-o universitate, pentru guvern sau pentru o mare corporație financiară pentru a atinge un sistem UNIX.

Dar calculatoare mai mici au apărut treptat și, pe la sfârșitul anilor 80, mulți oameni aveau calculatoare acasă. La acel timp, existau câteva versiuni de UNIX destinate arhitecturii PC, dar nici una din ele nu era cu adevărat liberă sau gratuită și, ceea ce era mai important: toate erau extrem de încete, aşa că cei mai mulți oameni rulau MS DOS sau Windows 3.1 pe calculatoarele de acasă.

1.1.2. Linus și Linux

Pe la începutul anilor 90, calculatoarele din segmentul PC erau, în sfârșit, suficient de puternice pentru ca să ruleze un UNIX în întregime. Linus Torvalds, un Tânăr care studia informatică la Universitatea din Helsinki, s-a gândit că ar fi o idee bună să aibă un fel de versiune academică liberă a UNIX-ului și s-a apucat să scrie cod pentru aceasta.

Ei a început să pună întrebări, să caute răspunsuri și soluții care l-ar putea ajuta să aibă UNIX-ul pe calculatorul său. Mai jos vă prezentăm una din primele sale postări în comp.os.minix, datând din 1991:

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: Gcc-1.40 and a posix-question
Message-ID: <1991Jul3.100050.9886@klaava.Helsinki.FI>
Date: 3 Jul 91 10:00:50 GMT
Hello netlanders,
Due to a project I'm working on (in minix), I'm interested in the posix
standard definition. Could somebody please point me to a (preferably)
machine-readable format of the latest posix rules? Ftp-sites would be
nice.
```

Încă de la început, țelul lui Linus a fost să aibă un sistem care să fie pe deplin compatibil cu UNIX-ul original. De aceea întreba despre standardul POSIX, acesta fiind și acum standardul pentru UNIX.

În acele zile nu se inventase încă ceea ce cunoaștem azi a fi plug-and-play, dar destul de mulți oameni doreau un sistem UNIX al lor, încât acest lucru era considerat doar un mic obstacol. Drivere noi pentru tot felul de dispozitive au devenit disponibile, cu o viteză din ce în ce mai mare. Aproape că de îndată ce un nou dispozitiv era scos pe piață, cineva îl cumpăra și îl supunea testării în Linux, după cum a început să fie cunoscut sistemul, conducând la apariția codului liber destinat unei palete largi de echipamente. Acești programatori nu se opreau la propriul PC; orice dispozitiv găseau era folositor Linux-ului.

Pe atunci, acești oameni erau numiți „încuiataj” sau „ciudați”, dar asta nu conta pentru ei, atât timp cât lista cu dispozitivele suportate devenea din ce în ce mai lungă. Mulțumită acestor oameni, Linux-ul nu numai că rulează ideal pe un calculator nou, dar este și singura alegere în cazul unor dispozitive vechi sau exotice, care ar fi inutilizabile dacă Linux-ul nu ar fi existat.

La doi ani după această postare a lui Linus, existau 12000 de utilizatori de Linux. Proiectul, popular printre hobiști, a crescut constant, rămânând în permanență în granițele standardului POSIX. Toate capabilitățile UNIX-ului au fost adăugate pe parcursul următorilor doi ani, rezultând sistemul de operare matur care este astăzi. Linux-ul este o clonă completă a UNIX-ului, potrivită pentru stații de lucru, dar și pentru zona de mijloc sau sectorul de vârf al serverelor. În zilele noastre, mulți jucători importanți de pe piață de dispozitive și programe au propriile echipe de dezvoltatori Linux. Din magazinele aflate în zona dumneavoastră puteți chiar cumpăra sisteme care au Linux-ul preinstalat, cu suport tehnic asigurat, după cum este adevărat și faptul că există dispozitive și programe care nu sunt încă integrate.

1.1.3. Aplicații curente pentru sisteme Linux

Astăzi, Linux-ul s-a alăturat produselor aflate pe piață pentru desktop-uri. Dezvoltatorii de Linux, concentrată în rețelistică și servicii la început, cât și în aplicațiile pentru birou, erau ultima barieră care trebuia înălăturată. Nu ne place să admitem că Microsoft dictează piață, aşa că mai multe alternative, care fac Linux-ul o alegere demnă de luat în seamă pentru calculatoare personale, au fost lansate pe parcursul ultimilor ani, alternative care asigură o interfață cu utilizatorul ușor de utilizat și aplicații pentru birou compatibile cu MS

Office, precum procesoare de text, foi pentru calcul tabelar, prezentări și altele de acest gen.

În ceea ce privește serverele, Linux-ul este binecunoscut ca o platformă stabilă și de încredere, asigurând baze de date și servicii de comerț pentru companii precum Amazon, faimoasa librărie online, Serviciul poștal al S.U.A., Armata germană, și multe altele. În special, furnizorii de Internet și furnizorii de servicii pe Internet s-au dezvoltat ajutați de Linux ca firewall, server proxy sau web; veți găsi un sistem Linux la îndemâna oricărui administrator de sisteme UNIX, care îl apreciază drept o stație confortabilă de gestionare. Sistemele Linux au fost folosite în realizarea unor filme ca „Titanic”, „Shrek” și altele. În oficile poștale, ele sunt centrii nervoși care sortează corespondența, iar în marile motoare de căutare ele sunt folosite pentru a realiza căutările pe Internet. Acestea sunt doar câteva din miile de lucruri complexe pe care le realizează Linux-ul zi de zi, în toată lumea.

Este demn de menționat că Linux-ul modern nu rulează numai pe stații de lucru și servere din sectoarele de mijloc sau de vârf, ci și pe gadgeturi ca PDA-uri, telefoane mobile, multe echipamente dedicate, chiar și pe ceasuri de mână experimentale. Acest lucru face din Linux singurul sistem de operare din lume care acoperă o atât de largă varietate de echipamente.

1.2. Interfața cu utilizatorul

1.2.1. Este Linux-ul dificil?

Răspunsurile la această întrebare diferă în funcție de cui te adresezi. Utilizatorii de UNIX vor spune nu, deoarece Linux-ul este sistemul de operare ideal pentru utilizatori avansați și programatori, pentru că a fost și este dezvoltat de către astfel de oameni.

Tot ceea ce are nevoie un bun programator găsiți aici: compilatoare, librării, unelte pentru dezvoltare și reparare de erori. Aceste pachete vin cu orice distribuție standard de Linux. Compilatorul C inclus este gratuit – lucru opus multor distribuții de UNIX, care impun o taxă de licență pentru această unealtă. Toată documentația și manualele sunt incluse, iar adesea sunt oferite și exemple, pentru a vă ajuta să începeți imediat. Este percepția ca un UNIX, iar trecerea de la UNIX la Linux este un lucru natural.

În zilele de început ale Linux-ului, a fi expert era un fel de cerință pentru a începe să utilizezi sistemul. Cei care stăpâneau Linux-ul se simțeau superiori celorlalți care „nu fuseseră iluminati”... Să spui unui începător „RTFM” (adică să citească ... manualele) era o practică des întâlnită la acea vreme. Manuale erau în orice sistem, dar documentația era greu de găsit, și chiar dacă cineva o găsea, explicațiile erau date în termeni atât de tehnici, încât un începător era repede descurajat din învățarea sistemului.

Comunitatea din jurul Linux-ului a început să înțeleagă că, dacă se dorea ca Linux-ul să devină un jucător important de piață sistemelor de operare, atunci trebuia să apară schimbări serioase în accesibilitatea acestui sistem.

1.2.2. Linux pentru utilizatorii fără experiență

Au apărut companii precum RedHat, SuSE și Mandriva, care au furnizat distribuții de Linux pregătite pentru consumul de masă. Ele au integrat în acestea interfața grafică cu utilizatorul (GUI), dezvoltată de către comunitate, în ideea de a ușura gestionarea programelor și serviciilor. Ca utilizatori ai Linux-ului din zilele noastre, aveți toate mijloacele prin care să vă cunoașteți sistemul din toate punctele de vedere, dar nu mai este necesar să faceți asta pentru a comunica sistemului cererile dumneavoastră.

Acum vă puteți autentifica în modul grafic și lansa toate aplicațiile necesare fără să fiți nevoiți să introduceți comenzi de la tastatură, având, în același timp, posibilitatea de a accesa inima sistemului, dacă este necesar. Datorită structurii sale, Linux-ul permite utilizatorilor să descopere gradual posibilitățile sistemului: se adresează în aceeași măsură utilizatorilor începători și celor experimentați. Noii veniți nu sunt forțați să facă lucruri dificile, iar utilizatorii experimentați nu sunt forțați să lucreze în același mod în care au început să învețe despre Linux.

În timp ce dezvoltările în acest domeniu continuă, lucruri importante au fost făcute pentru utilizatorii de calculatoare personale, considerați a fi cei care cunosc cel mai puțin modul de funcționare al unui sistem. Dezvoltatorii de aplicații pentru PC fac eforturi incredibile pentru a realiza cel mai frumos spațiu de lucru sau pentru a face ca sistemul dumneavoastră Linux să arate ca fostul MS Windows sau Apple. Ultimele versiuni includ suport pentru accelerare 3D și pentru dispozitive USB, realizarea de actualizări ale sistemului sau pachetelor printre-o singură apăsare a butoanelor mausului, și altele de acest fel. Linux-ul are toate aceste

lucruri și se încearcă prezentarea tuturor serviciilor disponibile într-o infățișare logică pe care și oamenii obișnuiau să o pot înțelege. Mai jos este o listă care conține câteva exemple excepționale; aceste saituri au multe capturi de ecran care vă vor oferi o idee despre cum poate arăta un Linux pe un calculator personal:

- <http://www.gnome.org>
 - <http://www.kde.org/screenshots/>
 - <http://www.openoffice.org>
 - <http://www.mozilla.org>
 - <http://www.ubuntu.com>
-

1.3. Are Linux-ul viitor?

1.3.1. Sursa deschisă

Ideeia din spatele programelor Sursă deschisă este una destul de simplă: atunci când programatorii pot citi, distribui și schimba codul, codul se va maturiza. Oamenii îl pot adapta, repara, explora defectele și pot face aceste lucruri la o viteză care spulberă performanțele dezvoltatorilor de programe din cadrul companiilor convenționale. Aceste programe vor fi mai flexibile și de o calitate mai bună decât programele care au fost dezvoltate pe canale convenționale, deoarece le-au testat mai multe persoane, în condiții diferite de cele în care dezvoltatorii de programe proprietare pot să o facă.

Inițiativa Sursă deschisă (Open Source) a început să se impună în lumea comercială și, foarte încet, distribuitorii înțeleg punctul de vedere al acesteia. În timp ce mulți oameni din lumea academică și tehnică erau deja convingați de aproape 20 de ani că acesta este drumul de urmat, distribuitorii din lumea comerțului au avut nevoie de aplicații ca Internetul pentru a realiza că și ei pot profita de pe urma Sursei deschise. Acum Linux-ul a trecut de stadiul în care era un sistem aproape exclusiv academic, folosit doar unei mâini de oameni cu aptitudini tehnice.

Linux-ul asigură mai mult decât un sistem de operare: există o întreagă infrastructură care susține efortul concentrat necesar realizării unui sistem de operare, scrierii și testării de programe pentru acesta, aducerii roadelor întregului efort către utilizatori, asigurarea întreținerii, suportului, actualizărilor și personalizărilor, etc. Astăzi, Linux-ul este gata să accepte provocările unei lumi aflate într-o transformare rapidă.

1.3.2. Zece ani de experiență la dispoziția dumneavoastră

Pe când Linux-ul este, probabil, cea mai bine cunoscută inițiativă a Sursei deschise, există un alt proiect care a contribuit enorm la popularitatea sistemului Linux. Acest proiect poartă numele de SAMBA și contribuția lui este aplicarea ingineriei inverse asupra protocolului Server Message Block (SMB)/Common Internet File System (CIFS), utilizat la gestionarea fișierelor și tipăririi pe mașinile compatibile PC, integrat nativ de MS Windows NT și OS/2, precum și de Linux. Sunt disponibile pachete pentru aproape orice sistem, ele asigurând soluții de interconectare în medii diferite, folosind protocoalele MS Windows: cu servere pentru transfer de fișiere și servere pentru tipărire, compatibile cu Windows (incluzând WinXP).

Poate că mai de succes decât proiectul SAMBA este proiectul pentru server Apache HTTP. Serverul rulează pe UNIX, Windows NT și multe alte sisteme de operare. Cunoscut la început ca „serverul A PatCHy”, pe baza unui cod existent și a unei serii de „fișiere de peticire”, numele pentru codul ajuns la maturitate merită să fie legat de tribul de americani năvârți Apache, bine cunoscuți pentru însușirile lor superioare în strategia confruntărilor și adaptarea extraordinară la mediu.

Apache s-a dovedit a fi mult mai rapid, mai stabil și oferă mai multe facilități decât alte programe pentru servere web. Apache rulează pe saituri care au milioane de vizitatori pe zi, în timp ce nu este asigurat suport oficial din partea dezvoltatorilor, însă răspunsurile la problemele întâlnite sunt furnizate de comunitatea utilizatorilor. Asistență contra cost este asigurată de terțe părți.

În categoria aplicațiilor pentru birou, sunt disponibile câteva clone ale suitei MS Office, care merg de la o implementare parțială până la cea totală a programelor existente pe stații de lucru MS Windows. Aceste inițiative au ajutat mult Linux-ul în efortul de a se face acceptabil pentru piața calculatoarelor personale, deoarece utilizatorii nu au nevoie de pregătire suplimentară pentru a lucra cu noile sisteme.

Odată cu pachetele pentru calculatoarele personale vine și atenția acordată utilizatorilor de rând, manifestată și prin implementarea cererilor acestora, care devin mai complexe și mai solicitante pe zi ce trece.

Comunitatea Sursei deschise, formată în principal din oameni care au contribuit mai bine de jumătate de deceniu, garantează poziția de jucător important a Linux-ului pe piața calculatoarelor personale, precum și pe cea a aplicațiilor generale din domeniul IT. Angajați plătiți, împreună cu voluntari, lucrează cu sărăcini pentru ca Linux-ul să-și mențină poziția pe piață. Mai mulți utilizatori înseamnă mai multe întrebări. Comunitatea Sursei deschise se asigură că răspunsurile vin neîncetat și supraveghează calitatea acestor răspunsuri cu o atitudine critică, adăugând astfel mai multă stabilitate și accesibilitate.

Acest ghid nu își propune să listeze toate programele existente pentru Linux, deoarece există zeci de mii de pachete. Prin acest curs vă vom prezenta cele mai utilizate pachete, care sunt aproape în totalitate disponibile gratuit. Pentru a vă elibera de teama pe care o simte un utilizator începător, vă prezentăm o captură a unui program foarte dorit de dumneavoastră. Puteți observa că au fost depuse toate eforturile pentru ca cei care renunță la Windows să se simtă ca acasă:

Figura 1-1. Foi de calcul OpenOffice compatibile cu MS Office

The screenshot shows a spreadsheet titled 'msedu.xls - OpenOffice.org 1.0.1'. The file path is 'file:///home/jenny/Documents/sales/msedu.xls'. The spreadsheet contains three rows of data:

	A	B	C	D	E
2237	810-01488	SQL Svr Enterprise Edtn Italian Lic/SA MVL	3 Yr(s) Remaining	75	Servers
2238	810-01570	SQL Svr Enterprise Edtn Spanish SA MVL	2 Yr(s) Remaining	30	Servers

1.4. Proprietățile Linux-ului

1.4.1. Avantajele Linux-ului

Multe din avantajele Linux-ului sunt o consecință a originilor sale, cu rădăcini adânci în UNIX, cu excepția primului, desigur:

- Linux-ul este gratuit.

Precum berea moacă, se spune. Dacă nu dorîți să cheltuiți nimic, nu trebuie să plătiți nici măcar prețul unui CD gol. Linux-ul poate fi descărcat în întregime de pe Internet absolut gratuit. Nu există taxe de înregistrare, costuri pe număr de utilizatori, iar actualizările sunt gratuite și codul sursă este disponibil în cazul în care dorîți să schimbați comportamentul sistemului.

Mai mult, Linux-ul este gratuit și liber, după cum se afirmă mai jos:

Licența folosită în general este Licența publică GNU (GPL). Ea spune că oricine dorește să schimbe Linux-ul și să, eventual, redistribuie această versiune modificată, are acest drept, cu o singură condiție, anume codul sursă să fie disponibil și după modificare. Concret, asta înseamnă că puteți lua o imagine a kernelului, să-i adăugați, de exemplu, suport pentru un dispozitiv de teleportare sau de călătorie în timp și să vindeți acest nou cod atât timp cât clientii dumneavoastră pot avea o copie a acestui cod.

- Linux-ul poate fi portat pe orice fel de dispozitive.

Un vânzător care dorește să comercializeze un tip nou de procesor și care nu știe ce fel de sistem de operare va rula pe acest echipament (să spunem că este vorba despre un procesor instalat în autoturismul dumneavoastră sau în mașina de spălat vase), poate lua și adapta echipamentului său un

kernel Linux, deoarece documentația necesară este disponibilă.

● Linux-ul este făcut pentru a funcționa neîntrerupt.

La fel ca UNIX-ul, un sistem Linux este făcut pentru a rula fără să fim nevoiți să-l repornim tot timpul. De aceea, o mulțime de sarcini sunt executate noaptea sau programate automat pentru momentele de acalmie, ceea ce înseamnă o mai mare disponibilitate pe parcursul perioadelor mai solicitante și o folosire echilibrată a sistemului. Această proprietate permite Linux-ului să fie potrivit pentru situații în care oamenii nu au timpul sau posibilitatea de a controla sistemele lor zi și noapte.

● Linux-ul este de încredere și versatil.

Modelul de securitate folosit pentru Linux este bazat pe conceptul de siguranță din UNIX, despre care se știe că este robust și de o calitate dovedită odată cu trecerea timpului. Dar Linux-ul nu este potrivit doar ca fortăreață împotriva atacurilor de pe Internet: el se va adapta în mod egal oricărei situații, însă folosind aceleași standarde ridicate în privința securității. Stația dumneavoastră folosită la dezvoltarea de programe sau pentru controlul rețelei va fi la fel de securizată precum firewall-ul.

● Linux-ul este scalabil.

De la un dispozitiv Palmtop cu 2 MO de memorie la un mediu de stocare de ordinul petaoctetilor, care înglobează sute de noduri: adăugați sau îndepărtați pachetele potrivite și Linux-ul se va potrivi tuturor. Nu mai aveți nevoie de un supercalculator, deoarece puteți folosi Linux-ul pentru lucruri deosebite, apelând doar la pachetele furnizate împreună cu sistemul. Dacă dorîți să faceți lucruri mai puțin solicitante, ca, de exemplu, să realizați un sistem de operare pentru un procesor dedicat sau să refolosiți vechea dumneavoastră mașină 486, Linux-ul va putea fi folosit și pentru aceste lucruri.

● Sistemul de operare Linux și cele mai multe aplicații pentru acesta se bucură de un timp scurt în care sunt puse la punct deficiențele semnalate.

Deoarece Linux-ul este dezvoltat și testat de mii de oameni, atât erorile, cât și cei care le pot repara sunt foarte ușor de găsit. Adesea se întâmplă să treacă doar câteva ore între descoperirea și repararea unei erori.

1.4.2. Dezavantajele Linux-ului

● Se poate spune că există prea multe distribuții.

„Câte capete, atâtea păreri”, spune un vechi proverb roman. Cu cât mai mulți oameni sunt implicați, vor exista mai multe păreri. La o primă vedere, numărul distribuțiilor de Linux poate părea însăprimător sau ridicol, în funcție de convingerile fiecăruia. Dar înseamnă și că oricine poate găsi tot ceea ce dorește. Nu trebuie să fiți un expert pentru a găsi ceea ce vi se potrivește.

Atunci când întrebăți un utilizator de Linux care este cea mai bună distribuție, în general vi se va răspunde că aceea pe care o folosește el. Așa că pe care să o alegeti? Nu vă bateți capul prea mult cu acest lucru: toate distribuțiile conțin, mai mult sau mai puțin, același set de pachete de bază. Peste această bază sunt adăugate programe speciale de la terțe părți, pentru a face, de exemplu, TurboLinux mai potrivit pentru întreprinderi mici și mijlocii, RedHat pentru servere și SuSE (sau Ubuntu^{n.t.}) pentru stațiiile de lucru. Totuși, diferențele sunt mai degrabă superficiale. Cea mai bună strategie este să testați mai multe distribuții; din nefericire, nimeni nu are timp pentru așa ceva. Există însă multe sfaturi pe marginea subiectului alegerii Linux-ului.

O căutare rapidă pe Google (<http://www.google.com/linux>) folosind cuvintele cheie „alegeti-vă distribuția”, va returna zeci de legături către sfaturi demne de luat în considerare. Capitolul CUM SĂ instalați va discuta, de asemenea, despre alegerea distribuției.

● Linux-ul nu este foarte prietenos cu utilizatorii, fiind considerat confuz de către cei începători.

Trebuie spus că Linux-ul și, în special, inima sistemului, sunt mai greu accesibile utilizatorilor decât MS Windows, chiar mai greu accesibile decât MacOS, dar, pentru a-i spori popularitatea, au fost depuse eforturi considerabile pentru a face Linux-ul mai ușor de folosit de către utilizatorii începători. Zilnic sunt puse la dispoziție din ce în ce mai multe informații, precum acest ghid, pentru a ajuta la umplerea golurilor din documentația existentă pentru utilizatorii de toate nivelurile.

● Sunt produsele Sursei deschise demne de încredere?

Cum poate ceva care este gratuit, să fie și de încredere? Utilizatorii de Linux au măcar libertatea de a alege dacă să folosească Linux sau nu, ceea ce le oferă un avantaj major în comparație cu utilizatorii programelor proprietare, care nu se bucură de acest gen de libertate.

După îndelungi perioade de testare, cei mai mulți utilizatori ai Linux-ului au ajuns la concluzia că Linux-ul nu este doar la fel de bun, fiind chiar, în multe cazuri, mai bun și mai rapid decât soluțiile tradiționale. Dacă Linux-ul nu ar fi demn de încredere, ar fi dispărut de multă vreme, necunoscând

popularitatea pe care o are, cu milioane de utilizatori.

Acum utilizatorii pot să influențeze comportamentul sistemelor proprii și să împărtășească experiența lor comunității, astfel sistemul fiind perfecționat zi după zi. Este un proiect permanent, e adevărat, dar, într-un mediu aflat într-o continuă schimbare, Linux-ul este un proiect care continuă să urmărească perfecțunea.

1.5. Aromele Linux

1.5.1. Linux și GNU

Cu toate că există numeroase implementări ale Linux-ului, veți găsi multe similitudini în acestea, deoarece mașinile Linux sunt ca un întreg alcătuit din piese pe care le alăturăți în funcție de nevoile pe care le aveți sau de viziunea proprie. Instalarea sistemului este doar începutul unei relații pe termen lung. Atunci când credeți că ati pus la punct un sistem care rulează superb, Linux-ul vă va stimula imaginația și creativitatea, iar cu cât înțelegeți mai multe despre ceea ce puteți face cu sistemul, veți încerca și mai mult să-i redefiniți limitele.

Linux-ul poate să vă apară ca fiind diferit în funcție de distribuții, echipamentul fiecărui și preferințele personale, dar fundamentele de la care toate interfețele grafice și alte unelte sunt construite, sunt aceleași. Linux-ul se bazează pe uneltele GNU (GNU nu este UNIX), care asigură un set de standarde pentru alcătuirea și gestionarea sistemului. Toate uneltele GNU sunt sursă deschisă, aşa că pot fi instalate pe orice sistem.

Cele mai multe distribuții vă oferă pachete precompilate ale celor mai cunoscute unelte, cum ar fi pachetele RPM de la RedHat și pachetele Debian (denumite și deb sau dpkg) din distribuția cu același nume (sau din Ubuntu^{n.t.}), astfel că nu trebuie să fiți un bun programator pentru a instala un pachet în sistem. Dacă sunteți și dacă vă place să faceți lucrurile dumneavoastră însivă, vă veți bucura enorm de Linux, pentru că majoritatea distribuțiilor vin cu un set complet de unelte pentru dezvoltatori, care permit instalarea de programe noi din codul sursă pur. Această abordare vă ajută să instalați și programele care nu sunt împachetate într-o formă potrivită sistemului dumneavoastră.

Iată o listă a unor programe GNU mai cunoscute:

- Bash: consola GNU
- GCC: compilatorul C al GNU
- GDB: unealta pentru căutarea erorilor
- Coreutils: un set de unelte de bază în stilul UNIX-ului, precum **ls**, **cat** și **chmod**
- Findutils: utilitar pentru căutarea și cercetarea fișierelor
- Fontutils: folositor la convertirea fonturilor dintr-un format în altul și realizarea de fonturi
- The Gimp: programul pentru gestionarea imaginilor din GNU
- Gnome: interfață grafică cu utilizatorul
- Emacs: un editor puternic
- Ghostscript și Ghostview: interprotoare și interfață grafică pentru fișierele PostScripts
- GNU Photo: program pentru interacțiunea cu aparatelor foto digitale
- Octave: un limbaj de programare, destinat inițial pentru executarea de calcule numerice și procesări de imagine
- GNU SQL: baze de date relaționale
- Radius: server pentru autentificare la distanță și gestiune rețea
- ...

Multe aplicații comerciale sunt disponibile pentru Linux, dar pentru mai multe informații despre ele vă vom îndruma către documentația acestora. Pe parcursul acestui ghid vom discuta numai programele disponibile liber, gratuit, care vin (de obicei) sub licență GNU.

Pentru a instala pachetele lipsă sau pachete noi, aveți nevoie de un tip de program pentru gestionarea pachetelor. Cele mai cunoscute implementări includ RPM și dpkg. RPM este numit Programul de gestionare a pachetelor RedHat (RedHat Package Manager), și este utilizat de mai multe sisteme Linux, deși numele nu sugerează acest lucru. Dpkg este programul de gestionare a pachetelor Debian, care utilizează o interfață numită **apt-get**, care poate gestiona și pachete RPM. Novell Ximian Red Carpet este o

implementare a programului RPM, care dispune de o interfață grafică, realizată de către compania Novell. Alte companii implicate în comercializarea de programe pot pune la dispoziție propriile lor proceduri de instalare, câteodată asemănătoare cu InstallShield-urile cunoscute din MS Windows sau alte platforme. Pe măsură ce veți avansa în Linux, veți ajunge să cunoașteți unul sau mai multe din aceste programe.

1.5.2. GNU/Linux

Kernelul Linux (coloana vertebrală a sistemului, vezi [Secțiunea 3.2.3.1](#)) nu este parte a proiectului GNU, dar utilizează aceeași licență cu programele GNU. Marea majoritate a utilităților și uneltele pentru dezvoltare (mușchii sistemului), care nu sunt specifice Linux-ului, sunt luate din proiectul GNU. Deoarece orice sistem care se dorește a fi utilizabil trebuie să conțină atât kernelul, cât și un set minimal de utilități, există voci care cer ca un asemenea sistem să fie numit un sistem GNU/Linux.

În ideea de a obține cel mai înalt grad de independentă față de sisteme, acesta este felul de Linux pe care-l vom discuta pe parcursul acestui curs. Dacă nu vom vorbi despre un sistem GNU/Linux, vor fi menționate numele distribuției, versiunii sau programului.

1.5.3. Ce distribuție trebuie să instalez?

Înainte de instalare, factorul determinant este echipamentul de care dispuneți. Deoarece fiecare distribuție de Linux conține pachetele de bază, și poate fi configurată pentru a răspunde aproape oricărora cerințe (deoarece toate utilizează kernelul Linux), trebuie doar să aveți în vedere dacă o distribuție anume va rula pe echipamentul dumneavoastră. LinuxPPC de exemplu, a fost destinat pentru Apple și alte PowerPC, așa că nu va rula pe calculatoare compatibile x86. LinuxPPC rulează pe noile Mac-uri, dar nu poate fi folosit pentru câteva din modelele mai vechi, cu o tehnologie bus învechită. Alt caz care pune unele probleme este cel al echipamentelor Sun, care pot fi bazate pe un procesor mai vechi SPARC CPU sau pe mai noi UltraSparc, fiecare necesitând versiuni diferite de Linux.

Câteva distribuții sunt optimizate pentru un anumit procesor, cum ar fi procesoarele Athlon, având și posibilitatea de a rula decent pe procesoarele compatibile cu Intel 486, 586 și 686. Distribuțiile pentru procesoare speciale pot fi mai puțin fiabile, întrucât sunt testate de mai puțini oameni.

Cele mai multe distribuții oferă un set de programe pentru calculatoarele personale, în pachete speciale, care conțin kernele optimizate pentru procesoarele compatibile Intel x86. Aceste distribuții sunt testate extensiv și actualizate periodic, cu o atenție mare acordată implementării în condiții de siguranță pe servere și unei proceduri de instalare și actualizare care să nu ridice probleme deosebite. Exemple care pot fi date sunt Debian, Ubuntu, Fedora, SuSE și Mandriva, care sunt, de departe, cele mai populare sisteme Linux, considerate totodată ușor de folosit de utilizatorii începători și care nu limitează profesioniștii în dorința lor de a obține totul de la un sistem. Linux-ul rulează bine și pe laptopuri sau servere. Drivere pentru dispozitivele noi sunt incluse numai după o testare îndelungată, ceea ce adaugă stabilitate sistemului.

Mediul de lucru cel mai des întâlnit este Gnome, dar alții pot furniza KDE preinstalat. Atât Gnome, cât și KDE sunt disponibile tuturor distribuțiilor principale de Linux. Alți administratori pentru spațiul de lucru sunt disponibili utilizatorilor mai avansați.

Procesul standard de instalare permite utilizatorilor să aleagă între mai multe configurații predefinite, ca, de exemplu, pentru stații de lucru, caz în care sunt instalate toate pachetele pentru sarcinile zilnice și dezvoltare, sau pentru servere, caz în care pot fi selectate mai multe unele pentru serviciile de rețea. Utilizatorii experimentați pot instala orice combinație de pachete doresc pe parcursul procesului inițial de instalare.

Scopul acestui ghid este ca el să fie util tuturor distribuțiilor de Linux. Pentru confortul dumneavoastră, sfătuim pe începători să se limiteze la o distribuție principală, care suportă cât mai multe dispozitive și oferă cât mai multe aplicații. Iată câteva alegeri foarte bune pentru acești:

- [Fedora Core](#)
- [Debian](#)
- [Ubuntu^{n.t.}](#): un sistem de operare care rulează și de pe CD-ROM, fără instalare
- [SuSE Linux](#)
- [Mandriva \(fostul MandrakeSoft\)](#)
- [Knoppix](#): un sistem de operare care rulează și de pe CD-ROM, fără instalare

1.6. Sumar

În acest capitol am învățat că:

- Linux-ul este o implementare a UNIX-ului.
 - Sistemul de operare Linux este scris în limbajul de programare C.
 - "De gustibus et coloribus non disputandum est": există câte un Linux pentru fiecare.
 - Linux-ul folosește uneltele GNU, un set de unelte standard disponibile oricui, pentru manipularea sistemului de operare.
-

1.7. Exerciții

Un exercițiu practic pentru începători: instalați Linux-ul pe calculatoarele dumneavoastră. Citiți manualul distribuției alese și/sau CUM SĂ instalăm și punem să pe treabă.



Citiți documentația!

Cele mai multe erori survin pentru că informațiile furnizate pe parcursul instalării nu sunt citite. Citirea acestor mesaje cu atenție este primul pas pe drumul spre succes.

Lucruri pe care trebuie să le știți ÎNAINTE de a începe o instalare a Linux-ului:

- Va rula această distribuție pe calculatorul meu?

Verificați la <http://www.tldp.org/HOWTO/Hardware-HOWTO/index.html> atunci când aveți dubii despre compatibilitatea cu echipamentul dumneavoastră.

- Ce tip de tastatură am (numărul de taste, poziționarea acestora)? Ce fel de mouse (serial, paralel, numărul de butoane)? Câtă MO de RAM am?
- Voi alege o instalare pentru o stație de lucru sau pentru un server, ori va trebui să selectez pachetele necesare eu însuși?
- Voi instala de pe discul fix (hard disk), de pe un CD-ROM, sau voi utiliza rețea? Va trebui să adaptez BIOS-ul pentru oricare dintre acestea? Metoda de instalare necesită un disc de pornire?
- Linux-ul va fi singurul sistem instalat pe calculator, sau va fi o instalare dual-boot? Ar trebui să fac o partitie mai mare pentru a instala pe ea sisteme virtuale mai târziu, sau este o instalare virtuală de la început?
- Calculatorul este legat la rețea? Care este numele gazdei, adresele de IP? Există servere tip gateway sau alte dispozitive importante de rețea cu care sistemul meu ar trebui să comunice?



Linux-ul se așteaptă să fie conectat

Neutilizarea rețelei sau configurarea incorRECTă a acesteia poate încetini pornirea sistemului.

- Calculatorul acesta este un gateway/router/firewall? (Dacă trebuie să vă gândiți prea mult la această întrebare, probabil că nu este.)
- Partiționarea: lăsați programul de instalare să o facă pentru dumneavoastră de această dată, vom discuta partiționarea în detaliu în [Capitolul 3](#). Există documentație specifică fiecărui sistem, dacă doriți să știți totul despre acest lucru. Dacă distribuția aleasă nu vă oferă posibilitatea de partiționare automată, asta înseamnă, de regulă, că nu este potrivită începătorilor.
- Calculatorul va porni în modul text sau în cel grafic?
- Gândiți-vă la o parolă bună pentru administratorul acestui sistem (root). Faceți un cont de utilizator obișnuit (care nu are acces privilegiat la calculator – *nu este necesar pentru Ubuntu, deoarece această distribuție utilizează un sistem care cere autentificare pentru sarcinile realizate ca administrator* n.t.).
- Am nevoie de un disc pentru recuperare? (recomandat).

- Ce limbă doresc pentru sistemul meu?

Lista întreagă cu verificările pe care ar trebui să le parcurgeți poate fi găsită la <http://www.tldp.org/HOWTO/Installation-HOWTO/index.html>.

În capitolele următoare vom afla dacă instalarea este una reușită.

Capitolul 2. Să Începem

Pentru ca acest ghid să vă fie cu adevărat folositor, vom începe imediat cu un capitol practic despre autentificarea într-un sistem Linux și anumite lucruri de bază.

Vom discuta:

- ◆ Autentificarea la sistem
- ◆ Ieșirea din sistem
- ◆ Modul text și cel grafic
- ◆ Schimbarea parolei
- ◆ Navigarea prin sistemul de fișiere
- ◆ Determinarea tipului de fișiere
- ◆ Deschiderea fișierelor text
- ◆ Căutarea de ajutor

2.1. Autentificarea la sistem, activarea interfeței cu utilizatorul și deautentificarea

2.1.1. Introducere

Pentru a lucra direct cu un sistem Linux trebuie să îi furnizați un nume de utilizator și o parolă. Întotdeauna trebuie să vă autentificați față de sistem. După cum am arătat deja în exercițiul de la Capitolul 1, cele mai multe calculatoare care au ca sistem de operare Linux-ul pot lucra în două moduri de bază: ori în modul consolei text, rapid dar sobru, care arată ca un DOS cu maus, cu facilități care asigură realizarea de sarcini multiple și conectarea mai multor utilizatori, ori în modul grafic, care arată mai bine, dar consumă mai multe resurse.

2.1.2. Modul grafic

Acesta este modul normal pentru calculatoarele de astăzi. Veți ști că vă conectați la un calculator care folosește modul grafic atunci când prima dată vă este cerut numele de utilizator, apoi, într-o altă fereastră, vi se cere să introduceți parola.

Pentru a vă autentifica, puneți cursorul mausului în fereastra de conectare, tastați numele de utilizator și parola, apăsați OK sau tasta **Enter**.



Fiți foarte atenți când folosiți contul de administrator (root)!

În general, se consideră că este o idee proastă să vă conectați în modul grafic folosind contul de *root*, contul administratorului de sistem, deoarece utilizarea modului grafic include rularea mai multor programe, ce vor avea mai multe permisiuni dacă vă autentificați ca *root*. Pentru a menține cota de risc cât mai jos posibil, folosiți un cont de utilizator normal pentru a vă autentifica în modul grafic. Însă există și alte motive pentru a vă folosi de acest sfat, ca o regulă a conectării la un sistem Linux: autentificați-vă ca administrator numai atunci când vi se solicită permisiuni speciale.

După ce introduceți numele de utilizator și parola, poate trece puțin timp până ce mediul grafic este pornit, în funcție de viteza procesorului, de programele pe care le folosiți sau de setările personale.

Pentru a continua, trebuie să deschideți o *fereastră terminal* sau *xterm* pe scurt (X fiind numele prin care se arată că anumite programe suportă interfață grafică). Acest program poate fi găsit în Aplicații → Utilități, Sistem sau Internet, în funcție de ce administrator grafic folosiți. Pot exista iconițe (pictograme) pe care să le folosiți ca scurtături pentru a obține terminalul, iar dacă apăsați pe butonul din dreapta al mausului, pe spațiul de lucru, vi se va prezenta un meniu care poate conține și o aplicație terminal.

Dacă vă plimbați prin meniu, veți observa că pot fi făcute foarte multe lucruri fără a introduce comenzi de la tastatură. Pentru cei mai mulți utilizatori, vechea metodă de lucru cu mausul va fi suficientă pentru a lucra cu calculatorul. Dar acest ghid se adresează viitorilor administratori de sistem și de rețea, care vor avea nevoie să lucreze cu măruntalele sistemului. Ei au nevoie de o unealtă mai puternică decât cele pe

care le pot acționa cu mouse-ul, pentru a face față tuturor sarcinilor. Această unealtă este consola și, din mediul grafic, puteți să o activați prin deschiderea unei ferestre terminal.

Terminalul este panoul de control al sistemului dumneavoastră. Aproape orice lucru din cele ce urmează poate fi făcut folosind această unealtă text, simplă dar puternică. O fereastră terminal ar trebui să arate întotdeauna un prompter standard, care afișează numele utilizatorului și directorul curent în care se lucrează, reprezentat de simbolul ~ :

Figura 2-1. Fereastra terminal



O altă formă comună pentru un prompter este următoarea:

```
[user@host dir]
```

În exemplul de mai sus, *user* va fi numele dumneavoastră de utilizator, *host* numele calculatorului pe care lucrăți și *dir* indică locul în care vă aflați în sistemul de fișiere.

Mai târziu vom vorbi despre promptere și despre comportamentul acestora în detaliu. Acum este suficient să știți că ele afișează tot felul de informații, dar nu sunt parte a comenziilor pe care le dați sistemului.

Pentru a ieși din sistem în mediul grafic, trebuie să închideți toate ferestrele terminal și alte aplicații pe care le-ați deschis. După care apăsați pe iconița de ieșire (Log Out) din meniu. Nu este neapărat necesar să închideți totul, sistemul poate face acest lucru în locul dumneavoastră, dar managerul de sesiune poate restaura acele aplicații lăsate deschise atunci când vă reautentificați, ceea ce va lua mai mult timp, iar pe de altă parte, deschiderea lor nu este întotdeauna ceea ce doriți să se întâpte. Acest comportament se poate configura, în funcție de preferințele fiecăruia.

Atunci când vedeați din nou fereastra de autentificare, care vă cere numele de utilizator și parola, înseamnă că v-ați deconectat.



Gnome sau KDE?

Am menționat ambele medii grafice de lucru, Gnome și KDE, de mai multe ori. Ele sunt cei mai cunoscuți administratori grafici prin care vă puteți gestiona spațiul de lucru, deși există mult mai mulți. Orice administrator grafic alegeți este bun – atât timp cât știți să deschideți un terminal. Totuși, noi vom continua să ne referim la Gnome și la KDE pentru a vă prezenta unele metode prin care puteți realiza anumite sarcini.

2.1.3. Modul text

Veți ști dacă sunteți în modul text atunci când tot ecranul este negru, afișând litere (în cele mai multe cazuri, albe). Un ecran de autentificare în modul text va afișa, de regulă, anumite informații despre calculatorul la care lucrăți, numele acestuia și un prompter care vă cere să vă conectați:

```
RedHat Linux Release 8.0 (Psyche)
blast login: _
```

Conecțarea este diferită de cea din modul grafic, în acest caz trebuind să apăsați tasta **Enter** după ce introduceți numele de utilizator, deoarece nu sunt butoane pe ecran pe care să le puteți apăsa cu mouse-ul.

Apoi trebuie să introduceți parola, urmată de apăsarea din nou a tastei **Enter**. Nu veți observa nici o indicație a ceea ce introduceți de la tastatură, nici măcar asteriscuri și nu veți vedea cursorul mișcându-se. Este un lucru normal în Linux și este aşa din cauza cerințelor de securitate.

Atunci când sistemul vă recunoaște ca utilizator legitim, puteți primi mai multe informații, numite *mesajele zilei*, care pot fi orice. În plus, este răspândită practica prin care sistemele UNIX să afișeze o considerație care să conțină anumite gânduri, înțelepte sau nu (depinde de fiecare). După acestea, va porni consola, indicată de același prompter pe care îl primiți în modul grafic.



Nu vă autentificați ca administrator (root)

Este valabil și în modul text: autentificați-vă ca root numai pentru a face setări și configurări ale sistemului care necesită privilegiilor de administrator, precum adăugarea de utilizatori, instalarea pachetelor de programe, configurarea rețelei și altele de acest fel. Odată ce ati terminat, părăsiți imediat contul special și continuați-vă munca pe un cont de utilizator obișnuit. Ca alternativă, unele sisteme, precum Ubuntu, vă forțează să folosiți **sudo**, pentru a nu avea nevoie de acces direct la un cont de administrator.

Deconectarea se face prin comanda **logout**, urmată de apăsarea tastei **Enter**. Sunteți deconectat atunci când este afișat din nou ecranul de autentificare.



Butonul de oprire

Deoarece Linux-ul nu a fost proiectat să fie oprit fără a trece prin procedurile adecvate de închidere a sistemului, apăsarea butonului de oprire lansează aceste proceduri pe *sistemele mai noi*. Oprirea unui sistem mai vechi fără a trece prin procedurile de închidere poate duce la distrugerea sistemului. Dacă dorîți să fiți siguri că acționați corect, folosiți butonul de Oprire atunci când vă aflați în interfața grafică, iar atunci când sunteți în ecranul de autentificare (în care trebuie să introduceți un nume de utilizator și o parolă) căutați un buton de oprire.

Acum, pentru că știm cum să ne autentificăm la sistem și să ieșim din el, suntem gata pentru primele noastre comenzi.

2.2. Lucruri de bază

2.2.1. Comenzile

Vă prezintăm câteva comenzi, de care avem nevoie pentru început. Le vom discuta mai târziu în detaliu.

Tabelul 2-1. Comenzi de bază

Comanda	Ce execută
ls	Afișează o listă cu fișierele din directorul de lucru, precum comanda dir din DOS
cd director	schimbă directorul
passwd	schimbă parola pentru utilizatorul curent
file filename	afișează tipul fișierului al cărui nume este filename
cat textfile	afișează conținutul unui fișier text pe ecran
pwd	afișează directorul în care se lucrează
exit sau logout	terminare sesiune
man comandă	citiți paginile din manualul comenzi comandă
info comandă	citiți paginile de informații asupra comenzi comandă
apropos string	căutați în baza de date whatis

2.2.2. Generalități

ACESTE COMENZI LE VEȚI INTRODUCEREA DUPĂ PROMPTER, ÎNTR-O FEREASTRĂ TERMINAL, DACĂ SUNTEȘTI ÎN MODUL

grafic, sau în modul text, urmate apoi de apăsarea tastei **Enter**.

Comenzile pot fi introduse ca atare, cum este cazul comenzi **ls**. O comandă se comportă diferit atunci când specificați o *opțiune*, care este, de regulă, precedată de caracterul (-), precum în **ls -a**. Același caracter al unei opțiuni poate avea un alt înțeles dacă este folosită în altă comandă. Programele GNU pot avea opțiuni lungi, precedate de două caractere (-), precum **ls --all**. Alte comenzi nu au nici o opțiune.

Argumentele unei comenzi reprezintă specificații despre obiectul asupra căruia comanda trebuie să acționeze. Un exemplu ar fi **ls /etc**, unde directorul **/etc** este argument al comenzi **ls**. Această comandă înseamnă că dumneavoastră dorîți să vedeti conținutul acelui director, în loc de ceea ce afișează comanda simplă **ls** urmată de tastarea **Enter**, adică listarea conținutului directorului în care ne aflăm. Anumite comenzi au nevoie neapărat de argumente, aceste argumente fiind opționale în cazul altor comenzi.

Puteți verifica dacă o comandă suportă opțiuni și argumente și care dintre ele sunt valide, consultând online paginile de ajutor pentru comanda dorită, vezi [Sectiunea 2.3](#).

În Linux, la fel ca în UNIX, directoarele sunt separate de caracterul (/), la fel cu cel pe care îl folosim în adresele web (URL). Vom discuta la obiect structura directoarelor mai târziu.

Simbolurile (.) și (...) au înțelesuri speciale în directoare. Vom prezenta mai multe informații despre aceste lucruri în exerciții și pe parcursul următorului capitol.

Încercați să evitați folosirea contului de administrator, root. Puteți lucra pe un cont de utilizator obișnuit, care nu acordă permisiuni speciale, pentru că majoritatea sarcinilor, inclusiv verificarea sistemului, colectarea de informații și altele, pot fi făcute în acest mod. Atunci când este nevoie să adăugați un nou utilizator sau să instalați programe noi, modul preferat prin care obțineți acces ca administrator este schimbarea identității utilizatorului, precum se arată în [Sectiunea 3.2](#) (după cum am mai spus, acest lucru este evitat în Ubuntu prin folosirea **sudo**, comandă care acordă privilegii de root pe o perioadă limitată^{n.t.}).

Aproape toate comenzi din această carte pot fi executate fără privilegii de administrator. În cele mai multe cazuri, atunci când lansăm ca utilizator obișnuit o comandă sau un program care necesită privilegii de root, sistemul va cere parola de administrator. Odată ce ați terminat, încheiați sesiunea care vă acordă aceste privilegii imediat.

Citirea documentației trebuie să vă intre în sânge. În special la început, este important să citiți documentația sistemului, manualele comenziilor de bază, CUM SĂ-urile (HOWTO) și altele de acest fel. Întrucât volumul documentației este enorm, este imposibil să includem toate referințele. Această carte încearcă să vă îndrumă către documentația adecvată fiecărui subiect discutat, pentru a vă stimula obiceiul de a citi paginile manualelor.

2.2.3. Utilizarea facilităților Bash

Anumite combinații de taste vă permit să faceți câteva lucruri mai ușor în consola GNU, Bash, care vine pe aproape orice sistem Linux, vezi [Sectiunea 3.2.3.2](#).

Tabelul 2-2. Combinații de taste în Bash

Tastă sau combinații de taste	Funcție
Ctrl+A	Mută cursorul la începutul unei linii de comandă.
Ctrl+C	Termină programul care rulează și afișează prompterul, vezi Capitolul 4 .
Ctrl+D	Închide sesiunea curentă, similară cu tastarea exit sau logout .
Ctrl+E	Mută cursorul la sfârșitul liniei de comandă.
Ctrl+H	Șterge poziția curentă (similar cu Backspace).
Ctrl+L	Șterge tot ce este afișat în consolă.
Ctrl+R	Caută în istoricul comenziilor, vezi Sectiunea 3.3.3.4 .
Ctrl+Z	Suspendă un program, vezi Capitolul 4 .
Săgeată stânga și Săgeată dreapta	Mută cursorul cu un caracter la stânga sau la dreapta în linia de comandă, pentru a insera caractere în alt loc decât la începutul sau sfârșitul liniei.

Săgeată în sus sau Săgeată în jos	Navigare prin istoricul comenzi. Mergeți la linia pe care dorîți să o repetați, editați-o dacă este necesar, și apăsați Enter .
Shift+PageUp sau Shift+PageDown	Navigare prin terminal (pentru a vedea textul care a „ieșit” din ecran).
Tab	Completarea comenzi sau a numelor de fișiere; atunci când există mai multe posibilități, sistemul vă avertizează cu un semnal audio sau video ori, dacă posibilitățile sunt prea multe, vă întreabă dacă dorîți să le afișeze pe toate.
Tab Tab	Arată posibilitățile pentru completarea unei comenzi sau a unui nume de fișier.

Ultimele două intrări din tabelul de mai sus necesită câteva explicații suplimentare. De exemplu, dacă dorîți să vă mutați în directorul `director_cu_un_nume_foarte_lung`, nu trebuie să tastați tot acel nume foarte lung. Tastați doar comanda `cd dir`, apoi apăsați tasta **Tab** și consola va completa numele în locul dumneavoastră, în cazul în care alte fișiere nu încep cu primele trei litere din numele directorului dumneavoastră. Desigur, dacă nu există alte directoare care încep cu „d”, puteți tasta doar `cd d` și apoi apăsa tasta **Tab**. În cazul în care numele mai multor fișiere încep cu aceleași caractere, consola va semnala acest lucru, iar apăsarea tastei **Tab** de două ori într-un interval scurt de timp va afișa posibilitățile pe care le aveți:

```
your_prompt> cd st
starthere      stuff      stuffit
```

În exemplul de mai sus, dacă tastați „a” după primele două litere și apăsați tasta **Tab**, nu există alte posibilități decât completarea de către consolă a numelui directorului, fără ca dumneavoastră să mai tastați literele „rthere”:

```
your_prompt> cd starthere
```

Desigur, trebuie să apăsați tasta **Enter** pentru a confirma alegerea prezentată de consolă.

În același exemplu, dacă tastați „u” și apoi apăsați tasta **Tab**, consola va adăuga literele „ff” în locul dumneavoastră, dar va protesta din nou, din pricina că sunt posibile mai multe alegeri. Dacă apăsați **Tab Tab** din nou, veți vedea care sunt acestea; dacă veți introduce una sau mai multe caractere care fac sistemul să distingă o posibilitate unică și apăsați din nou **Tab** sau **Enter** atunci când vă aflați la sfârșitul liniei cu numele pe care dorîți să îl alegeti, consola completează numele fișierului și schimbă directorul curent în cel dorit – dacă este, cu adevărat, un nume de director.

Cele prezentate mai sus sunt valabile pentru toate numele de fișiere folosite ca argumente la comenzi.

Același lucru se aplică și la completarea numelor comenzi. Dacă tastați `ls` și apăsați tasta **Tab** de două ori, veți primi o listă cu toate comenzi disponibile (vezi [Secțiunea 3.2.1](#)).

```
your_prompt> ls
ls          lsdev        lspci       lsraid      lsw
lsattr     lsmod        lsgpgot    lss16toppm
lsb_release lsof         lspnp       lsusb
```

2.3. Căutarea de ajutor

2.3.1. Luăți aminte

GNU/Linux încurajează indivizii în atitudinea lor de a nu depinde de nimeni și de nimic. Caracteristica acestui sistem este aceea că există întotdeauna mai multe căi pentru atingerea unui scop. O cale de a primi ajutor este găsirea cuiva care se pricepe, care are răbdare și dorește să fie de ajutor, din comunitatea Linux, dar oricine se aşteaptă ca dumneavoastră să fi încercat una sau mai multe metode din această secțiune, înainte de a întreba pe altcineva, iar modul în care acest punct de vedere este exprimat poate fi mai degrabă unul dur, dacă se dovedește că nu ați urmat această regulă de bază.

2.3.2. Paginile manualelor (man)

Mulți utilizatori începători se tem de paginile man (manualelor), deoarece acestea conțin prea multe informații. Totuși, ele sunt foarte bine structurate, după cum veți vedea din exemplul de mai jos: **man** man.

Citirea paginilor man se face într-un terminal, atunci când sunteți în modul grafic sau direct în modul text, dacă pe acesta îl preferați. Tastați comanda următoare, urmată de **Enter**:

```
yourname@yourcomp ~> man man
```

Documentația pentru **man** va fi afișată în terminal, după ce apăsați **Enter**:

```
man(1)
```

```
man(1)
```

NAME

man - format and display the on-line manual pages
manpath - determine user's search path for man pages

SYNOPSIS

```
man [-acdfFhkKtwW] [--path] [-m system] [-p string] [-C config_file]  
[-M pathlist] [-P pager] [-S section_list] [section] name ...
```

DESCRIPTION

man formats and displays the on-line manual pages. If you specify section, man only looks in that section of the manual.
name is normally the name of the manual page, which is typically the name of a command, function, or file. However, if name contains a slash (/) then man interprets it as a file specification, so that you can do man ./foo.5 or even man /cd/foo/bar.1.gz.

See below for a description of where man looks for the manual page files.

OPTIONS

```
-C config_file  
lines 1-27
```

Navigați la pagina următoare folosind tasta Space. Vă puteți întoarce la pagina precedentă folosind tasta „b”. Când ajungeți la sfârșit, **man** va da comanda de ieșire și consola va returna prompterul. Tastați „q” dacă dorîți să ieșiți din paginile man înainte de a ajunge la sfârșitul lor sau dacă programul nu se închide automat la sfârșitul paginilor.



Pagere

Combinațiile de taste necesare pentru mănuirea paginilor man depind de *pagerul* folosit de distribuția dumneavoastră. Cele mai multe distribuții folosesc **less** pentru a vizualiza aceste pagini și pentru a naviga prin ele. Pentru mai multe informații despre pagere, mergeți la [Sectiunea 3.3.4.2](#).

Fiecare pagină man conține mai multe secțiuni standard, după cum putem observa și în exemplul **man man**:

- Prima linie conține numele comenzii despre care citiți și identificarea secțiunii în care sunt localizate paginile man. Aceste pagini sunt organizate pe capitole. Comenzile pot avea pagini în mai multe secțiuni, de exemplu paginile man din secțiunea utilizatori, paginile din secțiunea administratorului de sistem, paginile man din secțiunea programatorilor.
- Este prezentat numele comenzii și o scurtă descriere, folosite și la indexarea acestor pagini. Puteți căuta în acest index orice dorîți folosind comanda **apropos**.
- Sintaxa comenzii vă furnizează notațiile tehnice ale tuturor opțiunilor și/sau argumentele pe care le suportă comanda respectivă. Puteți să vă gândiți la o opțiune ca la o cale de executare a comenzii. Prin argument înțelegem asupra cui este exercitată comanda respectivă. Anumite comenzi nu au opțiuni sau argumente. Opțiunile și argumentele optionale, adică cele care pot fi lăsate necomplete, sunt puse între simbolurile „[” și „]”.

- Este arătată și o descriere mai detaliată a comenzi.
- Sunt listate opțiunile, împreună cu descrierea acestora. De regulă, opțiunile pot fi combinate. Dacă nu, în această secțiune vi se va spune acest lucru.
- Sunt descrise variabilele care influențează comportamentul comenziîn consolă (nu pentru toate comenzile).
 - Câteodată există și secțiuni specifice comenzi discutate.
 - Referințe către alte pagini man sunt date în secțiunea „SEE ALSO”. Între paranteze este prezentat numărul secțiunii în care găsim comanda despre care se vorbește. Utilizatorii experimentați merg la „SEE ALSO” prin utilizarea comenzi / urmată de SEE și apăsarea tastei **Enter**.
 - Sunt prezentate și informații despre erorile (anomalii) cunoscute și unde să raportați alte erori pe care le-ați putea descoperi.
 - Pot fi afișate informații despre autor și licență.

Anumite comenzi au mai multe pagini man. De exemplu, comanda **passwd** are pagină man în secțiunea 1 și alta în secțiunea 5. Atunci când deschideți paginile man, vă sunt prezentate cele din secțiunea cu cel mai mic număr de identificare. Dacă dorîți să vedeti o altă secțiune decât cea prezentată automat, specificați acest lucru după comanda **man**:

man 5 passwd

Dacă dorîți să vă fie prezentate toate paginile man ale unei comenzi, una după cealaltă, folosiți –a după **man**:

man -a passwd

În acest mod, când ajungeți la sfârșitul primei pagini, prin apăsarea tastei **SPACE**, va fi deschisă pagina man din secțiunea următoare.

2.3.3. Mai multe informații

2.3.3.1. Paginile Info

În cazul comenziilor, ca o completare a paginilor man, puteți citi paginile Info, folosind comanda **info**. Aceste pagini conțin, de regulă, informații mai recente și, într-un fel, mai ușor de utilizat. Paginile man ale anumitor comenzi fac trimitere la paginile Info.

Puteți începe prin tastarea comenzi **info info** într-un terminal:

```
File: info.info,  Node: Top,  Next: Getting Started,  Up: (dir)

Info: An Introduction
*****



Info is a program, which you are using now, for reading
documentation of computer programs. The GNU Project distributes most
of its on-line manuals in the Info format, so you need a program called
"Info reader" to read the manuals. One of such programs you are using
now.

If you are new to Info and want to learn how to use it, type the
command `h' now. It brings you to a programmed instruction sequence.

To learn advanced Info commands, type `n' twice. This brings you to
`Info for Experts', skipping over the `Getting Started' chapter.

* Menu:

* Getting Started::           Getting started using an Info reader.
* Advanced Info::             Advanced commands within Info.
* Creating an Info File::     How to make your own Info file.
---zz-Info: (info.info.gz)Top, 24 lines --Top-----
Welcome to Info version 4.2. Type C-h for help, m for menu item.
```

Folosiți tastele săgeți pentru a naviga prin text și poziționați cursorul pe o linie care începe cu un asterisc și care conține cuvântul cheie despre care dorîți informații, apoi apăsați **Enter**. Folosiți tastele **P** și **N** pentru a merge la subiectul anterior sau următor. Tasta **SPACE** va afișa următoarea pagină, indiferent dacă pe pagina următoare se tratează un subiect nou sau o nouă comandă. Folosiți **Q** pentru ieșire. Programul **info** vă prezintă mai multe informații despre utilizarea sa.

2.3.3.2. Comenzile **whatis** și **apropos**

Un index cu explicații scurte despre comenzi este disponibil prin comanda **whatis**, precum în exemplul de mai jos:

```
[your_prompt] whatis ls
ls                      (1) - list directory contents
```

Se observă explicația, pe scurt, referitoare la comandă (- listează conținutul directorului) și afișarea primei secțiuni din colecția de pagini man care conține o pagină cu prezentarea comenzi **ls**.

Dacă nu știți de unde să începeți și ce pagină de manual să citiți, comanda **apropos** vă dă mai multe informații. Să spunem că nu știți cum să porniți un browser; în acest caz, puteți introduce următoarea comandă:

```
another prompt> apropos browser
Galeon [galeon] (1) - gecko-based GNOME web browser
lynx          (1) - a general purpose distributed information browser
                  for the World Wide Web
ncftp          (1) - Browser program for the File Transfer Protocol
opera          (1) - a graphical web browser
pilot          (1) - simple file system browser in the style of the
                  Pine Composer
pinfo          (1) - curses based lynx-style info browser
pinfo [pman]   (1) - curses based lynx-style info browser
viewres        (1x) - graphical class browser for Xt
```

După apăsarea tastei **Enter** veți vedea mai multe informații legate de navigatoarele (browserele) care sunt în sistemul dumneavoastră; nu doar despre browsere web, ci și despre navigatoare de fișiere și FTP sau browsere pentru documentație. Dacă aveți pachetele pentru dezvoltare instalate, se vor afișa și paginile de manual care prezintă programele în care au fost scrise aceste browsere. În general, o comandă care arată o pagină man în secțiunea întâi, marcată cu „(1)”, poate fi folosită de un utilizator începător. Utilizatorul care a folosit comanda de mai sus poate încerca să introducă în terminal comenzi **galeon**, **lynx** sau **opera**, deoarece este limpede că acestea sunt legate de navigarea pe web.

2.3.3.3. Opțiunea **--help**

Cele mai multe comenzi GNU suportă opțiunea **--help**, care oferă explicații scurte despre utilizarea comenzi și o listă a opțiunilor disponibile. Exemplul următor arată ce se întâmplă în cazul comenzi **cat**:

```
userprompt@host: cat --help
Usage: cat [OPTION] [FILE]...
Concatenate FILE(s), or standard input, to standard output.

-A, --show-all           equivalent to -vET
-b, --number-nonblank    number nonblank output lines
-e                         equivalent to -vE
-E, --show-ends          display $ at end of each line
-n, --number              number all output lines
-s, --squeeze-blank      never more than one single blank line
-t                         equivalent to -vT
-T, --show-tabs          display TAB characters as ^I
-u                         (ignored)
-v, --show-nonprinting    use ^ and M- notation,
                           except for LFD and TAB
--help        display this help and exit
--version     output version information and exit

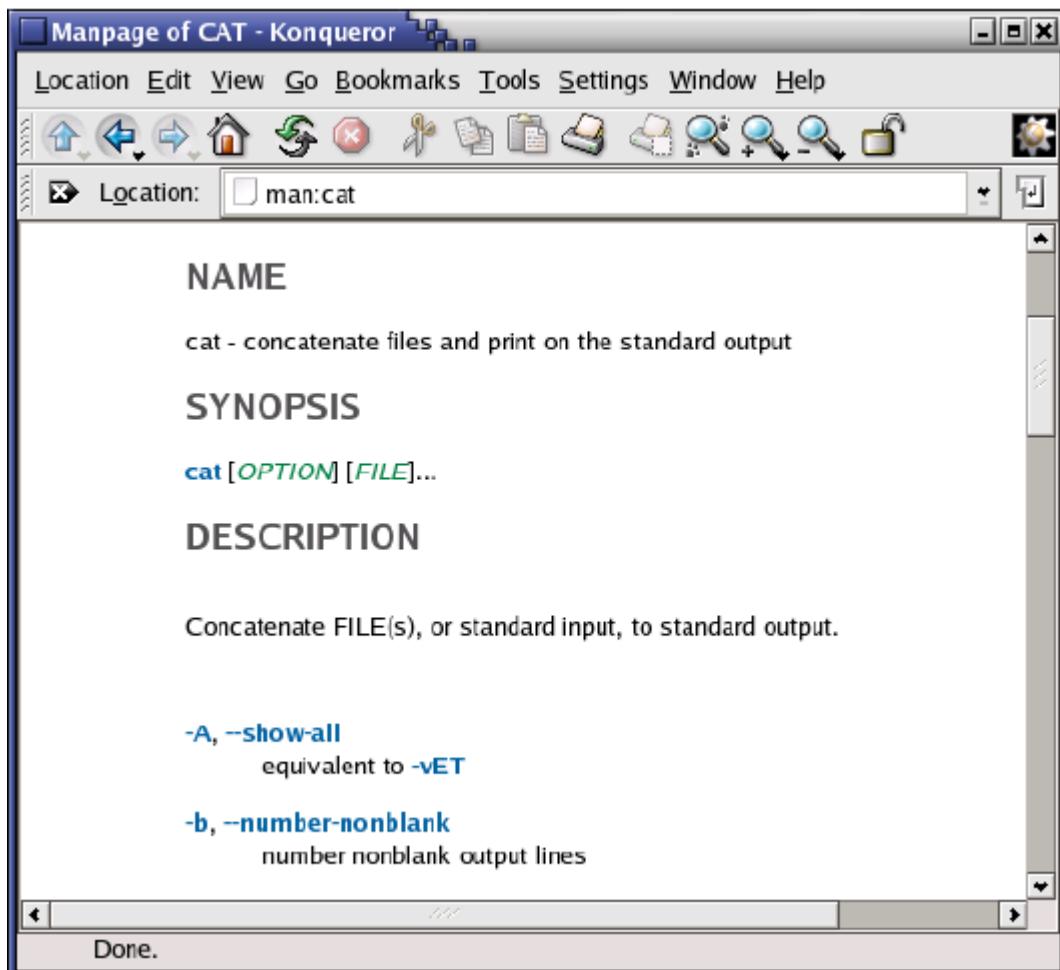
With no FILE, or when FILE is -, read standard input.
```

Report bugs to <bug-textutils@gnu.org>.

2.3.3.4. Ajutor în modul grafic

Nu vă neliniștiți dacă preferați o interfață grafică. Konqueror, administratorul de fișiere din KDE, asigură acces la paginile man și Info într-un mod plin de culoare și care nu dă dureri de cap. Puteți să introduceți în câmpul *Location* (bara de adrese) comanda „info:info” pentru a obține o pagină Info în care vă puteți plimba în căutarea informațiilor despre comanda **info**. În același mod, „man:ls” va afișa pagina man pentru comanda **ls**. Vă puteți baza și pe autocompletarea numelui comenzii: sunt prezentate toate paginile man pentru comenzile care încep cu „ls” într-un meniu pe care îl puteți parcurge în modul derular (scroll-down). Dacă introduceți „info:/dir” în bara de adrese veți primi toate paginile Info, grupate pe utilități. Un conținut excelent în privința ajutorului vă este adus de Konqueror Handbook, de scurtătura Startup din meniu sau prin introducerea comenzii **konqueror** într-un terminal, urmată de apăsarea tastei **Enter**; vezi imaginea următoare.

Figura 2-2. Konqueror ca navigator pentru meniul de ajutor



Navigatorul pentru meniul de ajutor din Gnome este la fel de prietenos. Îl puteți lansa prin selectarea Applications→Help din meniul Gnome, prin apăsarea iconiței care prezintă un colac de salvare sau prin introducerea într-un terminal a comenzii **gnome-help**. Documentația sistemului și paginile man sunt ușor de accesat într-o interfață prin care putem naviga în căutarea ajutorului.

Managerul de fișiere **nautilus** are un index din care putem căuta paginile man și Info, care sunt legate prin referințe și prin care se poate naviga ușor. Nautilus se pornește din linia de comandă, prin apăsarea iconiței care arată directorul tău personal (sau dosarul personal - home) ori din meniul Gnome.

Marele avantaj al interfețelor grafice (GUI's) legat de documentația sistemului este acela că prezintă toate informațiile într-un mod în care acestea sunt conectate prin legături, astfel încât puteți folosi secțiunea „SEE ALSO” (VEZI ȘI) pentru a naviga fără întrerupere, cu orele chiar, în căutarea cunoștințelor de care aveți nevoie.

2.3.3.5. Excepții

Câteva comenzi nu au documentație separată, pentru că ele sunt parte ale altor comenzi. **cd**, **exit**, **logout** și **pwd** sunt astfel de excepții. Ele sunt părți ale consolii, programul dumneavoastră shell și sunt numite comenzi *proprietăți consolii (shell built-in commands)*. Pentru mai multe informații despre ele, căutați prin paginile man sau Info ale consolii pe care o folosiți. Cei mai mulți utilizatori începători de Linux folosesc o consolă Bash. Consultați [Sectiunea 3.2.3.2](#) pentru mai multe detalii despre console.

Dacă ați schimbat configurația originală a sistemului, este posibil ca paginile man să fie încă în sistem, dar să nu le puteți accesa, pentru că mediu în care rulează consola nu mai este același. În acest caz, trebuie să țineți cont de variabila `MANPATH`. Detalii despre acest subiect sunt prezentate în [Sectiunea 7.2.1.2](#).

Anumite programe sau pachete au doar un set de instrucțiuni sau referințe în directorul `/usr/share/doc`. Consultați [Sectiunea 3.3.4](#) pentru detalii.

În cel mai rău caz, există posibilitatea ca documentația să fie stearsă accidental (sperăm că accidental, deoarece ștergerea acesteia pentru că aşa ați dorit este o idee foarte proastă). În acest caz, verificați dacă prin folosirea uneltelor de căutare vi se returnează ceva legat de subiectul în cauză ([Sectiunea 3.3.3](#)). Dacă nu, va trebui să reinstalați pachetul care conține comanda asupra căreia ați consultat documentația, după cum se arată în [Sectiunea 7.5](#).

2.4. Sumar

Sistemul de operare Linux lucrează tradițional în modul text sau în modul grafic. Deoarece costul procesoarelor și al RAM-ului nu mai este exorbitant în zilele noastre, orice utilizator de Linux poate lucra în modul grafic și va face o regulă din acest lucru. Asta nu înseamnă însă că nu trebuie să știți nimic despre modul text: vom lucra în mediul liniei de comandă pe parcursul acestui curs folosind o fereastră terminal.

Linux-ul încurajează utilizatorii să acumuleze cunoștințe și să devină independenți. Inevitabil, va trebui să citiți multă documentație pentru a reuși; de aceea, după cum veți vedea, vă vom îndruma către alte surse de documentare pentru aproape orice comandă, unealtă sau problemă prezentate în această carte. Cu cât citiți mai multă documentație, cu atât mai repede vă veți obișnui cu manualele și veți înțelege mai ușor conținutul acestora. Faceți-vă cât mai curând un obicei din citirea documentației. Atunci când nu știți răspunsul la o problemă, căutarea prin documentație ar trebui să fie prima dumneavoastră opțiune.

Am învățat câteva comenzi:

Tabelul 2-3. Comenzi noi în capitolul 2: Să începem

Comanda	Înțelesul comenzi
apropos	Caută informații despre o comandă sau un subiect
cat	Arată conținutul unui fișier sau a mai multor fișiere
cd	Schimbă directorul
exit	Terminarea unei sesiuni a consolii
file	Afișează informații despre conținutul unui fișier
info	Citiți paginile Info ale unei comenzi
logout	Terminarea unei sesiuni a consolii
ls	Arată conținutul unui director
man	Citiți paginile de manual ale unei comenzi
passwd	Schimbă parola dumneavoastră
pwd	Afișează directorul în care vă aflați

2.5. Exerciții

Multe lucruri din cele pe care le învățăm sunt despre cum se pot face greșeli și cum putem să evităm să perseverăm în a le face. Exercițiile următoare sunt importante pentru că vă pune în situația de a citi unele mesaje de eroare. Ordinea în care le faceti este foarte importantă.

Nu uitați să folosiți facilitățile oferite de Bash în ceea ce privește linia de comandă: încercați să faceți exercițiile tastând cât mai puține caractere posibile!

2.5.1. Autentificarea și deautentificarea

- Determinați modul în care lucrați, text sau grafic.
Eu lucrez în modul text/grafic. (Tăiați ce este incorect)
- Autentificați-vă cu numele de utilizator și parola pe care le-ați stabilit pe parcursul instalării.
- Deautentificați-vă.
- Autentificați-vă din nou cu un nume de utilizator inexistent.
→ Ce se întâmplă?

2.5.2. Parole

Autentificați-vă din nou cu numele dumneavoastră de utilizator și cu parola.

- Schimbați-vă parola în `P6p3.aa!` și apăsați tasta **Enter**.
→ Ce se întâmplă?
- Încercați din nou cu o parolă ridicol de simplă, precum `123` sau `aaa`.
→ Ce se întâmplă?
- Încercați comanda `psswd` în locul `passwd`.
→ Ce se întâmplă?



Parolă nouă

Dacă nu ați revenit la parola dumneavoastră pe care ați avut-o înaintea exercițiului de mai sus, această parolă va fi „P6p3.aa!”. Schimbați-vă parola la loc după efectuarea exercițiului!

Anumite sisteme pot să nu permită reciclarea parolelor, adică schimbarea în cea originală, în cazul dumneavoastră, decât după un anumit timp sau după un anumit număr de cicluri de schimbare a parolelor, ori chiar pot lua în calcul ambele variabile.

2.5.3. Directoare

Iată câteva exerciții care vă ajută să intrați în subiect:

- Introduceți comanda `cd blah`.
→ Ce se întâmplă?
- Introduceți comanda `cd ..`
Luati aminte la spațiul dintre „cd” și „..”! Folosiți comanda `pwd`.

- Ce se întâmplă?
 - Listați conținutul directorului cu ajutorul comenzi **ls**.
 - Ce vedeți?
 - Ce credeți că sunt acestea?
 - Verificați folosind comanda **pwd**.
 - Introduceți comanda **cd**.
 - Ce se întâmplă?
 - Repetați pasul 2 de două ori.
 - Ce se întâmplă?
 - Afisați conținutul acestui director.
 - Încercați comenziile **cd root** (pentru ubuntu **cd /; ls; ls -a; cd .config** în această ordine^{n.t.}).
 - Ce se întâmplă?
 - La care directoare aveți acces?
 - Repetați pasul 4 (**cd**).

Știți o altă posibilitate prin care să ajungeți în acest loc?
-

2.5.4. Fișierele

- Schimbați directorul pe / și apoi pe etc. Tastați **ls**; dacă răspunsul calculatorului este mai mare decât fereastra, faceți-o mai lungă sau încercați **Shift+PageUp** și **Shift+PageDown**.

Fișierul **inittab** conține răspunsul la întrebarea următoare. Încercați comanda **file** pe acesta.



Fișierul **inittab** nu se regăsește în Ubuntu

Nu puteți aplica acest exercițiu pentru această distribuție. Puteți încerca cu **/etc/event.d/rc-default** dar nu este vorba despre același fișier, după cum se vede, iar ceea ce se returnează la comanda **cat /etc/event.d/rc-default** este dificil de înțeles^{n.t.}.

- Tipul fișierului **inittab** este
- Folosiți comanda **cat inittab** și citiți răspunsul.
 - Care este modul de lucru în care pornește calculatorul dumneavoastră?
- Mergeți la directorul personal (home) cu comanda **cd**.
- Introduceți comanda **file** .
 - Vă ajută să aflați înțelesul lui „.”?
- Puteți să aflați ceva prin folosirea comenzi **cat** asupra lui „.”?
- Afisați ajutorul în cazul comenzi **cat** prin utilizarea opțiunii **-help**. Folosiți opțiunea de

numărare a liniilor returnate pentru a număra câți utilizatori sunt listăți în fișierul /etc/passwd.

2.5.5. Căutarea de ajutor

- Citiți **man intro**
 - Citiți **man ls**
 - Citiți **info passwd**
 - Introduceți comanda **apropos pwd**
 - Încercați comanda **man** sau **info** asupra **cd**
→ Ce mai puteți găsi în plus despre **cd**?
 - Citiți și încercați ceea ce vă este returnat de comanda **ls - -help**
-

Capitolul 3. Despre fișiere și sistemul de fișiere

După explorarea initială din [Capitolul 2](#), suntem gata să discutăm mai în detaliu despre fișierele și directoarele dintr-un sistem Linux. Multii utilizatori au dificultăți în utilizarea Linux-ului, deoarece nu au o vedere de ansamblu asupra a ce fel de date sunt păstrate în anumite locuri. Vom încerca să explicăm organizarea fișierelor în sistemul de fișiere.

Vom arăta, de asemenea, cele mai importante fișiere și directoare, vom folosi metode diferite prin care putem privi conținutul acestora și vom învăța cum pot fi create, mutate sau șterse fișierele și directoarele.

După terminarea exercițiilor din acest capitol, veți putea:

- ◆ Să descrieți structura de fișiere a unui sistem Linux
- ◆ Să citiți și să scrieți calea către un fișier
- ◆ Să descrieți cele mai importante fișiere, inclusiv kernelul și shell-ul
- ◆ Să găsiți fișierele ascunse sau pierdute
- ◆ Să creați, să mutați și să ștergeți fișiere și directoare
- ◆ Să afișați conținutul unui fișier
- ◆ Să înțelegeți și să folosiți mai multe tipuri de legături
- ◆ Să afișați proprietățile fișierelor și să schimbați permisiunile acestora

3.1. O privire asupra sistemului de fișiere din Linux

3.1.1. Fișierele

3.1.1.1. Generalități

O descriere simplă a unui sistem UNIX, care se aplică și Linux-ului, este următoarea:

„Într-un sistem UNIX, orice este un fișier; dacă ceva nu este un fișier, atunci este un proces.”

Această propoziție este adevărată deoarece există fișiere speciale care sunt mai mult decât niște simple fișiere (cele numite pipes – conexiuni și sockets, de exemplu), dar pentru simplicitate, a spune că orice este un fișier constituie o generalizare acceptată. Un sistem Linux, la fel ca UNIX-ul, nu face nici o diferență între un director și un fișier, deoarece un director este doar un fișier care conține numele altor fișiere. Programele, serviciile, texte, imaginile și aşa mai departe, sunt toate fișiere. Dispozitivele, în general, sunt considerate și ele fișiere, din punctul de vedere al sistemului.

Pentru a ne descurca cu toate aceste fișiere într-o manieră ordonată, suntem obișnuiți să le vedem organizate pe discul fix într-o structură arborescentă, pe care o cunoaștem din MS-DOS (Disk Operating System), de exemplu. Din trunchi pleacă ramurile groase care au alte ramuri, iar ramurile din vârf se termină cu frunzele copacului, care sunt fișiere obișnuite. Pentru început vom folosi această imagine a unui copac, dar vom afla mai târziu că nu este o imagine care corespunde întocmai cu realitatea.

3.1.1.2. Felurile fișierelor

Cele mai multe fișiere sunt fișiere simple, numite fișiere *obișnuite*; ele conțin date normale, de exemplu fișierele text, fișierele executabile sau programele, comenzi și răspunsurile unui program, și altele.

În mod normal este rezonabil să presupunem că tot ceea ce întâlnim într-un sistem Linux este un fișier, cu câteva excepții.

- *Directoare*: fișiere care sunt o listă a altor fișiere.
- *Fișiere speciale*: mecanismele folosite pentru introducerea comenzi și returnarea răspunsurilor. Cele mai multe fișiere speciale se găsesc în /dev, le vom discuta mai târziu.
- *Legături*: un sistem pentru a face un fișier sau un director vizibil în mai multe părți ale sistemului de fișiere. Vom discuta în detaliu despre aceste legături.

- *Socket-uri (de domenii)*: un fel special de fișiere, asemănătoare cu socket-urile TCP/IP, care asigură desfășurarea proceselor în rețea, protejate prin controlul accesului la sistemul de fișiere.
- *Numite conexiuni (pipe)*: se comportă mai mult sau mai puțin ca socket-urile și formează o cale pentru comunicarea între procese, fără a utiliza semantica socket-urilor de rețea.

Opțiunea **-l** a comenzi **ls** vă afișează tipul fișierului, utilizând primul caracter al fiecărei linii:

```
jaime:~/Documents> ls -l
total 80
-rw-rw-r-- 1 jaime jaime 31744 Feb 21 17:56 intro Linux.doc
-rw-rw-r-- 1 jaime jaime 41472 Feb 21 17:56 Linux.doc
drwxrwxr-x 2 jaime jaime 4096 Feb 25 11:50 course
```

Tabelul 3-1. Felul fișierelor în listarea amănunțită

Simbol	Înțelesul simbolului
-	Fișier obișnuit
d	Director
l	Legătură
c	Fișier special
s	Socket
p	Numite conexiuni (pipe)
b	Dispozitiv

Pentru a nu afișa mereu listarea amănunțită doar pentru a vedea felul fișierelor, mai multe sisteme, în mod normal, utilizează și comanda **ls -F**, care pune înaintea fiecărui nume de fișier unul din caracterele „/ =*!@” pentru a indica felul fișierului.

Pentru a face totul mai accesibil utilizatorului începător, ambele opțiuni, **-F** și **--color** sunt combinate, precum se arată în [Secțiunea 3.3.1.1](#). Noi vom folosi **ls -F** pe parcursul acestui document, pentru o afișare a tipului de fișier mai la obiect.

Pentru Ubuntu, răspunsul la comanda ls ia în considerare din oficiu opțiunea --color, felul fișierelor fiind evidențiat prin culori^{n.t.}

Ca utilizator, veți folosi doar fișiere obișnuite, fișiere executabile, directoare și legături. Tipurile speciale de fișiere sunt găndite pentru a face sistemul să execute ceea ce doriți dumneavoastră și sunt mânuite de administratorii de sistem și de programatori.

Acum, înainte de a ne apela asupra directoarelor și fișierelor importante, trebuie să învățăm mai multe despre partii.

3.1.2. Despre partii

3.1.2.1. De ce partii?

Cei mai mulți oameni au o idee, chiar dacă doar vagă, despre partii, pentru că orice sistem de operare poate face sau șterge partii. Pentru că Linux-ul folosește mai mult de o partitie pe același disc, chiar dacă este instalat prin procedura automată, poate părea ciudat unora, așa că se cere o explicație.

Unul dintre scopurile pentru care sunt făcute mai multe partii este obținerea unei securizări mai puternice a datelor, în eventualitatea unui dezastru. Prin împărțirea discului fix (hard disk-ului) în partii, datele pot fi grupate și separate. Atunci când se întâmplă accidente, numai datele din partită atacată vor fi distruse, în timp ce datele din celelalte partii vor scăpa neatinse.

Acest principiu datează din zilele în care Linux-ul nu avea un sistem de fișiere de tip jurnalier iar căderile de tensiune puteau conduce la un dezastru. Utilizarea în continuare a partitiilor se datorează rațiunilor de securitate și robustețe, pentru că, în acest mod, o breșă într-o parte a sistemului nu va pune în pericol întregul calculator. Aceasta este principalul motiv pentru care se fac partii. Iată un exemplu simplu: un utilizator creează un script, un program sau o aplicație web care începe să umple discul. Dacă

sistemul are o singură partitie, el nu va mai functiona atunci când discul este plin. Dacă utilizatorul stochează acel program pe o altă partitie, atunci numai datele conținute de partitie în cauză vor fi afectate, pe când celelalte partitii, de sistem sau de date, vor funcționa în continuare.

Trebuie arătat faptul că sistemul de fișiere de tip jurnalier vă protejează datele numai în cazurile de cădere de tensiune sau deconectării improprii ale dispozitivelor de stocare. Nu vă protejează împotriva blocurilor defecte sau erorilor logice din sistemul de fișiere. Pentru acestea din urmă, ar trebui să folosiți o soluție RAID (Redundant Array of Inexpensive Disks).

3.1.2.2. Feluri și tipuri de partitii

Sunt două feluri principale de partitii într-un sistem Linux:

- *partitiile de date*: pe ele sunt stocate datele normale ale unui sistem Linux, care includ și *partitia root*, care conține toate datele care permit pornirea și rularea sistemului; și
- *partitia swap*: o expandare a memoriei fizice a calculatorului, memorie în plus pe discul fix.

Cele mai multe sisteme au o partitie root, una sau mai multe partitii de date și una sau mai multe partitii swap. Sistemele mixte pot avea partitii de date proprii altui sistem, precum partitii care folosesc FAT sau VFAT ca sisteme de fișiere pentru datele specifice MS Windows.

Cele mai multe sisteme Linux folosesc **fdisk** în momentul instalării pentru partitiorare. După cum probabil ați observat din exercițiul de la [Capitolul 1](#), acest lucru se întâmplă, de regulă, automat. În anumite ocazii, totuși, s-ar putea să nu fiți aşa de norocoși. În asemenea situații, trebuie să selectați tipul partitiei manual și să faceți tot manual partitiorarea în sine. Partitii standard ale Linux-ului au numărul 82 pentru swap și 83 pentru cele de date, care pot fi jurnaliere (ext3) sau normale (ext2, pe sistemele mai vechi). Utilitarul **fdisk** conține și partea de ajutor, așa că nu trebuie să memorați aceste valori.

În afară de aceste două tipuri, Linux-ul poate lucra cu mai multe tipuri de sisteme de fișiere, precum relativ mai nou Reiser, JFS, NFS, FATxx și alte sisteme de fișiere proprii altor sisteme de operare (proprietare).

O partitie root standard (arătată prin caracterul „/”) are mărimea de 100-500 MO și conține fișierele de configurare ale sistemului, comenziile principale și programele serviciilor, librariile sistemului, un spațiu temporar și directorul personal (home) al utilizatorului cu drepturi administrative. O instalare standard solicită în jur de 250 MO pentru partitie de root.

Spațiul swap (indicat prin *swap*) este accesibil doar sistemului, fiind ascuns în timpul operării. Swap-ul este acel lucru prin care ne asigurăm că, precum în sistemele UNIX, putem continua să lucrăm indiferent de ceea ce se întâmplă. Pe sistemele Linux, putem spune că, practic, nu veți fi deranjat niciodată de mesaje iritante precum *Memorie depășită, vă rugăm să închideți unele aplicații mai întâi și încercați din nou*, deoarece această memorie în plus este mereu la dispozitie. Memoria swap sau virtuală a fost implementată, în zilele noastre, și de alte sisteme de operare care nu au nimic de-a face cu UNIX-ul.

Folosirea memoriei de pe discul fix este un proces mai lent decât folosirea cipurilor de memorie propriu-zise, dar disponibilitatea memoriei swap asigură o fiabilitate mărită. Vom discuta mai multe despre swap odată cu prezentarea proceselor, în Capitolul 4.

Linux-ul se bazează, în general, pe alocarea unui spațiu pe discul fix destinat memoriei swap de o mărime egală cu dublul memoriei fizice. Atunci când instalați un sistem, trebuie să știți cum să faceți acest lucru. Iată un exemplu pentru un sistem cu o memorie fizică de 512 MO:

- prima posibilitate: o partitie swap de 1 GO
- a doua posibilitate: două partitii swap de 512 MO
- a treia posibilitate: pe două discuri fixe: o partitie swap de 512 MO pe fiecare disc fix.

Ultima opțiune vă garantează cel mai bun rezultat în cazul în care solicitați intensiv sistemul.

Citiiți documentația programelor pentru aspectele specifice. Anumite aplicații, cum ar fi bazele de date, pot să vă solicite mai mult spațiu pentru partitie swap. Alte sisteme pot să nu aibă deloc acest swap, din cauză că nu au, prin construcție, discuri fixe. Spațiul alocat memoriei swap poate depinde și de versiunea de kernel.

Kernelul se află pe o partitie separată în multe distribuții, deoarece este cel mai important fișier al

sistemului. Dacă este și cazul distribuției alese de dumneavoastră, veți găsi o partitură /boot, care conține kernelul și fișierele de date care îl acompaniază.

Restul discului fix este împărțit în partitii de date, deși este posibil ca toate informațiile care nu sunt deosebit de importante pentru funcționarea sistemului, să se găsească pe o singură partitură, cum ar fi în cazul unei instalări standard pentru stații de lucru. Atunci când astfel de informații sunt împărțite pe mai multe partitii, puteți găsi următorul tipar:

- o partitură pentru programele utilizatorului (/usr)
- o partitură folosită pentru datele care aparțin utilizatorului (/home)
- o partitură pentru stocarea datelor temporare, precum fișierele de tipărit sau mail-urile (/var)
- o partitură pentru programele dezvoltate de terțe părți sau alte surse (/opt)

Odată ce ați făcut partitiiile, mai puteți doar adăuga altele. Schimbarea capacitatii sau tipului partitiiilor existente este posibilă dar nu vă sfătuim să faceți acest lucru.

Împărțirea discului fix în partitii este făcută de administratorul de sistem. Pe sistemele mai mari, el sau ea va expanda o partitură pe mai multe discuri fixe, folosind programe adecvate. Cele mai multe distribuții permit setări prestabilite, optimizate pentru stații de lucru (utilizatori de nivel mediu) sau servere cu scop general, acceptând, în același timp, și partitonarea manuală. Pe parcursul procesului de instalare puteți defini propriul tipar de partitonare, folosind fie unealta specifică distribuției alese, care este prezentată în interfață grafică, fie **fdisk**, o unealtă bazată pe modul text, cu care puteți face partitii și alege tipul acestora.

O instalare ca stație de lucru standard este destinată folosirii, în principal, de către una și aceeași persoană. Programele selectate pentru instalare reflectă acest lucru și greul apăsă asupra pachetelor utilizate în mod curent, precum teme atrăgătoare pentru spațiul de lucru, unelte pentru dezvoltare, programe pentru clientul de poștă electronică, programe multimedia, internet și alte servicii. Totul este pus laolaltă pe o partitură mare, spațiul alocat pentru swap este setat la o mărime de două ori mai mare decât memoria fizică și iată că stația ta de lucru este gata, asigurând cel mai mare spațiu disponibil pe discul fix pentru folosință personală, dar cu dezavantajul posibilelor pierderi de date în cazul situațiilor care ridică probleme.

Pe servere, datele sistemului sunt ținute separat de datele utilizatorilor. Programele care oferă servicii sunt ținute separat de datele necesare acestor servicii. Partitonarea este făcută diferit pe astfel de sisteme:

- o partitură care conține toate datele necesare pentru pornirea sistemului
- o partitură care conține fișierele de configurare și programele serverului
- una sau mai multe partitii care conțin date stocate în server, precum tabele din bazele de date, mail-urile utilizatorilor, arhiva ftp, etc
- o partitură care conține programele și aplicațiile utilizatorului
- una sau mai multe partitii care conțin fișiere specifice utilizatorilor (directoarele personale)
- una sau mai multe partitii swap (memoria virtuală)

Serverele au mai multă memorie fizică și, deci, mai mult spațiu alocat pentru swap. Anumite procese asigurate de server, cum ar fi bazele de date, pot solicita mai mult swap decât de obicei; citiți documentația specifică pentru informații detaliate. Pentru a mări performanțele, swap-ul este divizat în mai multe partitii.

3.1.2.3. Punctele de montare

Toate partitiiile sunt atașate la sistem prin punctele de montare. Punctul de montare definește locul unui set anume de date în sistemul de fișiere. De regulă, partitiiile sunt conectate prin partitură *root*. În această partitură, care este indicată prin caracterul (/), sunt create niște directoare. Aceste directoare goale vor fi punctul de start al partitiiilor care vor fi atașate la ele. De exemplu: se dă o partitură care are următoarele directoare:

videos/ cd-images/ pictures/

Dorim să atașăm această partitură în sistemul de fișiere într-un director numit /opt/media. Pentru a face acest lucru, administratorul de sistem trebuie să se asigure că directorul /opt/media există în sistem. Este de preferat ca acest director să fie și gol. Despre cum se face acest lucru vom vorbi tot în acest

capitol. Apoi, utilizând comanda **mount**, administratorul poate ataşa partitia la sistem. Dacă verificați conținutul directorului /opt/media, care era gol, veți constata că el conține acum fișierele și directoarele care există pe mediul montat (disc fix sau partitia a discului fix, CD, DVD, memorie flash, USB sau alte dispozitive de stocare).

Puteți obține informații despre partitii și punctele lor de montare prin comanda **df** (care vine de la *disk full - disc plin* sau *disk free - disc liber*). În Linux, **df** este versiunea GNU și suportă opțiunea **-h** sau *human readable – inteligibilă*, care oferă date mai prietenoase cu utilizatorul. Rețineți că mașinile UNIX au propria versiune a comenzii **df** și propriile versiuni ale altor comenzi. Comportamentul acestora este același, în mod obișnuit, dar versiunile GNU ale uneltelelor de bază au mai multe facilități și, în general, sunt mai bune.

Comanda **df** afișează informații doar despre partitiile active și care nu sunt de tipul swap. Ele pot include partitiile ale altor sisteme din rețea, precum în exemplul de mai jos, în care directoarele personale sunt montate dintr-un server de fișiere accesat prin rețea, o situație des întâlnită în mediile corporatiste.

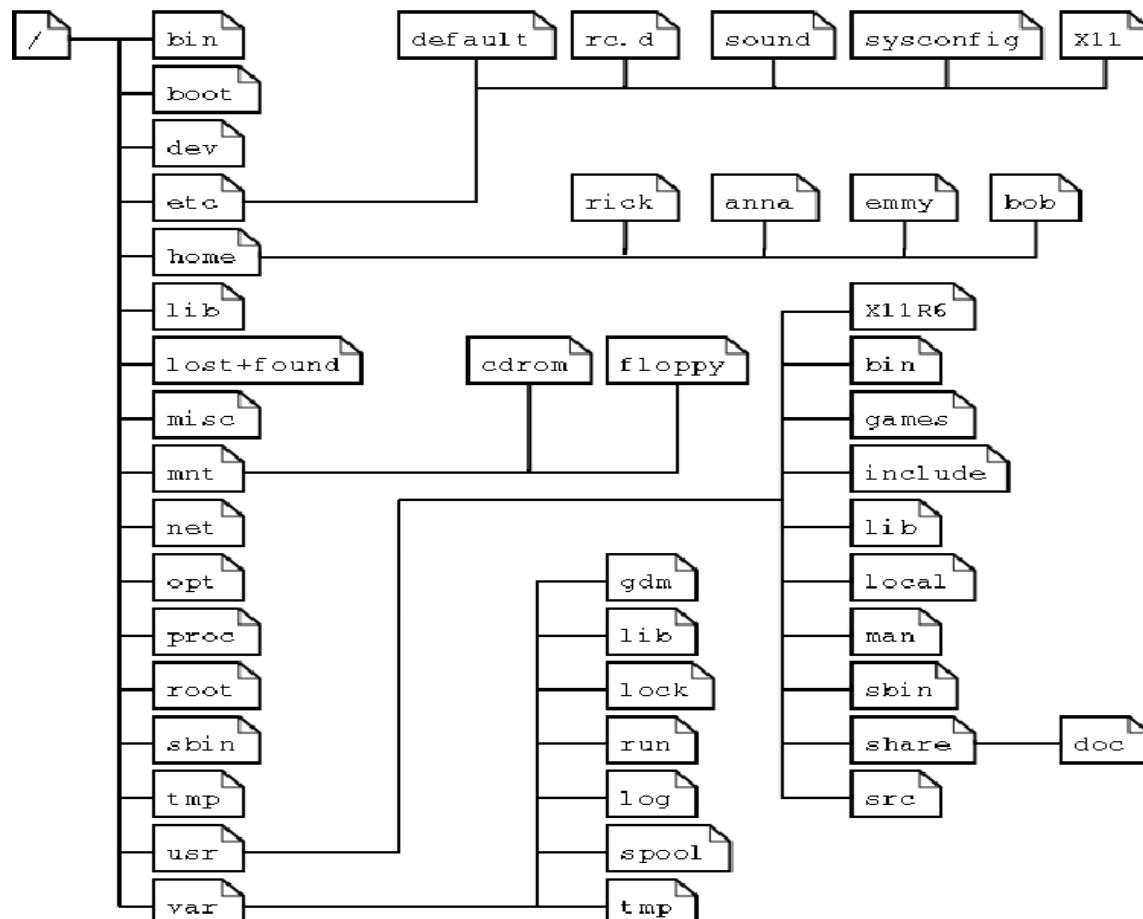
Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/hda8	496M	183M	288M	39%	/
/dev/hda1	124M	8.4M	109M	8%	/boot
/dev/hda5	19G	15G	2.7G	85%	/opt
/dev/hda6	7.0G	5.4G	1.2G	81%	/usr
/dev/hda7	3.7G	2.7G	867M	77%	/var
fs1:/home	8.9G	3.7G	4.7G	44%	/.automount/fs1/root/home

3.1.3. Mai multe despre tiparul sistemului de fișiere

3.1.3.1. Reprezentare

Fișierele din Linux sunt prezentate, prin convenție, ca o structură arborescentă. Pe un sistem Linux standard veți găsi că tiparul sistemului de fișiere se încadrează în schema prezentată mai jos.

Figura 3-1. Tiparul sistemului de fișiere din Linux



Acesta este tiparul unui sistem RedHat. În funcție de administratorul sistemului, de distribuție și de destinația calculatorului în cauză, structura poate varia pentru că directoarele pot fi adăugate sau scoase după dorință. Nici numele directoarelor prezentate mai sus nu sunt neapărat necesare în această formă; ele sunt doar o convenție.

Structura arborescentă a sistemului de fișiere începe cu trunchiul, indicat de caracterul (/). Acest director, care conține toate directoarele și fișierele sistemului, este numit *directorul root* sau „rădăcina” sistemului de fișiere.

Directoarele care se află pe palierul imediat următor directorului rădăcină (root) sunt adesea precedate de caracterul (/), pentru a indica poziția acestora și pentru a evita confuzia cu alte directoare care pot avea același nume. Atunci când începeți să lucrați pe un sistem nou, este o idee bună să explorați directorul rădăcină al acestuia. Iată ce puteți obține:

```
emmy:~> cd /
emmy:/> ls
bin/ dev/ home/ lib/ misc/ opt/ root/ tmp/ var/
boot/ etc/ initrd/ lost+found/ mnt/ proc/ sbin/ usr/
```

Tabelul 3-2. Subdirectoarele directorului rădăcină (root)

Director	Conținut
/bin	Programe obișnuite, folosite de sistem, de administratorul sistemului și de utilizatori.
/boot	Fișierele necesare pornirii sistemului și kernelul, <code>vmlinuz</code> . În unele distribuții recente, datele grub. Grub vine de la GRand Unified Boot Loader și este o încercare de a scăpa de diversele încărcătoare de sistem care există azi.
/dev	Conține referințe la toate echipamentele periferice, care sunt reprezentate ca fișiere cu proprietăți speciale.
/etc	Cele mai importante fișiere de configurare ale sistemului sunt în directorul <code>/etc</code> , acest director conținând date asemănătoare cu cele din Panoul de control al Windows-ului.
/home	Directoarele personale ale utilizatorilor obișnuiți.
/initrd	(în anumite distribuții) Informații pentru pornirea sistemului. Nu îl stergeti!
/lib	Fișierele librăriilor, care includ fișierele programelor folosite de sistem sau de utilizatori.
/lost+found	Fiecare partitie are un <code>lost+found</code> (pierdute și găsite) în directorul principal. Fișierele salvate în urma avariilor sunt în acest director.
/misc	Pentru scopuri diverse
/mnt	Punctul standard de montare pentru sisteme de fișiere externe, cum ar fi CD-ROM sau camere digitale.
/net	Punctul standard de montare pentru sistemele de fișiere din rețea
/opt	Conține în mod normal programe de la terțe părți
/proc	Un sistem de fișiere virtual care conține informații despre resursele sistemului. Mai multe informații despre înțelesul fișierelor din <code>proc</code> sunt obținute prin comanda <code>man proc</code> introdusă într-o fereastră terminal. Fișierul <code>proc.txt</code> discută sistemul de fișiere virtual în detaliu.
/root	Directorul personal al administratorului de sistem. Remarcă diferența dintre <code>/</code> , directorul rădăcină, și <code>/root</code> , directorul personal (home) al utilizatorului <code>root</code> , cel cu privilegii administrative.
/sbin	Programele folosite de sistem și de administratorul sistemului.
/tmp	Spațiu pentru folosință temporară a sistemului, care este șters la repornire, așa că nu îl folosiți pentru a salva ceva!
/usr	Programe, librării, documentație și.a. pentru toate programele folosite de utilizatori.
/var	Loc de stocare pentru toate fișierele variabile și temporare create de utilizatori, precum fișiere jurnal, poștă electronică în aşteptare, fișiere care așteaptă să fie tipărite, fișiere descărcate de pe Internet, sau fișiere care păstrează imaginea unui CD înainte de scrierea acestuia.

Cum puteți afla cărei partitii îi aparține un director? Folosind comanda `df` cu un punct (.) drept

optiune, se afisează partitia căreia aparține directorul în care ne aflăm și informații despre spațiul utilizat pe această partitie:

```
sandra:/lib> df -h .
Filesystem           Size  Used Avail Use% Mounted on
/dev/hda7            980M  163M  767M  18% /
```

Ca regulă, fiecare director care se află imediat sub directorul rădăcină se află pe partitia root, cu o singură excepție: să nu fie afișat ca intrare separată în listarea completă returnată la comanda **df** (sau **df -h** fără alte opțiuni).

Citiți mai multe despre acest subiect în **man hier**.

3.1.3.2. Sistemul de fișiere aşa cum este în realitate

Pentru cei mai mulți utilizatori și în cazul sarcinilor obișnuite de administrare a sistemului, este suficient să se accepte că fișierele și directoarele sunt ordonate într-o structură arborescentă. Calculatorul însă nu înțelege nimic din noțiunea de copac sau structură arborescentă.

Fiecare partitie are un sistem de fișiere propriu. Dacă ne imaginăm toate sisteme de fișiere împreună, ne putem forma o idee despre structura arborescentă a întregului sistem, dar nu este chiar aşa de simplu. Într-un sistem de fișiere, un fișier este reprezentat de un punct sau nod (inode), un fel de serie care conține informații despre datele care individualizează acel fișier: cui aparține și unde se află stocat pe discul fix.

Fiecare partitie are seturi proprii de noduri; pe un sistem cu mai multe partitii pot exista fișiere cu numere de serie identice.

Fiecare nod descrie structura datelor aflate pe discul fix, păstrând informații despre proprietățile fișierului, inclusiv locul fizic de pe discul fix în care se află un anume fișier. Atunci când un disc fix este inițializat ca mediu de stocare, în mod obișnuit pe parcursul procesului de instalare a sistemului sau atunci când sunt adăugate discuri noi unui sistem, este creată multimea acestor noduri, o multime care are un număr fix, limitat, de elemente. Elementele acestei multimi corespund numărului maxim de fișiere, de toate tipurile (se regăsesc aici directoarele, fișierele speciale, legăturile, etc.) care pot exista pe aceeași partitie. Putem socoti că ne bazăm pe existența unui nod la fiecare 2 până la 8 ko de memorie ai mediului de stocare.

Atunci când facem un fișier nou, el primește un nod liber. În acest nod sunt următoarele informații:

- Deținătorul și grupul căruia acesta aparține.
- Tipul fișierului (obișnuit, director,...).
- Permișunile care sunt atașate fișierului, consultați [Sectiunea 3.4.1](#).
- Data și ora la care fișierul a fost creat, deschis și editat.
- Data și ora la care aceste informații conținute de nod au fost schimbate.
- Numărul de legături către fișierul respectiv (subiect detaliat pe parcursul acestui capitol).
- Mărimea fișierului.
- O adresă care arată locul fizic pe discul fix ocupat de fișier.

Sigurele informații care nu sunt incluse într-un nod sunt cele referitoare la numele fișierului și directorul său. Aceste informații sunt păstrate într-un director special dedicat fișierelor. Prin compararea numerelor fișierelor și a nodurilor, sistemul construiește o structură arborescentă, pe care utilizatorii o pot înțelege. Utilizatorii pot afișa numărul nodurilor prin opțiunea `-i` a comenzi `ls`. Nodurile au alocat un spațiu special pe discul fix.

3.2. Orientarea prin sistemul de fișiere

3.2.1. Trasee (path – adresă, cale)

Atunci când doriți ca sistemul să execute o comandă, nu trebuie să dați adresa ei completă pentru

executarea acelei comenzi. De exemplu, știm despre comanda **ls** că se află în directorul /bin (verificați cu **which -a ls**), totuși nu trebuie să folosim această comandă în formatul /bin/ls pentru a lista conținutul directorului curent.

Variabila de mediu PATH se ocupă de aceste situații. Această variabilă listează acele directoare din sistem în care se găsesc fișiere executabile, pentru ca utilizatorii să nu fie obligați să memoreze adresele comenziilor sau să tasteze foarte multe caractere. Astfel, PATH conține multe directoare care au în numele lor bin, după cum ne arată imaginea de mai jos. Comanda **echo** este folosită pentru afișarea conținutului („\$”) variabilei PATH:

```
rogier:> echo $PATH  
/opt/local/bin:/usr/X11R6/bin:/usr/bin:/usr/sbin:/bin
```

În acest exemplu, directoarele /opt/local/bin, /usr/X11R6/bin, /usr/bin, /usr/sbin și /bin sunt explorate frecvent în căutarea programelor solicitate. De îndată ce este găsită o potrivire, căutarea este oprită, chiar dacă nu toate directoarele din variabila de mediu au fost cercetate. Acest lucru poate să conducă la situații neașteptate. În exemplul următor, utilizatoarea știe că există un program numit **sendsms** prin care poate trimite mesaje SMS și că un alt utilizator al aceluiași sistem îl poate folosi, dar ea nu îl poate accesa. Diferența este datorată configurației variabilei PATH (adică ordinea în care sunt dispuse directoarele conținute de variabilă):

```
[jenny@blob jenny]$ sendsms  
bash: sendsms: command not found  
[jenny@blob jenny]$ echo $PATH  
/bin:/usr/bin:/usr/bin/X11:/usr/X11R6/bin:/home/jenny/bin  
[jenny@blob jenny]$ su - tony  
Password:  
tony:~>which sendsms  
sendsms is /usr/local/bin/sendsms  
  
tony:~>echo $PATH  
/home/tony/bin.Linux:/home/tony/bin:/usr/local/bin:/usr/local/sbin:\  
/usr/X11R6/bin:/usr/bin:/usr/sbin:/bin:/sbin
```

Vă rugăm să remarcăți folosirea comenzi **su** (schimbă utilizatorul), care vă permite să deschideți o consolă în mediul altui utilizator, cu condiția să cunoașteți parola aceluia utilizator.

Caracterul (\) ne arată că linia curentă se continuă cu următoarea, fără a fi separate prin apăsarea tastei **Enter**.

Iată un exemplu în care un utilizator dorește să verifice numărul de linii dintr-un fișier prin comanda **wc** (word count – numără cuvintele), dar trebuie să anuleze comanda prin combinăția de taste **Ctrl+C**, deoarece nu se întâmplă nimic:

```
jumper:~> wc -l test  
(Ctrl-C)  
jumper:~> which wc  
wc is hashed (/home/jumper/bin/wc)  
  
jumper:~> echo $PATH  
/home/jumper/bin:/usr/local/bin:/usr/local/sbin:/usr/X11R6/bin:\  
/usr/bin:/usr/sbin:/bin:/sbin
```

Comanda **which** ne arată că acest utilizator are un director bin în directorul său personal, care conține un program numit tot **wc**. Deoarece programul **wc**, care se află în directorul său personal, este găsit primul la căutarea prin variabila de mediu, acest program este cel executat, iar ceea ce i se spune să execute nu este recunoscut, aşa că trebuie opri. Pentru a rezolva această problemă există mai multe căi (întotdeauna există mai multe căi pentru a rezolva o problemă în Linux): una ar fi redenumirea programului **wc** propriu utilizatorului sau utilizatorul poate să furnizeze adresa completă a comenzi pe care o dorește, care poate fi aflată prin folosirea comenzi **which** cu opțiunea -a.

Dacă un utilizator folosește mai frecvent programe din alte directoare, el poate schimba comportamentul variabilei de mediu, pentru ca aceasta să caute în directorul personal la sfârșit:

```
jumper:~> export PATH=/usr/local/bin:/usr/local/sbin:/usr/X11R6/bin:\n/usr/bin:/usr/sbin:/bin:/sbin:/home/jumper/bin
```



Schimbările nu sunt permanente!

Trebuie să știi că atunci când folosiți comanda **export** într-o consolă, schimbările sunt temporare, ele fiind valide doar pentru sesiunea curentă (până ce vă deautentificăți). Deschiderea unei noi sesiuni, chiar în cazul în care sesiunea curentă este în desfășurare, va păstra comportamentul inițial al variabilei de mediu. În [Secțiunea 7.2](#) vom învăța cum putem face ca aceste schimbări să devină permanente, prin adăugarea liniilor în cauză la fișierele de configurare ale consolei.

3.2.2. Trasee absolute și relative

O adresă, care este traseul pe care trebuie să-l urmați în structura de fișiere pentru a ajunge la un anumit fișier, poate fi descrisă ca pornind de la trunchiul copacului (/ sau directorul rădăcină – root). În acest caz, traseul începe cu caracterul (/) și este denumit traseu absolut, deoarece nu pot fi făcute confuzii: numai un singur fișier din sistem corespunde acestui traseu.

În situația în care traseul nu începe cu caracterul (/), pot exista confuzii, precum aceea dintre ~/bin/wc (fișier care se află în directorul personal al utilizatorului) și bin/wc, care se găsește în directorul /usr, confuzie reflectată de exemplul pe care l-am oferit mai sus. Traseele care nu încep cu / sunt considerate trasee relative.

Pentru traseele relative vom folosi întotdeauna caracterele (.) și (..) pentru directorul curent și directorul superior. Iată câteva exemple:

- Atunci când doriți să compilați din codul-sursă, documentația de instalare vă oferă adesea instrucțiuni care vă spun să rulați comanda **./configure**, care pornește programul *configure* din directorul curent (cel care vine cu noul cod) și nu alt program *configure* care se găsește în altă parte a sistemului.
- În fișierele HTML, traseele relative sunt folosite la realizarea unor pagini care pot fi ușor mutate în alte locuri:

```

```

- Remarcați diferențele din captura de ecran de mai jos:

```
theo:~> ls /mp3\nls: /mp3: No such file or directory\ntheo:~>ls mp3/\noriental/ pop/ sixties/
```

3.2.3. Cele mai importante fișiere și directoare

3.2.3.1. Kernelul

Kernelul este inima sistemului. El asigură comunicațiile dintre dispozitivele sistemului. De asemenea, kernelul pornește și oprește procesele și serviciile exact atunci când este nevoie. O mulțime de alte sarcini importante sunt executate de kernel, atât de multe încât există o listă specială de discuții, destinață dezvoltatorilor kernelului, numai pe acest subiect, prin care se tratează un volum impresionant de informații. Discutarea kernelului în detaliu ne-ar purta prea departe de scopul acestui ghid. Pentru moment, este suficient să știi despre acesta că este cel mai important fișier al sistemului.

3.2.3.2. Consola (shell-ul)

3.2.3.2.1. Ce este o consolă?

Atunci când am căutat o definiție potrivită conceptului de *consolă*, am întâmpinat multe greutăți.

Există tot felul de definiții, care încep cu simpla comparație care spune "consola este precum volanul unei mașini", inclusiv definiția destul de vagă din manualul Bash: „consola este un interpreter de comenzi compatibil cu limbajul sh” și se termină cu expresia, și mai obscură, „consola asigură interacțiunea dintre sistem și utilizatorii acestuia”. Consola este mai mult decât atât.

Cea mai potrivită imagine este aceea potrivit căreia consola este o cale de comunicare cu sistemul, un limbaj. Cei mai mulți utilizatori folosesc alt limbaj, cel al clicului de mouse, dezvoltat pentru spațiul de lucru. Dar în acest limbaj calculatorul conduce conversația, utilizatorul având doar un rol pasiv, în care poate alege doar dintre sarcinile care îi sunt oferite. Este greu pentru un programator să includă toate opțiunile și toate comenziile, cu posibilitățile lor, într-o interfață grafică. De aceea, interfețele grafice (GUI's) sunt mai puțin capabile decât comanda sau comenziile care sunt incluse în suportul (backend-ul) sistemului.

Consola, pe de altă parte, este o cale avansată prin care comunică cu sistemul, deoarece permite o comunicare în ambele sensuri, în care dumneavoastră puteți chiar prelua inițiativa. Ambii parteneri sunt egali, aşa că pot fi testate noi idei. Consola permite utilizatorilor să folosească sistemul într-un mod flexibil. O însușire suplimentară este aceea prin care se poate realiza automatizarea sarcinilor.

3.2.3.2.2. Tipuri de console

După cum oamenii cunosc mai multe limbi și dialecte, calculatorul înțelege și el mai multe tipuri de console:

- **sh** sau consola Bourne: consola originală din sistemele UNIX, folosită încă în aceste sisteme sau în mediile înrudite cu UNIX-ul. Este o consolă de bază, un program mic cu câteva facilități. Atunci când este în modul compatibil POSIX, consola **bash** emulează comportamentul consoli **sh**.
- **bash** sau consola Bourne Again (Bourne Again SHell): este consola standard GNU, intuitivă și flexibilă. Recomandată pentru utilizatorii începători, este, în același timp, o unealtă puternică la îndemâna celor avansați sau profesioniști. În Linux, **bash** este consola standard pentru utilizatorii obișnuiți. Această consolă mai este numită *superconsola* Bourne, deoarece conține mai multe suplimente și module. Deci consola Bourne Again este compatibilă cu consola Bourne: comenziile care merg în **sh** vor funcționa și în **bash**. Inversul nu este valabil întotdeauna. Toate exemplele și exercițiile din această carte folosesc consola **bash**.
- **csh** sau consola C: sintaxa acestei console este asemănătoare cu cea a limbajului de programare C. Este solicitată îndeosebi de programatori.
- **tcsh** sau consola Turbo C: este o dezvoltare a consoli C, mai prietenoasă cu utilizatorii și mai rapidă.
- **ksh** sau consola Korn: este apreciată mai mult de persoanele care au experiență în mediul UNIX. Este o dezvoltare a consoli Bourne; în configurația standard, este un coșmar pentru utilizatorii începători.

Fisierul `/etc/shells` vă arată consolele existente într-un sistem Linux:

```
mia:~> cat /etc/shells
/bin/bash
/bin/sh
/bin/tcsh
/bin/csh
```



Consolă Bourne simulată

Luați aminte că directorul `/bin/sh` conține, de regulă, o legătură către consola Bash, executată în modul de compatibilitate cu consola Bourne, atunci când este adresată în acest fel.

Consola implicită a sistemului dumneavoastră se află în fisierul `/etc/passwd` și este arătată de o linie asemănătoare cu următoarea:

```
mia:L2NOfqdlPrHwE:504:504:Mia Maya:/home/mia:/bin/bash
```

Pentru a schimba o consolă cu alta, introduceți numele consoli pe care o dorîți într-un terminal activ. Sistemul găsește directorul în care apare numele introdus de dumneavoastră prin utilizarea setărilor din `PATH` și, fiind vorba despre un fisier executabil (program), consola curentă îl activează și execută. Va fi afișat un alt prompter, deoarece fiecare consolă se manifestă într-un mod aparte:

```
mia:~> tcsh  
[mia@post21 ~]$
```

3.2.3.2.3. Ce consolă folosesc?

Dacă nu știți ce consolă folosiți, fie verificați linia care aparține contului dumneavoastră din /etc/passwd, fie tasteați:

```
echo $SHELL
```

3.2.3.3. Directorul (dosarul) dumneavoastră personal (home directory)

Directorul personal este destinația implicită după conectarea la sistem. În cele mai multe cazuri, este un subdirector al directorului /home, deși acest lucru poate să difere la unele distribuții. Același director poate să se afle pe discul fix al unui server de fișiere din rețea; în acest caz, directorul personal poate fi găsit la /nethome/numele_dumneavoastră_de_utilizator (/nethome/your_user_name). Altă posibilitate ar fi aceea în care administratorul de sistem optează pentru un tipar mai puțin inteligibil și directorul personal ar putea exista în /disk6/HU/07/jgillard.

Oricare ar fi traseul directorului personal, nu trebuie să vă bateți capul prea mult cu el. Traseul corect al directorului dumneavoastră personal este stocat în variabila de mediu HOME, pentru ca programele care solicită acest traseu să îl poată găsi. Comanda echo va afișa conținutul acestei variabile:

```
orlando:~> echo $HOME  
/nethome/orlando
```

Puteți face absolut orice dorîți în directorul personal. Puteți face cât de multe fișiere dorîți în tot atâtea directoare, ceea ce vă limitează fiind doar posibilitățile fizice ale echipamentului și mărimea partitiei sau, câteodată, administratorul, care poate să aloce o cotă fiecărui utilizator. Limitarea accesului la discurile fixe era o practică obișnuită pe vremea când prețul pe unitatea de stocare era destul de ridicat. În zilele noastre, cotele sunt alocate aproape exclusiv în cazul mediilor de stocare foarte mari. Puteți vedea dacă este alocată o cotă cu ajutorul comenzi quota:

```
pierre@lamaison:/> quota -v  
Disk quotas for user pierre (uid 501): none
```

În cazul în care sunt alocate cote, veți vedea o listă cu partitiiile restricționate și limitările specifice. Depășirea limitelor poate fi tolerată pe o perioadă de grătie cu puține restricții sau fără nici un fel de restricție. Informații detaliate pot fi găsite prin comenzi info quota sau man quota.



Nu găsiți quota?

Dacă sistemul nu poate găsi quota, atunci nu este aplicată nici o limitare în utilizarea sistemului de fișiere.

Directorul personal este indicat de caracterul (~), care reprezintă prescurtarea de la /traseul către directorul personal/numele tău de utilizator (/path_to_home/user_name). și prescurtarea se află stocată în variabila de mediu HOME, aşa că nu trebuie să faceți nimic pentru a o activa.

Iată o aplicație foarte simplă și la obiect: utilizatorul de mai jos se mută din directorul /var/music/albums/arno/2001 în directorul images, care se află în directorul său personal, folosind o comandă elegantă:

```
rom:/var/music/albums/arno/2001> cd ~/images  
rom:~/images> pwd  
/home/rom/images
```

Pe parcursul acestui capitol vom vorbi despre comenzi pe care le folosim în gestionarea fișierelor

și directoarelor pentru a menține directorul personal curat.

3.2.4. Cele mai importante fișiere de configurare

După cum am mai spus, cele mai multe fișiere de configurare se găsesc în directorul `/etc`. Conținutul poate fi afișat prin comanda `cat`, care trimite fișierele text la ieșirea standard (de regulă, monitorul). Sintaxa este simplă:

cat fișier1 fișier2...fișierN

Vom încerca să vă prezintăm în această secțiune o trecere în revistă a celor mai importante fișiere de configurare. Cu siguranță că lista nu e completă. Prin adăugarea pachetelor suplimentare, sunt adăugate și fișierele de configurare pentru aceste pachete în `/etc`. Când veți citi fișierele de configurare, veți observa că ele sunt bine comentate și foarte bine explicate. Anumite fișiere au chiar pagini de manual, care conțin documentație în plus, precum `man group`.

Tabelul 3-3. Cele mai întâlnite fișiere de configurare

Fisier	Informații/servicii
aliases	Fișier pentru aliasurile de poștă electronică utilizată cu serverele Sendmail și Postfix. Existența unui server de mail pe fiecare sistem este o practică comună în lumea UNIX-ului, iar aproape fiecare distribuție de Linux conține un pachet Sendmail. În acest fișier, numele utilizatorilor locali sunt comparate cu numele reale, aşa cum apar ele în adresele de e-mail sau cu alte adrese locale.
apache	Fișierele de configurare pentru serverul web Apache.
bashrc	Fișier de configurare extins pentru consola Bourne Again. Definește funcțiile și aliasurile pentru toți utilizatorii. Alte console pot avea propriile fișiere de configurare extinse, precum <code>cshrc</code> .
directoarele crontab și cron.*	Configurările sarcinilor care trebuie executate periodic – executarea copiilor de siguranță, actualizări ale sistemului, curățarea sistemului, actualizarea raportelor, etc.
default	Opțiunile implicate ale anumitor comenzi, precum <code>useradd</code> .
filesystems	Sistemele de fișiere cunoscute: ext3, vfat, iso9660, etc.
fstab	Listarea partitiilor cu punctele lor de montare.
ftp*	Configurările serverului ftp: cine se poate conecta, ce părți ale sistemului sunt accesibile, etc.
group	Fișierul de configurare pentru grupurile de utilizatori. Folosiți utilitarele securizate <code>groupadd</code> , <code>groupmod</code> și <code>groupdel</code> pentru editarea acestor fișiere. Editați manual numai dacă aveți experiență.
hosts	O listă a calculatoarelor care pot fi contactate folosind rețea, fără a avea nevoie de DNS-uri. Nu are nimic de-a face cu configurările de rețea ale sistemului, care se găsesc în <code>/etc/sysconfig</code> .
inittab	Informații pentru pornirea sistemului: mod, numărul consolelor text, etc.
issue	Informații despre distribuție (versiune și/sau informații despre kernel).
ld.so.conf	Localizarea fișierelor biblioteci.
lilo.conf, silo.conf, aboot.conf, etc.	Informații pentru pornirea sistemului necesare încărcătorului de sisteme din Linux, LILO (Linux Loader), dar care este înlocuit gradual cu GRUB.
logrotate.*	Actualizarea (rotirea) fișierelor jurnal, un sistem care previne acumularea în exces a fișierelor tip jurnal.
mail	Director care conține instrucțiuni pentru comportamentul serverului de mail.
modules.conf	Configurări ale modulelor care asigură însușiri speciale (drivere – module de nucleu).
motd	Mesajul zilei: este arătat oricui se conectează la sistem (în modul text), poate fi

	folosit de administratorul de sistem pentru a anunța închiderea sistemului pentru întreținere, etc.
mtab	Sistemele de fișiere montate. Vă sfătuim să nu editați acest fișier niciodată.
nsswitch.conf	Ordinea în care trebuie contactate numele de aplicația de soluționare a echivalenței, când un proces solicită apelarea unei gazde (host name).
pam.d	Configurările modulelor de autentificare.
passwd	Listează utilizatorii locali. Folosiți utilitarele securizate useradd , usermod și userdel pentru a edita acest fișier. Editați manual numai dacă aveți experiență.
printcap	Fișier depășit, dar încă utilizat, de configurare a imprimantei. Nu editați manual decât dacă sunteți un utilizator experimentat.
profile	Fișier de configurare extins al mediului consolei: variabile, proprietăți implicate ale fișierelor noi, limitări ale resurselor, etc.
rc*	Director care definește serviciile active pentru fiecare nivel de executare (runlevel).
resolv.conf	Ordinea în care sunt contactate serverele DNS (Domain Name Servers).
sendmail.cf	Fișierul principal de configurare pentru serviciul Sendmail.
services	Conexiunile acceptate de sistemul respectiv (porturile deschise).
sndconfig sau sound	Configurările plăcii de sunet și ale evenimentelor conexe.
ssh	Directorul care conține fișierele de configurare ale consolei securizate client și server.
sysconfig	Director care conține fișierele de configurare ale sistemului: maus, tastatură, rețea, spațiu de lucru, frecvența sistemului, administrarea consumului, etc (specific distribuției RedHat).
X11	Setările serverului grafic X. RedHat folosește Xfree, ceea ce se reflectă în numele principalului fișier de configurare, Xfree86Config. Conține și indicațiile generale pentru administratorii de ferestre disponibili în sistem, de exemplu gdm , fvwm , twm , etc.
xinetd.* sau inetd.conf	Fișiere de configurare pentru servicii Internet care rulează prin serviciile (extinse) de Internet ale sistemului (servere care nu rulează un serviciu independent).

Pe parcursul acestui ghid vom afla mai multe lucruri despre aceste fișiere și vom studia câteva în detaliu.

3.2.5. Cele mai întâlnite dispozitive

Dispozitivele, în general orice periferice atașate la un calculator în afară de procesorul propriu-zis, sunt văzute de sistem ca intrări în directorul /dev. Unul din avantajele modului în care UNIX-ul lucrează cu dispozitivele este acela că nici utilizatorul și nici sistemul nu trebuie să-și bată prea mult capul cu specificațiile acestor dispozitive.

Cei nou veniți în Linux, sau UNIX în general, sunt adesea copleșiți de volumul de concepte și termeni pe care trebuie să le învețe. De aceea vă prezintăm mai jos o listă cu dispozitivele despre care vom vorbi pe parcursul acestui ghid.

Tabelul 3-4. Dispozitive des întâlnite

Nume	Dispozitiv
cdrom	Unitate CD
console	Intrare specială pentru consola folosită în mod curent
cua*	Porturi seriale
dsp*	Dispozitive pentru eșantionare și înregistrare

fd*	Intrări pentru dispozitivele floppy (dischete), intrarea implicită fiind /dev/fd0, unitatea de dischete de 1,44 MO
hd [a-t] [1-16]	Asistență pentru unitățile IDE, care asigură numărul maxim de partiții
ir*	Dispozitive infraroșu
isdn*	Administrarea legăturilor ISDN
js*	Joystick-uri
lp*	Imprimante
mem	Memorie
midi*	Player midi
mixer* și music	Model ideal al unui mixer (combină sau adaugă semnale audio)
modem	Modem
mouse (sau msmouse, logimouse, psmouse, input/mice, psaux)	Toate felurile de mausuri
null	Sac fără fund de gunoi
par*	Intrări pentru asistența porturilor paralele
pty*	Pseudoterminale
radio*	Pentru dispozitive radioamatori (HAM`s)
ram*	Dispozitive de pornire a sistemului
sd*	Asistență discuri și partiții SCSI
sequencer	Pentru aplicații audio care folosesc facilitățile de sintetizare ale placii audio (controller-e ale dispozitivelor MIDI)
tty*	Console virtuale care simulează terminale vt100
usb*	Dispozitive USB și scannere
video*	De utilizat pentru plăcile grafice care includ video

3.2.6. Cele mai întâlnite fișiere variabile

În directorul /var se găsește un set de directoare pentru stocarea datelor care nu sunt, prin specificul lor, constante (opuse celor din programul ls sau celor cuprinse în fișierele de configurare ale sistemului, care se schimbă foarte rar sau chiar deloc). Toate fișierele care se schimbă frecvent, precum fișierele raport (log files), căsuțele de mail, cozile în așteptare, etc, sunt păstrate în subdirectoare ale directorului /var.

Ca măsură de securitate, aceste fișiere sunt păstrate, de regulă, separat de fișierele principale ale sistemului, pentru a le ține sub supraveghere strictă, cu îmăsprirea permisiunilor acolo unde este cazul. O mare parte a acestor fișiere au nevoie de mai multe permisiuni decât în mod normal, precum /var/tmp, care trebuie să fie scris de oricine. Multe din activitățile utilizatorilor pot fi găsite aici și chiar activități generate de utilizatori necunoscuți, conectați la sistemul dumneavoastră prin Internet. Iată deci un motiv pentru care directorul /var, cu toate subdirectoarele sale, se află, de regulă, pe o partitură separată. În acest mod nu există riscul ca o bombă de mail, de exemplu, să poată umple tot sistemul de fișiere, afectând datele importante, cum ar fi programele și fișierele de configurare.



/var/tmp și /tmp

Fișierele din /tmp pot fi sterse de către sistem fără preaviz, ca parte a sarcinilor obișnuite sau la repornirea calculatorului. Pe unele sisteme (personalizate) și /var/tmp poate să se comporte imprevizibil. Totuși, pentru că nu este un comportament implicit, vă sfătuim să folosiți directorul /var/tmp pentru salvarea fișierelor temporare. Dacă aveți dubii, întrebați administratorul de sistem. Dacă acest administrator sunteți chiar dumneavoastră, puteți fi îndeajuns de sigur că directorul despre care vorbim se comportă

normal dacă nu ați schimbat conștient însușirile /var/tmp (ca root, deoarece un utilizator obișnuit nu o poate face).

Orice ați face, încercați să vă limitați la privilegiile asociate unui utilizator obișnuit – nu salvați fișiere direct în secțiunea de root (/) a sistemului de fișiere, nu le puneți în directorul /usr sau în subdirectoarele acestuia ori în alt loc dedicat. Astfel vă este limitat accesul doar la fișierele sigure, care nu implică modificări majore în sistem.

Unul dintre principalele sisteme de securitate dintr-un sistem UNIX, implementat, de asemenea, pe toate mașinile Linux, este facilitarea de păstrare a fișierelor jurnal, care înregistrează toate acțiunile utilizatorilor, procesele, evenimentele din sistem, etc. Fișierul de configurare al așa-numitului *syslogdaemon* determină unde și pentru cât timp vor fi păstrate informațiile din fișierele jurnal. Locul implicit pentru fișierele jurnal este /var/log, și conține jurnalele de acces, jurnalele serviciilor, mesajele de sistem, și.a..

În directorul /var sunt și datele serverelor, care sunt ținute aici pentru a fi separate de datele sensibile, precum programele în sine și fișierele de configurare. Un exemplu tipic sistemelor Linux este directorul /var/www, care conține paginile HTML, scripturile și imaginile pe care le oferă serverul. Vă sfătuim ca și sistemul de fișiere FTP care se află pe un server FTP (datele care pot fi descărcate de un client aflat la distanță) să fie păstrate într-un subdirector al lui /var. Pentru că aceste date sunt accesibile publicului și schimbate adesea de utilizatori necunoscuți, este mai sigur să le păstrați aici, departe de partiiile și directoarele care conțin date sensibile.

În cazul celor mai multe instalări ca stații de lucru, /var/spool va conține cel puțin un director at și altul cron, în care vom găsi sarcinile programate. Pe stațiile de lucru pe care sunt instalate programele pentru birou, acest director conține și lpd, care păstrează informații despre lucrările care așteaptă tipărire, fișierele de configurare ale imprimantei, cât și fișierele jurnal ale acesteia.

Pe sistemele folosite ca servere vom găsi, în general, directorul /var/spool/mail, care conține mailurile primite de utilizatorii locali, stocate în fișiere separate pentru fiecare utilizator, așa numita „căsuță” a utilizatorului (user’s „inbox”). Un director înrudit este mqueue, coada de așteptare pentru mesajele de poștă electronică netrimise încă. Aceste părți ale sistemului pot fi foarte solicitate în cazul serverelor de mail care au foarte mulți utilizatori. Serverele de știri folosesc și ele directorul /var/spool din cauza cantităților enorme de mesaje pe care le proceseză.

Directorul /var/lib/rpm este specific distribuțiilor bazate pe RPM (Administratorul de pachete din RedHat); este locul în care sunt stocate informațiile despre pachetele RedHat. Alți administratori de pachete păstrează datele specifice tot în acest director /var.

3.3. Manipularea fișierelor

3.3.1. Consultarea proprietăților fișierelor

3.3.1.1. Mai multe lucruri despre comanda ls

În afară de numele unui fișier, **ls** vă poate oferi mult mai multe informații, cum ar fi tipul fișierului, despre care am vorbit deja. Vă poate arăta, de asemenea, permisiunile asociate unui fișier, mărimea fișierului, numărul de nod (inode), data și ora la care a fost făcut, deținători și numărul de legături către acel fișier. Dacă asociem comenzi **ls** opțiunea -a, sunt afișate și fișierele care sunt în mod normal ascunse. Numele acestor fișiere începe cu un punct (caracterul „.”). Fișierele de configurare din directorul dumneavoastră personal sunt exemple tipice. După ce veți lucra o perioadă de timp pe un anume sistem, veți observa că există zeci de fișiere și directoare care nu sunt indexate. În afară de acest lucru, fiecare director conține un fișier al cărui nume conține doar un punct (.) și altul care conține doar două puncte (..), care sunt folosite în combinație cu numărul nodului asociat lor pentru a determina poziția directorului în sistemul arborescent de fișiere.

Trebuie neapărat să citiți paginile **Info** despre **ls**, deoarece este o comandă des utilizată, care are multe opțiuni folosite. Opțiunile pot fi combinate, comportament care este specific multor comenzi și opțiuni ale UNIX-ului. O combinație des întâlnită este **ls -al**; ea arată lista întreagă a fișierelor cu proprietățile lor precum și destinațiile spre care tăntesc eventualele legături simbolice. **ls -latr** afișează aceleași fișiere, doar că în ordine inversă, în aşa fel încât fișierele care au fost modificate recent apar la baza listei. Iată câteva exemple:

```

kriessie:~/mp3> ls
Albums/ Radio/ Singles/ gene/ index.html

kriessie:~/mp3> ls -a
./ .thumbs Radio gene/
../ Albums/ Singles/ index.html

kriessie:~/mp3> ls -l Radio/
total 8
drwxr-xr-x 2 kriessie kriessie 4096 Oct 30 1999 Carolina/
drwxr-xr-x 2 kriessie kriessie 4096 Sep 24 1999 Slashdot/

kriessie:~/mp3> ls -ld Radio/
drwxr-xr-x 4 kriessie kriessie 4096 Oct 30 1999 Radio/

kriessie:~/mp3> ls -ltr
total 20
drwxr-xr-x 4 kriessie kriessie 4096 Oct 30 1999 Radio/
-rw-r--r-- 1 kriessie kriessie 453 Jan 7 2001 index.html
drwxrwxr-x 30 kriessie kriessie 4096 Oct 20 17:32 Singles/
drwxr-xr-x 2 kriessie kriessie 4096 Dec 4 23:22 gene/
drwxrwxr-x 13 kriessie kriessie 4096 Dec 21 11:40 Albums/

```

În cele mai multe distribuții de Linux, comanda **ls** este executată implicit sub forma aliasului color-ls. Această facilitate vă permite vizualizarea tipului fișierelor fără a folosi vreo opțiune asupra **ls**. Pentru a vă arăta aceste informații, fiecare tip de fișier este prezentat printr-o culoare distinctă. Schema standard se află în /etc/DIR_COLORS:

Tabelul 3-5. Schema implicită de culori pentru color-ls

Culoare	Tip de fișier
albastru	directoare
roșu	arhive comprimate
alb	fișiere text
roz	imagini
cyan	legături
galben	dispozitive
verde	executabile
roșu intermitent	legături deteriorate

Mai multe informații veți găsi în paginile man. Aceste informații erau afișate, la începuturi, prin atașarea unui sufix fiecărui fișier de tip special. Pentru utilizarea în medii monocolor (cum ar fi printarea unei liste a directoarelor) și pentru accesibilitate, această schemă este încă în uz:

Tabelul 3-6. Schema implicită pentru sufixele ls

Caracter	Tip de fișier
nimic	fișier obișnuit
/	director
*	fișier executabil
@	legătură
=	socket
	conexiuni (named pipe)

O descriere completă a funcțiilor și facilităților comenzi **ls** poate fi citită prin comanda **info coreutils ls**.

3.3.1.2. Mai multe unelte

Pentru a afla mai multe despre tipul de date cu care lucrați, folosiți comanda **file**. Prin teste care verifică proprietățile fișierelor, numere magice și teste de limbaj, **file** încearcă să afișeze proprietățile unui fișier într-un format mai elaborat. Iată câteva exemple:

```
mike:~> file Documents/
Documents/: directory

mike:~> file high-tech-stats.pdf
high-tech-stats.pdf: PDF document, version 1.2

mike:~> file Nari-288.rm
Nari-288.rm: RealMedia file

mike:~> file bijlage10.sdw
bijlage10.sdw: Microsoft Office Document

mike:~> file logo.xcf
logo.xcf: GIMP XCF image data, version 0, 150 x 38, RGB Color

mike:~> file cv.txt
cv.txt: ISO-8859 text

mike:~> file image.png
image.png: PNG image data, 616 x 862, 8-bit grayscale, non-interlaced

mike:~> file figure
figure: ASCII text

mike:~> file mettux.jpg
mettux.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI),
"28 Jun 1999", 144 x 144

mike:~> file 42.zip.gz
42.zip.gz: gzip compressed data, deflated, original filename,
`42.zip', last modified: Thu Nov 1 23:45:39 2001, os: Unix

mike:~> file vi.gif
vi.gif: GIF image data, version 89a, 88 x 31

mike:~> file slide1
slide1: HTML document text

mike:~> file template.xls
template.xls: Microsoft Office Document

mike:~> file abook.ps
abook.ps: PostScript document text conforming at level 2.0

mike:~> file /dev/log
/dev/log: socket

mike:~> file /dev/hda
/dev/hda: block special (3/0)
```

Comanda **file** poate fi folosită cu mai multe opțiuni; opțiunea **-z** vă permite să vă uitați în fișierele comprimate. Utilizați comanda **info file** pentru o descriere detaliată. Să nu uitați că rezultatul comenzi **file** nu este unul absolut, ci o presupunere. Cu alte cuvinte, **file** poate fi păcălită.



De ce se acordă o atenție atât de mare tipului și proprietăților fișierelor?

Pe scurt, vom discuta câteva unelte proprii liniei de comandă pentru a privi în *fișierele text simplu*. Aceste unelte nu vor funcționa dacă sunt folosite pe tipuri greșite de fișiere. În cel mai rău caz, ele vor determina închiderea consolei și/sau veți auzi multe proteste pe ieșirea audio. Dacă vi se întâmplă acest lucru, închideți sesiunea curentă a terminalului și deschideți una nouă. Însă încercați să evitați producerea

evenimentelor de acest tip, deoarece pot fi considerate supărătoare de către ceilalți.

3.3.2. Crearea și stergerea directoarelor

3.3.2.1. Dezordinea...

...nu este greu de atins. În zilele noastre aproape orice sistem se află într-o rețea, așa că fișierele sunt copiate cu ușurință dintr-un sistem în altul. În special atunci când lucrăți în mediul grafic, crearea de fișiere noi nu pune nici o problemă și se realizează adesea fără a se cere aprobarea utilizatorului. Pentru a vă prezenta problema, vă prezentăm conținutul complet al unui director personal nou creat pe un sistem standard RedHat:

```
[newuser@blob user]$ ls -al
total 32
drwx----- 3 user      user        4096 Jan 16 13:32 .
drwxr-xr-x  6 root      root        4096 Jan 16 13:32 ..
-rw-r--r--  1 user      user        24 Jan 16 13:32 .bash_logout
-rw-r--r--  1 user      user       191 Jan 16 13:32 .bash_profile
-rw-r--r--  1 user      user       124 Jan 16 13:32 .bashrc
drwxr-xr-x  3 user      user        4096 Jan 16 13:32 .kde
-rw-r--r--  1 user      user      3511 Jan 16 13:32 .screenrc
-rw-----  1 user      user       61 Jan 16 13:32 .xauthDqztLr
```

La prima vedere, nici conținutul unui director personal folosit îndelung nu arată prea rău:

```
olduser:~> ls
app-defaults/ crossover/   Fvwm@     mp3/      OpenOffice.org638/
articles/      Desktop/    GNUStep/   Nautilus/ staroffice6.0/
bin/           Desktop1/   images/    nqc/      training/
brol/          desktoptest/ Machines@ ns_imap/  webstart/
C/             Documents/  mail/      nsmail/   xml/
closed/         Emacs@     Mail/      office52/ Xrootenv.0
```

Dar dacă sunt incluse în această listă toate directoarele și fișierele care încep cu un punct, vom avea mult mai multe poziții în această listă, 185 în cazul de față. Acest lucru se întâmplă deoarece cele mai multe aplicații au propriile directoare și/sau fișiere, care conțin opțiunile de configurare ale utilizatorului, în directorul personal al acestuia. Aceste fișiere sunt create, de regulă, atunci când pornim o aplicație pentru prima dată. Câteodată, veți fi notificați atunci când un director va fi creat, dar de cele mai multe ori totul se face automat.

În plus, întotdeauna sunt create noi fișiere, deoarece utilizatorii doresc să salveze date, să păstreze mai multe versiuni a ceea ce au lucrat, să folosească aplicațiile pentru Internet și să descarce fișiere și atașamente în calculatorul propriu. și aceste activități sunt de neoprit. Este de înțeles că avem nevoie de o schemă care să ne ajute să controlăm aceste probleme.

În secțiunea următoare vom discuta mijloacele prin care putem păstra ordinea. Vom vorbi numai despre uneltele text din consolă, deoarece uneltele grafice sunt intuitive și au aceeași prezentare și mod de lucru precum binecunoscuții administratori de fișiere MS Windows, inclusiv funcții de ajutor în modul grafic și alte facilități care vin în mod normal cu astfel de aplicații. Lista de mai jos este o prezentare a celor mai cunoscuți administratori de fișiere din GNU/Linux. Cei mai mulți pot fi porniți din meniul administratorului spațiului de lucru, prin apăsarea iconiței care arată directorul dumneavoastră personal sau din linia de comandă, astfel:

- **nautlius:** este administratorul de fișiere implicit din Gnome, spațiul de lucru GNU. O documentație excelentă pentru această unealtă poate fi găsită la: <http://www.gnome.org>.
- **konqueror:** este administratorul de fișiere utilizat implicit în spațiul de lucru KDE. Manualul poate fi găsit la: <http://docs.kde.org>.
- **mc:** Midnight Commander, administratorul de fișiere din UNIX, în stilul Norton Commander. Documentația este disponibilă la <http://www.gnu.org/directory/> sau la un site-oglindă precum <http://www.ibiblio.org>.

Merită să vă familiarizați cu aceste aplicații, pentru că ele impresionează de obicei pe cei nou veniți în Linux, iar despre faptul că aveți de unde alege, vă putem spune că aplicațiile prezentate mai sus sunt doar

cele mai întrebuințate unelte de administrare a directoarelor și fișierelor, multe alte proiecte aflându-se în dezvoltare. Acum vom pătrunde în esența lor și vom vedea cum aceste unelte grafice folosesc comenzi obișnuite ale UNIX-ului.

3.3.2.2 Uneltele

3.3.2.2.1. Crearea directoarelor

O cale prin care puteți păstra lucrurile sub control este să alocați anumitor fișiere locuri implicate prin crearea de directoare și subdirectoare (sau foldere și subfoldere, dacă preferați să le numiți astfel). Acest lucru se face prin comanda **mkdir**:

```
richard:~> mkdir archive  
richard:~> ls -ld archive  
drwxrwxrwx 2 richard richard 4096 Jan 13 14:09 archive/
```

Crearea directoarelor și subdirectoarelor într-un singur pas se realizează prin opțiunea **-p**:

```
richard:~> cd archive  
richard:~/archive> mkdir 1999 2000 2001  
  
richard:~/archive> ls  
1999/ 2000/ 2001/  
  
richard:~/archive> mkdir 2001/reports/Restaurants-Michelin/  
mkdir: cannot create directory `2001/reports/Restaurants-Michelin/':  
No such file or directory  
  
richard:~/archive> mkdir -p 2001/reports/Restaurants-Michelin/  
  
richard:~/archive> ls 2001/reports/  
Restaurants-Michelin/
```

Dacă noul fișier are nevoie de alte permișii decât cele alocate implicit, noile drepturi de acces pot fi configurate într-un singur pas, tot prin comanda **mkdir**; consultați paginile Info pentru mai multe detalii. Vom discuta modurile de acces în secțiunea referitoare la securitatea fișierelor.

Numele unui director trebuie alocat după aceleași reguli care se aplică fișierelor normale. Una din cele mai importante restricții este că nu puteți avea două fișiere cu același nume într-un director (dar amintiti-vă ca Linux-ul tratează, precum UNIX-ul, majusculele ca fiind semnificative). Teoretic, nu există limite în numărul de caractere care pot compune numele unui director, dar în practică sunt folosite maxim 80 de caractere, pentru ca numele directorului să poată fi afișat pe o linie a terminalului. Puteți folosi orice caracter dorit în alcătuirea numelui unui fișier, dar vă sfătuim să excludeți caracterele care au un înțeles special în consolă. Dacă aveți dubii, verificați [Appendix C](#).

3.3.2.2.2. Mutarea fișierelor

După ce ne-am structurat directorul personal, este timpul să punem fiecare fișier răzleț în locul potrivit cu ajutorul comenzi **mv**:

```
richard:~/archive> mv ./report[1-4].doc reports/Restaurants-Michelin/
```

Această comandă se aplică și în cazul redenumirii fișierelor:

```
richard:~> ls To_Do  
-rw-rw-r-- 1 richard richard 2534 Jan 15 12:39 To_Do  
  
richard:~> mv To_Do done  
  
richard:~> ls -l done  
-rw-rw-r-- 1 richard richard 2534 Jan 15 12:39 done
```

Este clar pentru oricine că, în exemplul de mai sus, doar numele fișierului s-a schimbat. Toate celelalte proprietăți rămân aceleiași.

Informații detaliate despre sintaxa și facilitățile comenzi **mv** pot fi găsite în paginile man sau Info ale acestei comenzi.

Folosirea documentației trebuie să devină un act reflex pentru dumneavoastră în cazul în care vă confruntați cu o problemă. Răspunsurile la problemele pe care le întâlniți sunt conținute, de cele mai multe ori, în documentația sistemului. Chiar și utilizatorii experimentați consultă pagini man în fiecare zi, așa că cei nou veniți în Linux ar trebui să le citească întotdeauna.

După un timp, veți cunoaște cele mai utilizate comenzi și opțiunile lor, dar tot veți avea nevoie de documentație, ca sursă primară de informare. Menționăm că informațiile conținute de CUM SĂ-uri, Întrebări frecvente (FAQ's), paginile man și alte surse, sunt înglobate în ultimul timp, gradual, în paginile Info, care se constituie astfel în cea mai actualizată sursă de documentare.

3.3.2.2.3. Copierea fișierelor

Copierea fișierelor și directoarelor se face prin comanda **cp**. O opțiune folosită a acestei comenzi este copiere recursivă (copierea tuturor fișierelor și subdirectoarelor dintr-un director), atașând **-R** comenzi **cp**. Sintaxa, în cazul general, este:

cp [-R] fișier_de_copiat destinația_nouă (cp [-R] fromfile tofile)

Vă prezentăm drept exemplu cazul unui utilizator *newguy* care dorește să aibă aceleași configurații ale spațiului de lucru Gnome pe care le are utilizatorul *oldguy*. O cale prin care poate fi rezolvată această problemă este copierea configurațiilor care aparțin utilizatorului *oldguy* în directorul personal al utilizatorului *newguy*:

```
victor:~> cp -R .../oldguy/.gnome/ .
```

Vor exista niște erori legate de permisiunile atașate fișierelor, dar toate au de-a face cu fișiere personale, de care utilizatorul *newguy* nu are nevoie. Vom discuta în altă parte despre cum să schimbăm aceste permisiuni în cazul în care devin o problemă.

3.3.2.2.4. Ștergerea fișierelor

Pentru a șterge fișiere folosiți comanda **rm** iar pentru ștergerea directoarelor goale, folosiți comanda **rmdir** (folosiți **ls -a** pentru a verifica dacă un director este gol sau nu). Comanda rm are opțiuni pe care le putem folosi în ștergerea directoarelor care nu sunt goale, adică putem șterge directorul cu toate subdirectoarele sale. Consultați paginile Info pentru lămuriri în privința acestor opțiuni destul de periculoase.



Cât de gol poate fi un director?

Este un lucru normal ca directoarele . (punct) și .. (punct-punct) să fie imposibil de șters, deoarece ele sunt absolut necesare într-un director, chiar gol, pentru a determina rangul acelui director în sistemul de fișiere.

În Linux, la fel ca în UNIX, nu există coș de gunoi – cel puțin, nu în cazul consolei, pentru că în cazul interfeței grafice, există destule alternative. Așa că odată șters, un fișier este pierdut și, în general, nu există o cale prin care să îl puteți recupera, în afară de cazurile în care aveți copii de siguranță ale fișierului respectiv sau sunteți foarte rapid și aveți un administrator de sistem foarte bun. Pentru a proteja utilizatorii începători se poate activa comportamentul interactiv în cazul comenziilor **rm**, **cp** și **mv** prin opțiunea **-i**. În acest mod sistemul nu va reacționa doar la cererea simplă a utilizatorului. În schimb, va cere confirmare, așa că este nevoie de o apăsare a tastei Enter pentru a produce vreo pagubă, dacă este cazul:

```
mary:~> rm -ri archive/
rm: descend into directory `archive'? y
rm: descend into directory `archive/reports'? y
rm: remove directory `archive/reports'? y
rm: descend into directory `archive/backup'? y
rm: remove `archive/backup/sysup200112.tar'? y
rm: remove directory `archive/backup'? y
rm: remove directory `archive'? y
```

Vom discuta în Capitolul 7, în care vom prezenta detalii despre personalizarea consolelor, modul în

care puteți configura ca implicit acest comportament.

3.3.3. Căutarea fișierelor

3.3.3.1. Utilizarea facilităților consolei

Din exemplul în care am tratat mutarea fișierelor ati observat cum consola poate lucra cu mai multe fișiere în același timp. În acel exemplu consola știe ce i se cere prin caracterele cuprinse între parantezele drepte „ [„ și „] ”. Consola poate substitui serii de numere și caractere, majuscule ori nu. Poate substitui oricât de multe caractere doriti cu un asterisc și un singur caracter cu un semn de întrebare.

Toate substituirile pot fi folosite simultan; consola are o logică imbatabilă în această privință. Consola Bash, de exemplu, nu are nici o problemă în a trata expresii precum **ls dirname/*/*/*[2-3]**.

În alte console asteriscul este folosit pentru a ușura munca utilizatorilor, prin reducerea numărului de caractere care trebuie tastate: astfel puteți introduce **cd dir*** în loc de **cd directory**. În consola Bash nu este necesar să facem acest lucru, deoarece această unealtă GNU are o facilitate numită autocompletarea numelui fișierelor. Concret, dacă tastăti doar primele caractere ale comenzi (oriunde vă aflați în structura de fișiere) sau ale unui fișier (care aparține directorului curent) și dacă nu este posibilă nici o confuzie, consola va găsi ceea ce dumneavoastră doriti. De exemplu, dacă vă poziționați într-un director care conține multe fișiere, puteți verifica dacă există fișiere care încep cu litera A prin comanda **ls A** urmată de apăsarea tastei **Tab** de două ori în loc de apăsarea tastei **Enter**. Dacă există doar un singur fișier al cărui nume începe cu „A”, acest fișier vă fi prezentat imediat ca argument al comenzi **ls** (sau al oricărei comenzi).

3.3.3.2. Which

O cale foarte simplă prin care putem căuta fișiere este comanda **which**, prin care putem căuta în directoarele listate în variabila PATH a utilizatorului respectiv după fișierul dorit. Trebuie să știm că, deoarece variabila listează numai traseele directoarelor care conțin fișiere executabile, **which** nu ne spune nimic despre fișierele obișnuite. Comanda **which** este folositoare atunci când vă confruntați cu probleme de genul „command not found” - comandă lipsă, de negăsit. În exemplul de mai jos, utilizatoarea *tina* nu poate folosi programul acroread, în timp ce colegul ei nu are nici o problemă pe același sistem. Chestiunea este similară cu cea explicată în cazul variabilei PATH, într-o secțiune precedentă: colegul Tinei îi spune acesteia că programul există în /opt/acroread/bin, dar directorul nu este listat în variabila ei PATH:

```
tina:~> which acroread  
/usr/bin/which: no acroread in (/bin:/usr/bin:/usr/bin/X11)
```

Problema poate fi rezolvată prin tastarea comenzi ca traseu complet sau prin re-exportarea conținutului variabilei PATH:

```
tina:~> export PATH=$PATH:/opt/acroread/bin  
tina:~> echo $PATH  
/bin:/usr/bin:/usr/bin/X11:/opt/acroread/bin
```

Folosiți comanda **which** și pentru a verifica dacă o comandă este un alias al altiei comenzi:

```
gerrit:~> which -a ls  
ls is aliased to `ls -F --color=auto'  
ls is /bin/ls
```

În cazul în care comanda nu este executată de sistemul dumneavoastră, folosiți comanda **alias**:

```
tille@www:~/mail$ alias ls  
alias ls='ls --color'
```

3.3.3.3. Find și locate (Găsește și localizează)

Acestea sunt uneltele cele mai bune, care pot fi folosite și în căutarea fișierelor care nu sunt listate în traseele conținute de variabila PATH. **find**, aşa cum este cunoscută din UNIX, este o unealtă foarte

puternică, poate cu prețul unei sintaxe mai dificile. Dar comanda **find** din GNU se poate descurca cu problemele ridicate de sintaxă. Nu numai că vă permite căutarea după numele fișierelor, ci acceptă criterii ca mărimea fișierelor, data ultimei modificări precum și alte proprietăți ale fișierelor. Cea mai răspândită formă este cea prin care se caută fișiere:

```
find <path> -name <criterii_de_căutare>
```

Această expresie poate fi interpretată astfel: „Caută în toate subdirectoarele și fișierele determinate de traseul dat și afișează numele fișierelor care conțin în numele lor (nu în conținutul acestora) criteriu de căutare introdus”.

Alt mod în care putem folosi comanda **find** este căutarea fișierelor de o mărime determinată, după cum se observă în exemplul de mai jos, în care utilizatorul *peter* dorește să găsească toate fișierele din directorul curent sau din subdirectoarele acestuia care sunt mari de 5 MO:

```
peter:~> find . -size +5000k  
psychotic_chaos.mp3
```

Dacă veți căuta în paginile man, veți afla că **find** poate executa anumite operații asupra fișierelor pe care le găsește. Ca exemplu putem vorbi despre ștergerea fișierelor. Cel mai bine este să verificați dacă fișierul găsit este cel dorit de dumneavoastră, adică nu lansați comanda **file** cu opțiunea de execuție **-exec**, și doar după aceea reintroduceți comanda prin care ștergeți fișierul respectiv. Mai jos vă prezintăm o căutare a fișierelor de tipul **.tmp**:

```
peter:~> find . -name "*tmp" -exec rm {} \;  
peter:~>
```



Optimizați!

Această comandă va apela **rm** ori de câte ori este găsit un fișier ale cărui proprietăți se potrivesc cu criteriile de căutare. În cel mai rău caz, pot fi zeci de mii sau milioane de fișiere. Acest lucru va determina încărcarea sistemului.

Un mod de lucru mai realist este folosirea unei conexiuni (pipe – |) și a uneltei **xargs** împreună cu **rm** ca argument. În acest fel, comanda **rm** este apelată atunci când linia de comandă este plină, în loc să fie apelată la fiecare fișier găsit. Consultați [Capitolul 5](#) pentru mai multe amănunte despre utilizarea redirecționării I/O pentru a vă ușura sarcinile de zi cu zi.

Mai târziu (în 1999, dacă ne luăm după paginile man, după 20 de ani de **find**), a fost dezvoltată comanda **locate**. Este un program mai ușor de utilizat, dar cu mai multe restricții decât **find**, deoarece rezultatul afișat de această comandă este bazat pe un fișier care conține un index al unei baze de date actualizată o singură dată pe zi. Pe de altă parte, o căutare în baza de date a comenzi **locate** folosește mai puține resurse decât **find**, așa că rezultatele sunt afișate aproape instantaneu.

Cele mai multe distribuții de Linux folosesc acum **slocate**, adică o căutare care ia în considerare și aspecte legate de securitate, care nu permite utilizatorilor să aibă acces la un rezultat pe care nu au dreptul să-l citească. Fișierele din directorul personal al *administratorului* (*root*'s home directory) de exemplu, nu sunt, în mod normal, accesibile publicului. Un utilizator care dorește să găsească pe cineva care folosește consola C, poate introduce comanda **locate .cshrc**, pentru a-i se afișa toți utilizatorii care dețin un fișier personalizat pentru consola C. Presupunând că utilizatorii *root* și *jenny* folosesc consola C, atunci numai fișierul */home/jenny/.cshrc* va fi afișat, nu și acela din directorul personal al *administratorului*. Pe cele mai multe sisteme, **locate** este o legătură simbolică pentru programul **slocate**:

```
billy:~> ls -l /usr/bin/locate  
lrwxrwxrwx 1 root slocate 7 Oct 28 14:18 /usr/bin/locate -> slocate*
```

Utilizarea *tina* poate să folosească **locate** pentru a găsi aplicația de care are nevoie, după cum vă este prezentat în imaginea următoare:

```
tina:~> locate acroread
/usr/share/icons/hicolor/16x16/apps/acroread.png
/usr/share/icons/hicolor/32x32/apps/acroread.png
/usr/share/icons/locolor/16x16/apps/acroread.png
/usr/share/icons/locolor/32x32/apps/acroread.png
/usr/local/bin/acroread
/usr/local/Acrobat4/Reader/intellinux/bin/acroread
/usr/local/Acrobat4/bin/acroread
```

Directoarele care nu conțin numele bin nu pot avea programul – ele nu conțin fișiere executabile. Mai rămân trei posibilități. Fișierul din /usr/local/bin este cel pe care îl dorește *tina*: este o legătură către un script care pornește programul în sine:

```
tina:~> file /usr/local/bin/acroread
/usr/local/bin/acroread: symbolic link to ../Acrobat4/bin/acroread

tina:~> file /usr/local/Acrobat4/bin/acroread
/usr/local/Acrobat4/bin/acroread: Bourne shell script text executable

tina:~> file /usr/local/Acrobat4/Reader/intellinux/bin/acroread
/usr/local/Acrobat4/Reader/intellinux/bin/acroread: ELF 32-bit LSB
executable, Intel 80386, version 1, dynamically linked (uses
shared libs), not stripped
```

Pentru a păstra traseele cât mai scurte posibil, astfel încât sistemul să nu caute prea mult atunci când un utilizator dorește se execute o comandă, adăugăm /usr/local/bin variabilei PATH și nu celealte directoare, care conțin doar fișierele binare ale unui program anume, pe când /usr/local/bin conține și alte programe folositoare.

Vă spunem din nou că o descriere completă a facilităților comenziilor **find** și **locate** poate fi găsită în paginile Info.

3.3.3.4. Comanda grep

3.3.3.4.1. Sortarea liniară generală

Un program simplu dar foarte puternic, **grep**, este folosit pentru sortarea liniilor introduse și afișarea anumitor tipare ca rezultat al procesării. Există mii de aplicații ale programului **grep**. În exemplul de mai jos, *jerry* folosește **grep** pentru a vedea cum stau lucrurile cu **find**:

```
jerry:~> grep -a find .bash_history
find . -name userinfo
man find
find ... -name common.cfg
```



Căutarea în istoricul consolei

Foarte folositoare în aceste cazuri este funcția de căutare a consolei **bash**, activată de apăsarea tastelor **Ctrl+R** o singură dată, precum în exemplul care urmează, în care se dorește verificarea traseului apelat de ultima căutare realizată prin comanda **find**:

```
thomas ~> ^R
(reverse-i-search)`find': find `/home/thomas` -name *.xml
```

Introduceți sirul care conține criteriile dumneavoastră după prompterul de căutare. Cu cât tastăți mai multe caractere, căutarea devine mai precisă. Se citește istoricul comenziilor pentru această sesiune a consolei (care este scris în *.bash_history* din directorul dumneavoastră personal la închiderea sesiunii). Este afișată cea mai bună potrivire cu criteriile dumneavoastră de căutare. Dacă dorîți să vedeti comenziile anterioare care conțin același sir, apăsați din nou **Ctrl+R**.

Apelați la paginile Info despre **bash** pentru detalii.

Toate sistemele de operare UNIX care se respectă conțin un dicționar. La fel și Linux-ul. Dicționarul este o listă a cuvintelor cunoscute într-un fișier numit `words`, care se găsește în `/usr/share/dict` (în cazul Ubuntu, directorul este `/etc/dictionaries-common/words`). Pentru a verifica rapid ortografia unui cuvânt, nu este necesară nici o aplicație grafică:

```
william:~> grep pinguin /usr/share/dict/words
william:~> grep penguin /usr/share/dict/words
penguin
penguins
```



Dicționar vs. listă de cuvinte

Sunt și distribuții care oferă comanda `dict`, care oferă mai multe facilități decât căutarea unui cuvânt într-o listă.

Cine este deținătorul celuilalt director personal? Hei, iată numărul său de telefon!

```
lisa:~> grep gdbruyn /etc/passwd
gdbruyn:x:981:981:Guy Debruyne, tel 203234:/home/gdbruyn:/bin/bash
```

Și care era adresa de mail a lui Arno?

```
serge:~/mail> grep -i arno *
sent-mail: To: <Arno.Hintjens@celeb.com>
sent-mail: On Mon, 24 Dec 2001, Arno.Hintjens@celeb.com wrote:
```

`find` și `locate` sunt folosite adesea în combinație cu programul grep pentru a defini căutări mai rafinate. Pentru mai multe informații, consultați [Capitolul 5](#), partea despre redirecționarea I/O.

3.3.3.4.2. Caractere speciale

Atunci când folosiți în alte scopuri caracterele care au un înțeles aparte pentru consolă, ele trebuie să fie separate de acest înțeles special (*escaped*). Caracterul din Bash, precum și în alte console, care face acest lucru este linia oblică inversă (\); prin el se eludează înțelesul special al caracterului următor. Consola recunoaște aceste caractere speciale, printre care cele mai des întâlnite sunt /., ?, și *. Lista completă poate fi găsită în paginile Info și în documentația propriei consolei pe care o folosiți.

De exemplu, să spunem că dorîți să afișați fișierul „*” în loc de a afișa toate fișierele dintr-un director; atunci trebuie să folosiți

`less *`

Același lucru se aplică și numelor de fișiere care conțin un spațiu:

`cat This\ File`

3.3.4. Mai multe moduri în care puteți afișa conținutul unui fișier

3.3.4.1. Generalități

În afară de comanda `cat`, care nu prea face altceva în afară de a trimite fișierele la ieșirea standard, există alte unelte pentru a vizualiza conținutul unui fișier.

Cea mai usoară cale este, desigur, utilizarea interfeței grafice pentru utilizator în locul uneltelor linie de comandă. În introducerea acestei lucrări am aruncat o privire asupra unei aplicații de birou, OpenOffice.org. Alte exemple sunt GIMP (puteți să-l porniți cu `gimp` din linia de comandă), programul de manipulare a imaginilor din GNU (GNU Image Manipulation Program); `xpdf` pentru a vizualiza documentele în format PDF (Portable Document Format); GhostView (`gv`) pentru fișierele PostScript; Mozilla/Firefox, `links` (un navigator în modul text), Konqueror, Opera și multe altele pentru conținutul web; XMMS, Cdplay și altele pentru fișiere multimedia; AbiWord, Gnumeric, Koffice, etc. pentru tot felul de aplicații pentru birou și tot așa. Există mii de aplicații pentru Linux; pentru a le menționa pe toate avem nevoie de câteva zile.

În schimb, ne vom concentra pe aplicațiile pentru consolă sau aplicațiile pentru mediul text, pentru că ele formează baza tuturor celorlalte aplicații. Aceste comenzi se aplică în mediul text acelor fișiere care conțin text. Dacă aveți dubii legate de un fișier, verificați-l prin comanda **file**.

Să cunoaștem deci uneltele text pe care le avem la dispoziție pentru a privi în conținutul fișierelor.



Probleme legate de fonturi

Uneltele text simplu, precum cele pe care le vom discuta acum, au adesea probleme cu fișierele care conțin text „simplu”, din cauza codării folosite pentru fonturile din aceste fișiere. Caracterele speciale, cum sunt caracterele alfabetice care sunt accentuate, caracterele chinezești și alte caractere ale unor limbi ce folosesc un tipar diferit de codarea implicită *en_US*, sunt afișate incorrect sau înlocuite cu gunoaie care nu pot fi citite.

Aceste probleme sunt discutate în [Sectiunea 7.4](#).

3.3.4.2. „less înseamnă mai mult”

Fără îndoială că veți auzi pe cineva spunând această propoziție mai devreme sau mai târziu dacă lucrați într-un mediu UNIX. Puțină istorie a UNIX-ului este necesară pentru a explica acest lucru:

- La început a fost **cat**. Rezultatul comenzi era difuzat într-un mod incontrolabil.
- Apoi a fost **pg**, care poate fi găsită și acum pe unele UNIX-uri mai vechi. Această comandă afișează textul la ieșirea standard pagină cu pagină.
- Programul **more** a fost o versiune revizuită a lui **pg**. Această comandă este încă disponibilă pe orice sistem Linux.
- **less** este versiunea GNU a comenzi more și dispune de mai multe facilități care permit evidențierea criteriilor de căutare, derularea, etc. Sintaxa este foarte simplă:

less numele_fișierului (less name_file)

Mai multe informații sunt conținute de paginile Info.

Despre pagere aveți deja o idee deoarece ele sunt folosite pentru a vizualiza paginile de manuale.

3.3.4.3. Comenzile **head** și **tail**

Aceste două comenzi afișează primele/ultimele n linii din fișierul respectiv. Pentru a afișa ultimele 10 comenzi introduse, tastează:

```
tony:~> tail -10 .bash_history
locate configure | grep bin
man bash
cd
xawtv &
grep usable /usr/share/dict/words
grep advisable /usr/share/dict/words
info quota
man quota
echo $PATH
frm
```

head lucrează asemănător. Comanda **tail** are o opțiune folosită prin care arată continuu ultimele n linii ale unui fișier care are un conținut în permanentă schimbare. Această opțiune **-f** este utilizată de administratorii de sisteme pentru a verifica fișierele jurnal. Mai multe informații puteți găsi în fișierele de documentație ale sistemului.

3.3.5. Fișiere de legătură

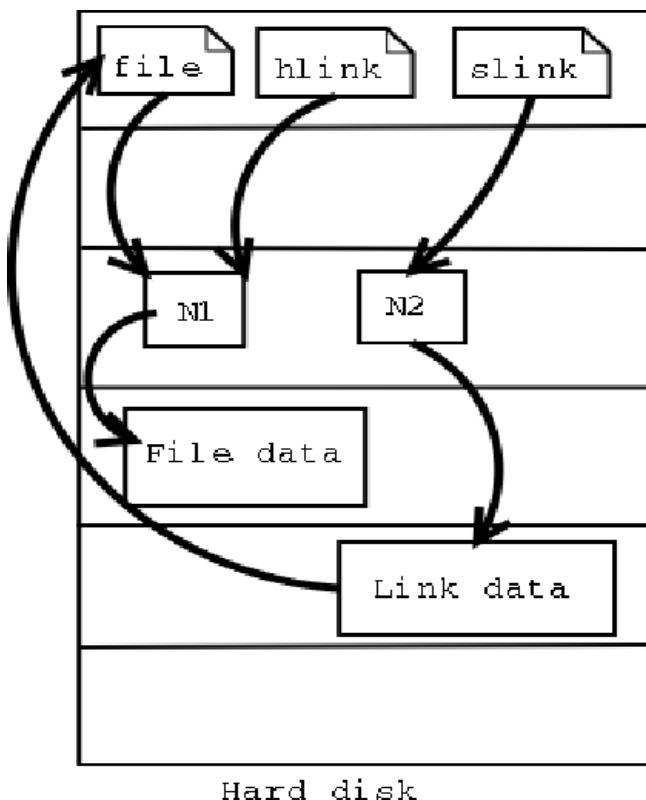
3.3.5.1. Tipuri de legături

Întrucât știm acum mai multe lucruri despre fișiere și reprezentarea acestora în sistemul de fișiere, înțelegerea legăturilor (sau scurtăturilor) este foare la ureche. O legătură nu este altceva decât o cale prin care potrivim două sau mai multe nume de fișiere în același set de date. Sunt două căi prin care obținem acest lucru:

- Legături fizice: asociază două sau mai multe nume de fișiere același nod. Legăturile fizice împart aceleași blocuri de date pe discul fix, în timp ce continuă să se comporte ca fișiere independente. Există un dezavantaj care sare în ochi: legăturile fizice nu pot sări peste partiții, deoarece numărul asociat nodului este unic doar pentru partită în cauză.
- Legături simbolice - numite și soft links (sau, pe scurt, symlink): un fișier mic care este un indicator către alt fișier. O legătură simbolică conține traseul către fișierul țintă, în schimbul locului fizic de pe discul fix al fișierului țintă. Deoarece nodurile nu sunt folosite în această metodă, legăturile simbolice pot traversa partiții.

Cele două tipuri de legături se comportă la fel, dar nu sunt același lucru, după cum puteți observa din schema de mai jos:

Figura 3-2. Mecanismul legăturilor fizice și simbolice



Trebuie să vă spunem că dacă ștergeți fișierul țintă al unei legături simbolice, acea legătură nu mai este folositoare.

Fiecare fișier obișnuit este, în principiu, o legătură fizică. Legăturile fizice nu pot traversa partiții, deoarece ele au ca referință nodurile (inodes), iar numărul unui nod este unic doar în interiorul unei partiții date.

Se poate argumenta că există un al treilea tip de legătură, legătura *utilizator-spațiu*, care este similară unei scurtături din sistemul MS Windows. Ele sunt fișiere care conțin meta-date care pot fi interpretate doar de administratorul grafic de fișiere. Pentru kernel și consolă, aceste fișiere sunt fișiere normale. Ele se pot termina cu sufixul (extensia) *.desktop* sau *.lnk*; exemple pot fi găsite în *~/gnome-desktop*:

```
[dupont@boulot .gnome-desktop]$ cat La\ Maison\ Dupont
[Desktop Entry]
Encoding=Legacy-Mixed
Name=La Maison Dupont
Type=X-nautilus-home
X-Nautilus-Icon=temp-home
URL=file:///home/dupont
```

lată un exemplu dintr-un sistem cu mediul grafic KDE:

```
[lena@venus Desktop]$ cat camera
[Desktop Entry]
Dev=/dev/sda1
FSType=auto
Icon=memory
MountPoint=/mnt/camera
Type=FSDevice
X-KDE-Dynamic-Device=true
```

Este ușor să faceți astfel de legături, folosind facilitățile puse la dispozitie de mediul grafic. Dacă aveți nevoie de ajutor, documentația sistemului trebuie să fie prima dumneavoastră resursă.

În secțiunea următoare vom studia realizarea legăturilor simbolice în stilul UNIX, folosind linia de comandă.

3.3.5.2. Realizarea legăturilor simbolice

Legăturile simbolice sunt interesante îndeosebi pentru utilizatorii începători: ele sunt ușor de văzut și nu trebuie să vă faceți griji în privința partitilor.

Comanda prin care facem legături este **In**. Pentru a realiza o legătură simbolică se folosește opțiunea **-s**:

In -s fișier_țintă nume_legătură (In -s targetfile linkname)

În exemplul de mai jos, utilizatorul *freddy* realizează o legătură simbolică într-un subdirector al directorului său personal cu altă parte a sistemului:

```
freddy:~/music> ln -s /opt/mp3/Queen/ Queen
freddy:~/music> ls -l
lrwxrwxrwx 1 freddy freddy 17 Jan 22 11:07 Queen -> /opt/mp3/Queen
```

Legăturile simbolice sunt fișiere foarte mici, pe când legăturile fizice au aceeași mărime ca fișierul original.

Aplicațiile legăturilor simbolice sunt variate. Ele sunt folosite pentru a salva spațiu pe disc, pentru a face o copie a unui fișier în ideea de a satisface cerințele de instalare ale unui program care se aşteaptă că un anume fișier să se afle într-un alt loc, sunt folosite pentru scripturile care trebuie să ruleze pe neașteptate în medii noi și vă pot scuti, în general, de multă muncă. Un administrator de sistem poate decide să mute directorul personal al unui utilizator într-un loc nou, discul 2 (disk 2), de exemplu, dar, dacă dorește ca totul să funcționeze ca înainte, cum ar fi fișierul /etc/passwd, cu un efort minim va realiza o legătură simbolică din directorul /home către noul loc /disk2/home.

3.4. Securitatea fișierelor

3.4.1. Drepturile de acces: prima linie de apărare a Linux-ului

Modelul de securitate din Linux este bazat pe cel folosit în sistemele UNIX și este la fel de rigid (câteodată chiar mai rigid) ca modelul UNIX-ului, care este oricum destul de robust. Într-un sistem Linux, orice fișier este deținut de un utilizator și un grup de utilizatori. Există, de asemenea, o a treia categorie de utilizatori, aceia care nu sunt utilizatori deținători și nici nu aparțin unui grup care deține un anume fișier. Pentru fiecare categorie de utilizatori pot fi acordate sau nu (interzise) permisiunile legate de citire, scriere

sau executare ale unui fișier.

Ați folosit deja opțiunea *completă* pentru a lista fișierele folosind comanda **ls -l**, deși din alte motive. Această comandă afișează permisiunile atașate fișierelor pentru aceste trei categorii de utilizatori; ele sunt indicate prin nouă caracter care urmează după primul caracter a cărui funcție este aceea de indicator al tipului de fișier, la începutul liniei care arată proprietățile fișierului. După cum se poate observa din exemplul următor, primele trei caractere din seria de nouă afișează drepturile de acces ale utilizatorului care deține fișierul. Următoarele trei sunt pentru grupul căruia aparține utilizatorul, iar ultimele trei, pentru ceilalți utilizatori. Permisiunile sunt listate întotdeauna în aceeași ordine: citire, scriere și executare pentru utilizator, grupul căruia aparține utilizatorul și pentru ceilalți. Iată câteva exemple:

```
marise:~> ls -l To_Do
-rw-rw-r-- 1 marise users 5 Jan 15 12:39 To_Do
marise:~> ls -l /bin/ls
-rwxr-xr-x 1 root root 45948 Aug 9 15:01 /bin/ls*
```

Primul fișier este un fișier obișnuit (proprietate arătată de primul caracter, „-“). Utilizarea cu numele *marise* sau utilizatorii care aparțin grupului *users* pot citi și scrie (adică schimba/muta/șterge) acest fișier, dar nu pot rula acest fișier ca program (a doua și a treia liniuță). Toți ceilalți utilizatori au doar permisiunea de a citi acest fișier, dar nu îl pot scrie sau rula/executa (cea de-a patra și a cincea liniuță).

Cel de-al doilea exemplu este un fișier executabil, diferență: oricine poate rula acest program, dar trebuie să fii *root*, *administrator*, pentru a-l schimba.

Paginile Info explică în ce mod comanda **ls** afișează permisiunile în detaliu; consultați secțiunea *Ce informații sunt listate*.

Pentru ca utilizarea comenzilor împreună cu fișierele să fie cât mai facilă, atât permisiunile sau modurile, drepturile de acces, cât și grupurile de utilizatori, au un cod. Consultați tabelul de mai jos.

Tabelul 3-7. Codurile care arată drepturile de acces.

Cod	Înțeles
0 sau -	Drepturile de acces asociate fișierului nu sunt acordate.
4 sau r	Categoria de utilizatori definită are drepturi de citire.
2 sau w	Categoria de utilizatori definită are drepturi de scriere.
1 sau x	Categoria de utilizatori definită poate rula fișierul.

Tabelul 3-8. Codurile care arată grupurile de utilizatori.

Cod	Înțeles
u	Permisiunile acordate utilizatorilor.
g	Permisiunile acordate grupurilor.
o	Permisiunile acordate celorlalți.

Această schemă este aplicată foarte strict, ceea ce permite un nivel destul de ridicat al securității, chiar dacă rețeaua nu este securizată. Printre alte funcții, schema de securitate asigură accesul utilizatorilor la programe, pune la dispoziție anumite fișiere considerate necesare tuturor utilizatorilor și protejează date sensibile, cum ar fi directoarele personale și fișierele de configurare ale sistemului.

Ar trebui să știți întotdeauna numele dumneavoastră de utilizator. Dacă nu, acesta poate fi afișat prin comanda **id**, care vă arată și grupul inițial de care aparțineți, eventual celealte grupuri al căror membru sunteți:

```
tilly:~> id
uid=504(tilly) gid=504(tilly) groups=504(tilly),100(users),2051(org)
```

De asemenea, numele dumneavoastră de utilizator este stocat în variabila de mediu **USER**:

```
tilly:~> echo $USER
tilly
```

3.4.2. Uneltele

3.4.2.1. Comanda chmod

O consecință normală a aplicării unor permisiuni stricte în privința drepturilor asupra fișierelor, care poate fi considerată căteodată chiar neplăcere, este aceea că drepturile de acces vor trebui schimbate din mai multe motive. Folosim comanda **chmod** pentru a face acest lucru, iar *to chmod aproape că a devenit un verb acceptat al limbii engleze, însemnând schimbarea modului de acces asupra unui fișier. Comanda chmod poate fi folosită cu opțiuni numerice sau alfanumerice, în funcție de care vă plac mai mult.*

Exemplul de mai jos arată opțiunile alfanumerice folosite pentru rezolvarea unei probleme cu care se confruntă utilizatorii începători:

```
asim:~> ./hello
bash: ./hello: bad interpreter: Permission denied

asim:~> cat hello
#!/bin/bash
echo "Hello, World"

asim:~> ls -l hello
-rw-rw-r-- 1 asim asim 32 Jan 15 16:29 hello

asim:~> chmod u+x hello

asim:~> ./hello
Hello, World

asim:~> ls -l hello
-rwxrwxr-- 1 asim asim 32 Jan 15 16:29 hello*
```

Operanții + și – sunt folosiți pentru a acorda sau interzice drepturile unui grup. Combinățiile separate prin virgule sunt permise. Paginile Info și man conțin exemple foarte folosoitoare. Vă prezentăm încă unul, care arată cum fișierul din exemplul anterior este transformat într-un fișier privat al utilizatorului *asim*:

```
asim:~> chmod u+rwx,go-rwx hello

asim:~> ls -l hello
-rwx----- 1 asim asim 32 Jan 15 16:29 hello*
```

Problema legată de mesajul de eroare care spune că nu ai permisiuni se datorează, în cele mai multe cazuri, aspectelor legate de drepturile de acces. Comentarii precum „ieri a funcționat” sau „dacă rulez acest fișier ca root, merge” sunt datorate, în principal, permisiunilor greșit alocate.

Atunci când folosim chmod cu argumente numerice, valorile fiecărui drept de acces acordat trebuie aplicate la nivelul grupului. De aceea avem un număr format din trei cifre, care conține valorile simbolice ale setărilor pe care chmod trebuie să le aplice. Tabelul următor vă arată cele mai întâlnite combinații:

Tabelul 3-9. Protejarea fișierelor cu chmod

Comanda	Înțelesul comenții
chmod 400 fișier	Pentru protejarea unui fișier de o suprascriere accidentală.
chmod 500 director	Pentru a vă opri pe dumneavoastră să ștergeți, redenumiți sau să mutați accidental fișiere care aparțin acestui director.
chmod 600 fișier	Un fișier privat, care poate fi schimbat doar de utilizatorul care a introdus această comandă.
chmod 644 fișier	Un fișier care poate fi accesat public dar care poate fi schimbat doar de utilizatorul care a introdus această comandă.
chmod 660 fișier	Utilizatorii care aparțin grupului tău pot schimba acest fișier pe când ceilalți nu au nici un fel de drepturi asupra lui.
chmod 700 fișier	Numai utilizatorul are drepturi depline, ceilalți, indiferent de grupul aparținător, nu au nici un fel de drepturi.

chmod 755 director	Pentru fișierele care trebuie să fie citite sau rulate și de către alți utilizatori, dar scrise (schimbată) doar de utilizatorul care a introdus comanda descrisă.
chmod 775 fișier	Modul standard de acordare a permisiunilor pentru un grup.
chmod 777 fișier	Oricine poate face orice cu acest fișier.

3.4.2.2. Autentificarea într-un alt grup

Atunci când tastați comanda **id** într-o linie de comandă, primiți o listă a tuturor grupurilor cărora este posibil să le apartineți, precedate de numele dumneavoastră de utilizator plus ID-ul și numele grupului plus ID-ul cu care sunteți conectat în acel moment. Însă, în multe sisteme Linux, poate fi activ doar într-un singur grup. Inițial, acest grup activ sau *primar* este acela care îți este asociat de fișierul /etc/passwd. Cel de-al patrulea câmp al acestui fișier arată ID-ul grupului primar al utilizatorului, care este verificat prin fișierul /etc/group. Iată un exemplu:

```
asim:~> id
uid=501(asim) gid=501(asim) groups=100(users),501(asim),3400(web)

asim:~> grep asim /etc/passwd
asim:x:501:501:Asim El Baraka:/home/asim:/bin/bash

asim:~> grep 501 /etc/group
asim:x:501:
```

Cel de-al patrulea câmp în linia din fișierul /etc/passwd conține valoarea „501”, care reprezintă grupul *asim* în exemplul de mai sus. Din /etc/group putem afla numele care corespunde acestui ID de grup. Acestui grup, și nu altuia, va apartine *asim* la autentificarea inițială în sistem.



Schema de grup privat al utilizatorului

Pentru a permite o flexibilitate mare, cele mai multe sisteme Linux folosesc o astă numită *schemă de grup privat al utilizatorului*, care asociază fiecare utilizator cu grupul său primar. Acest grup este un grup care conține numai acest utilizator particular, de unde numele de *grup privat*. De regulă, acest grup are același nume cu numele utilizatorului, ceea ce poate genera anumite confuzii.

În afară de acest grup privat, utilizatorul *asim* poate apartine și grupurilor *user* și *web*. Deoarece aceste grupuri sunt grupuri secundare pentru utilizatorul nostru, el trebuie să folosească **newgrp** pentru a se autentifica în oricare din ele (folosiți mai întâi **gpasswd** pentru a stabili parole pentru aceste grupuri). În exemplul care urmează, *asim* dorește să creeze fișiere care să fie deținute de grupul *web*.

```
asim:/var/www/html> newgrp web

asim:/var/www/html> id
uid=501(asim) gid=3400(web) groups=100(users),501(asim),3400(web)
```

Acum, în momentul în care *asim* creează fișiere noi, acestea vor fi deținute de grupul *web* în loc să fie deținute de grupul *asim*:

```
asim:/var/www/html> touch test

asim:/var/www/html> ls -l test
-rw-rw-r-- 1 asim web 0 Jun 10 15:38 test
```

Autentificarea într-un grup nou vă ajută să preîntâmpinați folosirea comenzi **chown** (vezi [Sectiunea 3.4.2.4](#)) sau chemarea administratorului de sistem pentru a schimba permisiunile asupra fișierelor pentru dumneavoastră.

Consultați paginile man ale comenzi **newgrp** pentru mai multe informații.

3.4.2.3. Masca fișierelor

Când un fișier nou este salvat undeva, devine, în primul rând, subiectul procedurilor standard de securitate. Fișiere care să nu aibă alocate permisiuni nu există în Linux. Permisiunile standard, care sunt

alocate implicit fișierelor noi, sunt determinate de *mască* (*mask*). Valoarea numerică a măștii poate fi afișată prin comanda **umask**:

```
bert:~> umask  
0002
```

În loc de a adăuga valori simbolice fiecarui fișier, cum se întâmplă prin folosirea **chmod**, pentru a calcula permisiunile asociate unui nou fișier, ele sunt scăzute din numărul care arată drepturile de acces posibile. În exemplul de mai sus, totuși, observăm afișate patru valori, în timp ce știm că există doar trei categorii cărora li se aplică permisiuni: *utilizator*, *grup* și *alții*. Primul zero este parte a atributelor speciale de configurare ale unui fișier, despre care vom discuta în [Secțiunea 3.4.2.4](#) și [Secțiunea 4.1.6](#). Se poate întâmpla ca acest prim zero să nu fie afișat de sistemul dumneavoastră atunci când introduceți comanda **umask** și să vedeați doar trei numere care arată masca implicită pentru crearea de fișiere.

Orice sistem asemănător UNIX-ului are o funcție pentru crearea de noi fișiere, care este apelată ori de câte ori un utilizator folosește un program care creează fișiere noi, de exemplu când descărcați un fișier de pe Internet, când salvați un nou document text și aşa mai departe. Această funcție creează atât fișiere cât și directoare. La crearea unui director nou, această funcție va acorda permisiuni pentru citire, scriere și executare pentru toate categoriile de utilizatori. La crearea unui fișier nou, funcția va acorda permisiuni de citire și scriere pentru toți, dar nici o categorie de utilizatori nu va avea permisiunea de executare a aceluui fișier. De aceea, înainte ca masca să fie aplicată, un director are permisiunea 777 sau *rxrwxrwx*, un fișier obișnuit 666 sau *rw-rw-rw*.

Valoarea *umask* este scăzută din aceste permisiuni implicate după ce funcția a creat un fișier sau un director nou. Astfel, un director va avea permisiunea implicită 775 și un fișier 664, dacă valoarea măștii este (0)002. Acest lucru vă este demonstrat în exemplul de mai jos:

```
bert:~> mkdir newdir  
  
bert:~> ls -ld newdir  
drwxrwxr-x    2 bert      bert          4096 Feb 28 13:45 newdir/  
  
bert:~> touch newfile  
  
bert:~> ls -l newfile  
-rw-rw-r--    1 bert      bert          0 Feb 28 13:52 newfile
```



Fișiere vs. directoare

Un director va primi implicit mai multe permisiuni: el va avea întotdeauna permisiuni de *executare*. Dacă nu le-ar avea, directorul nu ar fi accesibil. Încercați acest lucru prin schimbarea permisiunilor unui director (chmodding) în 644!

Dacă vă autentificați într-un alt grup folosind comanda **newgrp**, masca rămâne neschimbată. De aceea, în cazul în care este stabilită ca 002, fișierele și directoarele pe care le creați în timp ce sunteți în grupul nou vor fi accesibile și celorlalți membri ai grupului; nu trebuie să folosiți **chmod**.

Utilizatorul root are, de obicei, permisiuni inițiale asociate fișierelor noi mai stricte:

```
[root@estoban root]# umask  
022
```

Această valoare implicită este extinsă asupra întregului sistem prin fișierele de configurare a resurselor consolei, de exemplu */etc/bashrc* sau */etc/profile*. O puteți schimba din fișierul de configurare a consolei, vezi [Capitolul 7](#), despre personalizarea mediului consolei.

3.4.2.4. Modificarea apartenenței la un utilizator sau la un grup

Când un fișier figurează ca fiind deținut de un utilizator sau un grup despre care credeți că nu ar fi îndreptățit să o facă, puteți repara eroarea prin comenziile **chown** (schimbă utilizatorul) sau **chgrp** (schimbă grupul). Schimbarea deținătorilor unui fișier este o sarcină des întâlnită de administratorii de sistem în mediile în care este nevoie ca fișierele să fie partajate de către grupuri de utilizatori. Ambele comenzi sunt flexibile, după cum puteți afla prin opțiunea *-help*.

Comanda **chown** poate fi aplicată pentru a schimba deținătorii unui fișier atât la nivel de utilizator,

cât și la nivel de grup, pe când **chgrp** schimbă doar grupul care detine acel fișier. Desigur, sistemul va verifica dacă utilizatorul care introduce aceste schimbări are suficiente permisiuni asupra fișierului sau fișierelor asupra cărora dorește să folosească aceste comenzi.

Pentru a modifica doar utilizatorul deținător al unui fișier, folosiți sintaxa de mai jos:

chown utilizator_nou fișier

Dacă folosiți două puncte după numele de utilizator (consultați paginile Info), va fi modificat și grupul deținător, în grupul primar al utilizatorului care introduce comanda. Pe sistemele Linux, fiecare utilizator are propriul său grup, astfel că această formă a comenzi poate fi folosită pentru a crea fișiere private:

```
jacky:~> id  
uid=1304(jacky) gid=(1304) groups=1304(jacky),2034(pproject)  
  
jacky:~> ls -l my_report  
-rw-rw-r-- 1 jacky project 29387 Jan 15 09:34 my_report  
  
jacky:~> chown jacky: my_report  
  
jacky:~> chmod o-r my_report  
  
jacky:~> ls -l my_report  
-rw-rw--- 1 jacky jacky 29387 Jan 15 09:34 my_report
```

Dacă *jacky* va dori să partajeze acest fișier, fără să ofere oricui permisiuni de scriere, poate folosi comanda **chgrp**:

```
jacky:~> ls -l report-20020115.xls  
-rw-rw--- 1 jacky jacky 45635 Jan 15 09:35 report-20020115.xls  
  
jacky:~> chgrp project report-20020115.xls  
  
jacky:~> chmod o= report-20020115.xls  
  
jacky:~> ls -l report-20020115.xls  
-rw-rw--- 1 jacky project 45635 Jan 15 09:35 report-20020115.xls
```

Astfel, utilizatorii din grupul *project* vor putea să lucreze cu acest fișier. Utilizatorii care nu aparțin grupului menționat nu au nici o treabă cu acest fișier.

Ambele comenzi, **chown** și **chgrp**, pot fi folosite pentru schimbarea deținătorilor recursiv, prin opțiunea **-R**. În acest caz, toate subdirectoarele și fișierele conținute într-un director vor apartine utilizatorului sau/și grupului dat.



Restricții

Pe cele mai multe sisteme, folosirea comenziilor **chown** și **chgrp** este restricționată în cazul utilizatorilor obișnuiți. Dacă nu sunteți administratorul sistemului, nu puteți schimba utilizatorul sau grupul care detine un anumit fișier, din motive care îți de securitatea sistemului. Dacă utilizarea acestor comenzi nu ar fi restricționată, utilizatorii rău-intenționați pot modifica apartenența fișierelor la un utilizator sau la un grup și ar putea schimba mediul de lucru al utilizatorilor legitimi sau chiar distrugă fișierele celorlalți utilizatori.

3.4.2.5. Moduri speciale

Pentru ca administratorii de sistem să nu fie deranjați tot timpul pentru a rezolva probleme legate de drepturile de acces, drepturi speciale pot fi alocate directoarelor în întregime sau numai programelor. Există trei moduri speciale:

- Modul bit-ului lipicios (persistent – sticky bit mode): după executarea unei sarcini, comanda este păstrată în memoria sistemului. La începuturi, acest mod era folosit pe scară largă pentru a salva memorie: sarcinile mari erau încărcate în memorie doar o singură dată. În zilele noastre, memoria nu mai este aşa de scumpă și există tehnici mai bune de gestionare a acesteia, aşa că nu mai este folosit, pentru capacitatea de optimizare conținute, asupra fișierelor singulare. Totuși, aplicat asupra

unui întreg director, acest mod are un alt înțeles. În acest caz, un utilizator poate schimba fișierele conținute de directorul în cauză doar dacă este deținătorul acelui fișier sau dacă permisiunile asociate fișierului acordă dreptul de a face modificări. Acest mod este utilizat pentru directoare ca /var/tmp, care trebuie să fie accesibile tuturor, dar în care un utilizator oarecare nu are voie să schimbe sau să steargă datele care aparțin celorlalți. Bit-ul lipios este indicat printr-un *t* la sfârșitul liniei care arată permisiunile alocate unui fișier:

```
mark:~> ls -ld /var/tmp
drwxrwxrwt 19 root      root          8192 Jan 16 10:37 /var/tmp/
```

Modul este aplicat prin comanda **chmod o+t directory**. Originile istorice ale lui „t” se găsesc în facilitatea proprie UNIX-ului *save Text acces*.

- SUID (stabilită identitatea utilizatorului – set user ID) și SGID (stabilită identitatea grupului – set group ID): sunt reprezentate de caracterul *s* în linia care arată permisiunile alocate unui fișier. Când acest mod este aplicat unui fișier executabil, el va rula cu permisiunile utilizatorului și ale grupului, în loc de acelea ale utilizatorului care introduce comanda, oferind acces la resursele sistemului. Vom aprofunda aceste lucruri în [Capitolul 4](#).
- SGID (stabilită identitatea grupului) asupra unui director: în acest caz aparte, fișierele create în directorul în cauză vor apartine aceluiași grup căruia aparține directorul care le conține (în timp ce comportamentul normal este ca fișierele noi să fie deținute de utilizatorii care le creează). Astfel, utilizatorii nu trebuie să se gândească la aparținătorii unor fișiere care trebuie partajate:

```
mimi:~> ls -ld /opt/docs
drwxrws--- 4 root      users          4096 Jul 25 2001 docs/
mimi:~> ls -l /opt/docs
-rw-rw---- 1 mimi      users        345672 Aug 30 2001-Council.doc
```

Acesta este modul standard prin care se partajează fișierele în UNIX.



Fișierele existente nu sunt modificate!

Fișierele care sunt mutate într-un director SGID, dar care sunt create în altă parte, păstrează utilizatorul și grupul deținător (aparținător) inițial. Acest comportament poate produce confuzii.

3.5. Sumar

În UNIX, cât și în Linux, toate entitățile sunt prezentate sistemului, într-un fel sau altul, ca fișiere care au asociate anumite proprietăți. Utilizarea traseelor (predefinite) permite utilizatorilor și administratorilor de sistem să găsească, citească și să mânuiască fișierele.

Am făcut primii pași pe calea care ne va transforma în experți: am discutat structura adeverată și cea presupusă a unui sistem de fișiere și v-ați format o imagine despre modelul de securitate a fișierelor din Linux, cât și despre alte câteva precauții de securitate care sunt aplicate implicit în orice sistem.

Consola este cea mai importantă unealtă prin care interacționăm cu sistemul. În acest capitol ați învățat câteva comenzi pentru consolă, pe care vi le prezintăm din nou în tabelul de mai jos:

Comanda	Înțelesul comenzii
bash	Consola GNU.
cat fișier(e)	Trimite conținutul unui fișier (unor fișiere) la ieșirea standard.
cd director	Schimbă directorul.
chgrp grup_nou fișier(e)	Schimbă grupul deținător al unui fișier cu grupul nou.
chmod mod fișier(e)	Modifică permisiunile de acces asupra fișierului.
chown utilizator_nou [:grup_nou] fișier(e)	Schimbă utilizatorul și grupul deținător.

cp fișierul_sursă fișierul_țintă	Copiază fișierul sursă în fișierul țintă.
df fișier	Prezintă spațiul utilizat pe partitura discului care conține fișierul în cauză.
echo șir	Afișează șirul de caractere introdus.
export	Partea a comenziilor proprii consolei care anunță sistemului variabilele și valorile acestora.
file nume_fișier	Determină tipul unui fișier.
find traseu expresie	Găsește fișierele în ierarhia sistemului de fișiere.
grep TIPAR fișier	Afișează liniile din fișier care conțin tiparul introdus.
head fișier	Trimite prima parte a unui fișier la ieșirea standard.
id	Afișează numele reale și efective ale utilizatorilor și grupurilor.
info comandă	Afișează documentația disponibilă pentru comanda în cauză.
less fișier	Este arătat fișierul cu ajutorul unui vizualizator mai puternic.
ln fișier_țintă nume_legătură	Realizează o legătură numită nume_legătură către un fișier_țintă
locate criteriu_căutare	Afișează toate fișierele accesibile care au în nume elemente ale criteriului de căutare.
ls fișier(e)	Afișează conținutul directorului.
man comandă	Afișează manualele sistemului pentru comanda în cauză.
mkdir director_nou	Creează un director nou, gol.
mv fișier_vechi fișier_nou	Redenumește sau mută un fișier.
newgrp nume_grup	Autentificare într-un grup nou.
pwd	Prezintă directorul curent în care lucrezi.
quota	Arată limitele alocate și gradul de utilizare a discului.
rm fișier	Șterge fișiere și directoare.
rmdir fișier	Șterge directoare.
tail fișier	Afișează ultima parte a unui fișier.
umask [valoare]	Arată sau schimbă modul în care sunt create fișiere noi.
wc fișier	Numără liniile, cuvintele și caracterele dintr-un fișier.
which comandă	Arată traseul complet al comenzi în cauză.

Subliniem iarăși că ar trebui SĂ CITIȚI PAGINILE DE MANUALE. Documentația este trusa dumneavoastră de prim-ajutor și ea conține răspunsuri la multe întrebări. Lista de mai sus prezintă comenziile de bază, cu care vă veți întâlni în fiecare zi, dar acestea pot face mult mai multe lucruri decât cele pe care vi le-am arătat aici. Citirea documentației vă oferă posibilitatea de a controla totul aşa cum dorîți.

În sfârșit, dar având destulă importanță, o trecere în revistă a permisiunilor asociate fișierelor:

Tabelul 3-11. Permisiunile asociate fișierelor

Cine\Ce	r(ead) citire	w(rite) scriere	(e)x(ecute) execută
u(ser) - utilizator	4	2	1
group) - grup	4	2	1
o(thers) - alții	4	2	1

3.6. Exerciții

Autentificați-vă ca utilizator obișnuit (neprivilegiat).

3.6.1. Partiții

- Afipați traseul dumneavoastră de căutare.
- Exportați o cale inexistentă, de exemplu `export PATH=blah` și încercați să listați conținutul directorului.
- Care este traseul directorului dumneavoastră personal? Cum poate un alt utilizator să ajungă la directorul dumneavoastră personal, folosind un traseu relativ?
- Mergeți în directorul `tmp` din `var`.
- Mergeți în directorul `share` din `usr` folosind o singură comandă. Schimbați directorul cu `doc`. Care este directorul curent de lucru?

3.6.3. Turul sistemului

- Mergeți în directorul `/proc`.
- Pe ce procesor rulează sistemul dumneavoastră?
- Cât de multă RAM (memorie cu acces aleator) folosește acum?
- Cât spațiu alocat pentru swap aveți?
- Ce drivere sunt încărcate?
- De câte ore rulează sistemul?
- Ce sisteme de fișiere sunt recunoscute în sistem?
- Mutati-vă în `/etc/rc.d` | `/etc/init.d` | `/etc/runlevels` și alegeți directorul potrivit nivelului de executare (rulare) al sistemului dumneavoastră.
- Ce servicii ar trebui să ruleze în acest nivel?
- Ce servicii rulează în modul grafic și nu rulează în modul text?
- Mutati-vă în `/etc`.
- Cât timp păstrează sistemul fișierele jurnal în care sunt monitorizate autentificările utilizatorilor?
- Ce versiune folosiți?
- Există alte notificări sau mesaje ale zilei?
- Câți utilizatori există pe sistemul dumneavoastră? Nu îi numărați, lăsați calculatorul să facă acest lucru în locul dumneavoastră.
- Câte grupuri?
- Unde este păstrată informația referitoare la zona de timp?
- Sunt instalate în calculator informații de tipul CUM SĂ?
- Mutati-vă în `usr/share/doc`.
- Numiți trei programe care sunt incluse în pachetul GNU `coreutils`.
- Ce versiune a consolei (`bash`) este instalată în sistemul dumneavoastră?

3.6.4. Mânuirea fișierelor

- Creați un director nou în directorul dumneavoastră personal.
 - Puteți muta acest director pe același nivel cu directorul personal?
 - Copiați toate fișierele XPM din `/usr/share/pixmaps` în directorul cel nou. Ce înseamnă XPM?
 - Afizați toate fișierele în ordine alfabetică inversă.
 - Mutati-vă în directorul personal. Creați un director nou și copiați toate fișierele din directorul `/etc` în acesta. Asigurați-vă că ati copiat și toate fișierele și directoarele care sunt subdirectoare ale lui `/etc!` (copiere recursivă)
 - Mergeți în directorul nou și creați un director pentru fișierele care încep cu literă mare și un altul pentru fișierele care încep cu o literă mică. Mutati toate fișierele în directorul potrivit. Folosiți cât mai puține comenzi posibil.
 - Ștergeți fișierele care v-au rămas.
 - Ștergeți tot directorul și conținutul acestuia printr-o singură comandă.
 - Folosiți **grep** pentru a afla care script pornește serviciul fonturi (Font Server) în nivelul de rulare grafic.
 - Unde se află programul serviciului *sendmail*?
 - Creați o legătură simbolică în directorul dumneavoastră personal către `/var/tmp`. Verificați și dacă funcționează.
 - Creați altă legătură simbolică în directorul personal către legătura existentă. Verificați dacă funcționează. Ștergeți prima legătură și listați conținutul directorului. Ce s-a întâmplat cu cea de-a doua legătură?
-

3.6.5. Permisii asociate fișierelor

- Puteți schimba permisiunile asociate directorului personal `/home`?
 - Care este modul implicit pentru crearea fișierelor în cazul dumneavoastră?
 - Schimbați deținătorii directorului `/etc` în utilizatorul și grupul asociate dumneavoastră.
 - Schimbați permisiunile asociate fișierului `~/.bashrc` în aşa fel încât numai dumneavoastră și grupul dumneavoastră primar să îl poată citi.
 - Introduceți comanda **locate root**. Observați ceva deosebit?
 - Creați o legătură simbolică care este îndreptată către `/root`. Poate fi folosită?
-

Capitolul 4. Procesele

Alături de fișiere, procesele sunt cele mai importante lucruri într-un sistem UNIX/Linux. În acest capitol, vom prezenta îndeaproape aceste procese. Vom studia mai multe lucruri despre:

- ◆ Procesarea multi-utilizatori și multi-sarcini
 - ◆ Tipuri de procese
 - ◆ Controlarea proceselor cu semnale diferite
 - ◆ Atributele proceselor
 - ◆ Ciclul de viață al unui proces
 - ◆ Pornirea și oprirea sistemului
 - ◆ SUID și SGID
 - ◆ Viteza sistemului și răspunsul
 - ◆ Procese programate
 - ◆ Sistemul cron Vixie
 - ◆ Cum să obțineți cele mai bune performanțe de la sistemul dumneavoastră
-

4.1. Anatomia proceselor

4.1.1. Multi-utilizatori și multi-sarcini (multi-user și multi-tasking)

Pentru că acum v-ați obișnuit cu mediul de lucru și sunteți capabili să comunicați cât de cât cu sistemul, a venit vremea să studiem mai în detaliu procesele pe care le putem porni. Nu toate comenziile pornesc un singur proces. Anumite comenzi inițiază o serie de procese, cum este **mozilla**; altele, precum **ls**, sunt executate ca o singură comandă.

Mai mult, Linux-ul este bazat pe UNIX, în care comportamentul obișnuit este acela prin care mai mulți utilizatori lansează comenzi diferite, în același timp, pe același sistem. Devine evident faptul că a trebuit să fie luate măsuri pentru ca procesorul (CPU) să poată gestiona toate procesele și să fie asigurată o cale prin care utilizatorii să disponă de aceste procese. În anumite cazuri, procesele trebuie să ruleze chiar și atunci când utilizatorul care le-a inițiat se deautentifică. De asemenea, utilizatorii au nevoie de mijloace prin care să reactiveze procesele întrerupte.

Vom explica structura proceselor din Linux în secțiunile următoare.

4.1.2. Tipuri de procese

4.1.2.1. Procese interactive

Procesele interactive sunt inițializate și controlate printr-o sesiune terminal. Cu alte cuvinte, trebuie ca cineva să fie conectat la sistem pentru a porni aceste procese; ele nu pot porni automat ca parte a funcțiilor sistemului. Aceste procese pot rula în prim-plan, ocupând terminalul care a pornit programul și nu puteți porni alte procese atât timp cât procesele existente în prim-plan rulează. Alternativ, ele pot rula în fundal, astfel încât consola în care ați pornit programul poate accepta comenzi noi în timp ce programul este în desfășurare. Până acum, ne-am concentrat în principal pe programele care rulează în prim-plan – timpul necesar pentru execuția lor a fost prea scurt pentru a fi observate, dar vizualizarea unui fișier cu ajutorul comenzi **less** este un bun exemplu de comandă care ocupă o sesiune a consolei. În acest caz, programul activat așteaptă ca dumneavastră să faceți ceva. Programul este conectat la terminal din momentul în care a fost pornit și acest terminal este folosit numai dacă introduceți comenzi pe care programul le poate înțelege. Alte comenzi vor returna erori sau sistemul nu va da nici un răspuns.

Însă în cazul în care un proces rulează în fundal, utilizatorul poate face alte lucruri în consola din care a pornit acel program, chiar dacă programul este în desfășurare.

Consola dispune de o funcție numită *controlul sarcinilor* (*job control*), prin care gestionează cu ușurință mai multe procese. Funcția conține un mecanism care comută procesele din prim-plan în fundal. Prin acest sistem, programele pot fi pornite în fundal imediat.

Rularea unui proces în fundal este folositoare doar programelor care nu folosesc instrucțiuni de la utilizator (prin intermediul consolei). Plasarea unei sarcini în fundal se face, îndeosebi, atunci când executarea acelei sarcini necesită o perioadă de timp mai mare. Pentru a elibera un terminal după introducerea unei comenzi, adăugați după comandă caracterul „ & ”. În exemplul următor, în care se lucrează în mediul grafic, este deschis un alt terminal din interiorul precedentului:

```
billy:~> xterm &
[1] 26558

billy:~> jobs
[1]+  Running                  xterm &
```

Toate facilitățile funcției de control a sarcinilor sunt explicate în detaliu în paginile Info ale consolei (**bash**), aşa că doar cele mai folosite aplicații de control ale sarcinilor sunt listate aici:

Tabelul 4-1. Controlul proceselor

(partea de) comandă	Semnificație
comandă_obișnuită	Rulează această comandă în prim-plan.
comandă &	Rulează această comandă în fundal (eliberează terminalul).
jobs	Arată comenziile care sunt executate în fundal.
Ctrl+Z	Suspendă (dar nu întrerupe, nu termină) un proces care rulează în prim-plan.
Ctrl+C	Întrerupe (termină și stopează) un proces care rulează în prim-plan.
%n	Fiecare proces care rulează în fundal primește un număr prin care poate fi identificat. Folosind expresia % puteți să apelați acea sarcină, precum în fg %2 .
bg	Reactivează un program care a fost suspendat în fundal.
fg	Trece o sarcină din fundal în prim-plan.
kill	Termină procesul (consultați și secțiunea Shell Built-in Commands - comenziile proprii consolei din paginile Info despre bash - console, terminale).

Mai multe exemple veți găsi în exercițiile care vor urma.

Cele mai multe sisteme UNIX au posibilitatea de a rula programul **screen**, care se dovedește folositor în cazul în care dorîți să introduceți o comandă în altă consolă. Atunci când apelați acest program, este inițiată o altă sesiune a terminalului, împreună cu eventuala comandă specificată, pe care le puteți detașa. În această sesiune puteți face orice dorîți. Toate programele și operațiile vor rula independent de consola primară. Puteți detașa sesiunea în timp ce programele pe care le-ați pornit vor continua să ruleze, chiar dacă vă veți deautentifica din consola originară, pentru ca, oricând dorîți, să restaurați acel **screen**.

Acest program a fost dezvoltat mai demult, când nu se inventaseră consolele virtuale și totul trebuia făcut într-un singur terminal text. Pentru cei dependenți, a fost păstrat pe sistemele Linux, deși există console virtuale de aproape zece ani.

4.1.2.2. Procese automate

Procesele automate (sau periodice, intermitente, continue/discontinue - batch) nu sunt executate în terminal. Acestea sunt sarcini care pot fi gestionate într-o zonă de așteptare, de unde vor fi lansate în execuție pe criteriul primul venit - primul plecat (FIFO – first-in, first-out – basis). Astfel de sarcini pot fi executate luând în considerare unul din următoarele două criterii:

- La un moment de timp bine determinat: se face prin folosirea comenzi **at**, pe care o vom discuta în partea a doua a acestui capitol.
- În momentele în care sistemul este mai puțin solicitat, astfel încât poate prelua și alte sarcini, alegere care se face prin comanda **batch**. Implicit, sarcinile sunt aranjate într-o coadă de așteptare până ce sistemul coboară la un nivel de ocupare mai mic de 0,8. În rețelele mari, administratorul de sistem preferă mai degrabă acest mod, pentru că sistemele trebuie să prelucreze mari cantități de date sau pentru a se evita ca sarcini care solicită multe resurse de sistem să fie executate de un sistem deja încărcat. Procesarea batch este folosită și pentru optimizarea performanțelor sistemului.

4.1.2.3. Daemons (Procese daemon)

Daemon-ii sunt procese care rulează neîncetat. De cele mai multe ori, ele sunt inițializate la pornirea sistemului și apoi așteaptă în fundal până ce sunt solicitate. Un exemplu tipic este procesul de rețea *xinetd*, care este pornit aproape la orice procedură de boot. După ce sistemul este pornit, acest proces de rețea așteaptă până ce un program client, precum un client FTP, solicite conectarea.

4.1.3. Atributele proceselor

Un proces are o serie de caracteristici, care pot fi văzute cu ajutorul comenzi **ps**:

- ID-ul procesului sau PID: un număr unic de identificare folosit pentru apelarea procesului.
- ID-ul procesului părinte sau PPID: numărul procesului (PID) care a pornit procesul în cauză.
- Numărul de amicică: gradul de atitudine prietenoasă al acestui proces față de celelalte procese (a nu se confunda cu prioritatea procesului, care este calculată pe baza acestui număr de amicică și pe utilizarea recentă a procesorului).
- Terminal sau TTY: terminalul la care este conectat procesul.
- Numele de utilizator al utilizatorului real și efectiv (RUID și EUID): deținătorul procesului. Deținătorul real este utilizatorul care a introdus comanda, deținătorul efectiv fiind cel care determină accesul la resursele sistemului. RUID și EUID sunt, de regulă, aceleasi, iar procesul are drepturile de acces pe care le are utilizatorul care l-a pornit. Exemplul de mai jos clarifică aceste aspecte: navigatorul **mozilla** din /usr/bin/ este deținut de root:

```
theo:~> ls -l /usr/bin/mozilla
-rwxr-xr-x 1 root      root        4996 Nov 20 18:28 /usr/bin/mozilla*

theo:~> mozilla &
[1] 26595

theo:~> ps -af
UID      PID  PPID C STIME TTY          TIME CMD
theo  26601 26599 0 15:04 pts/5 00:00:00 /usr/lib/mozilla/mozilla-bin
theo  26613 26569 0 15:04 pts/5 00:00:00 ps -af
```

Când utilizatorul *theo* pornește acest program, procesul în sine și toate procesele care sunt pornite de procesul inițial vor fi deținute de utilizatorul *theo* și nu de administratorul sistemului. Când **mozilla** solicită accesul la anumite fișiere, el va fi determinat pe baza permisiunilor pe care le are *theo* și nu pe cele ale administratorului de sistem, *root*.

- Grupul real și efectiv care deține procesul (RGID și EGID): Grupul real care deține procesul este grupul primar al utilizatorului care a pornit procesul. Grupul efectiv este, în aproape toate cazurile, același, exceptând pe cel în care modul de acces SGID a fost aplicat unui fișier.
-

4.1.4. Afisarea informațiilor despre procese

Comanda **ps** este una din uneltele cu care afișați procesele. Această comandă are câteva opțiuni care pot fi combinate pentru afișarea diverselor atribute ale proceselor.

Dacă nu specificați nici o opțiune, **ps** va oferi doar informații despre consola curentă și procesele existente:

```
theo:~> ps
 PID TTY          TIME CMD
 4245 pts/7    00:00:00 bash
 5314 pts/7    00:00:00 ps
```

Întrucât nu oferă informații suficiente – în general, cel puțin câteva sute de procese rulând pe calculatorul dumneavoastră – veți selecta procesele care vă interesează din lista cu toate procesele, folosind comanda **grep** într-o conexiune *pipe*, consultați [Secțiunea 5.1.2.1](#), precum în linia pe care o prezentăm mai jos, comandă care va selecta și afișa toate procesele deținute de un anume utilizator:

```
ps -ef | grep bash nume_de_utilizator
```

Acest exemplu arată toate procesele cu numele bash, consola cea mai întâlnită pe sistemele Linux:

```
theo:> ps auxw | grep bash
brenda  31970  0.0  0.3  6080 1556 tty2   S  Feb23  0:00 -bash
root    32043  0.0  0.3  6112 1600 tty4   S  Feb23  0:00 -bash
theo    32581  0.0  0.3  6384 1864 pts/1   S  Feb23  0:00 bash
theo    32616  0.0  0.3  6396 1896 pts/2   S  Feb23  0:00 bash
theo    32629  0.0  0.3  6380 1856 pts/3   S  Feb23  0:00 bash
theo    2214   0.0  0.3  6412 1944 pts/5   S  16:18  0:02 bash
theo    4245   0.0  0.3  6392 1888 pts/7   S  17:26  0:00 bash
theo    5427   0.0  0.1  3720  548 pts/7   S  19:22  0:00 grep bash
```

În aceste cazuri, comanda **grep**, care cauță linii conținând sirul *bash*, este afișată adesea pe sistemele ale căror resurse nu sunt solicitate. Dacă nu vreți ca acest lucru să se întâmple, folosiți comanda **pgrep**.

Consolele bash reprezintă un caz special: lista proceselor arată, de asemenea, care dintre ele sunt autentificate în consolă (pentru care trebuie să furnizați numele de utilizator și parola, precum autentificarea în modul text sau autentificarea la distanță, diferite de cele care nu sunt autentificate în consolă, pornite, de exemplu, prin apăsarea pictogramei ferestrei terminal). Procesele autentificate în consolă sunt precedate de caracterul „-”.



|?

Vom explica operandul | în capitolul care urmează, consultați [Capitolul 5](#).

Mai multe informații pot fi obținute, ca de obicei, pe căile obișnuite: **ps --help** sau **man ps**. Comanda GNU **ps** poate lucra cu mai multe stiluri de formate ale opțiunilor. Exemplul următoare nu conține erori.

Trebuie să țineți minte că **ps** arată doar procesele active la momentul în care ați introdus această comandă, adică este o fotografie a proceselor la momentul respectiv. Programul **top** afișează informații mai precise, actualizând informațiile obținute cu **ps** (cu multe opțiuni) la fiecare 5 secunde, generând o listă nouă cu procesele care încarcă substanțial sistemul, integrând în același timp mai multe informații despre utilizarea memoriei swap și despre starea procesorului (CPU), din fișierul **proc**:

```
12:40pm up 9 days, 6:00, 4 users, load average: 0.21, 0.11, 0.03
89 processes: 86 sleeping, 3 running, 0 zombie, 0 stopped
CPU states: 2.5% user, 1.7% system, 0.0% nice, 95.6% idle

Mem: 255120K av, 239412K used, 15708K free, 756K shrd, 22620K buff
Swap: 1050176K av, 76428K used, 973748K free, 82756K cached

 PID USER PRI NI SIZE RSS SHARE STAT %CPU %MEM TIME COMMAND
 5005 root 14 0 91572 15M 11580 R 1.9 6.0 7:53 X
19599 jeff 14 0 1024 1024 796 R 1.1 0.4 0:01 top
19100 jeff 9 0 5288 4948 3888 R 0.5 1.9 0:24 gnome-terminal
19328 jeff 9 0 37884 36M 14724 S 0.5 14.8 1:30 mozilla-bin
 1 root 8 0 516 472 464 S 0.0 0.1 0:06 init
 2 root 9 0 0 0 0 SW 0.0 0.0 0:02 keventd
 3 root 9 0 0 0 0 SW 0.0 0.0 0:00 kapm-idled
 4 root 19 19 0 0 0 SWN 0.0 0.0 0:00 ksoftirqd_CPU0
 5 root 9 0 0 0 0 SW 0.0 0.0 0:33 kswapd
 6 root 9 0 0 0 0 SW 0.0 0.0 0:00 kreclaimd
 7 root 9 0 0 0 0 SW 0.0 0.0 0:00 bdflush
 8 root 9 0 0 0 0 SW 0.0 0.0 0:05 kupdated
 9 root -1-20 0 0 0 SW< 0.0 0.0 0:00 mdrecoveryd
13 root 9 0 0 0 0 SW 0.0 0.0 0:01 kjournald
89 root 9 0 0 0 0 SW 0.0 0.0 0:00 khubd
219 root 9 0 0 0 0 SW 0.0 0.0 0:00 kjournald
220 root 9 0 0 0 0 SW 0.0 0.0 0:00 kjournald

12:40pm up 9 days, 6:00, 4 users, load average: 0.21, 0.11, 0.03
89 processes: 86 sleeping, 3 running, 0 zombie, 0 stopped
CPU states: 2.5% user, 1.7% system, 0.0% nice, 95.6% idle
```

Prima linie din listarea generată de comanda **top** conține aceleași informații pe care le puteți obține prin comanda **uptime**:

```
jeff:~> uptime
3:30pm, up 12 days, 23:29, 6 users, load average: 0.01, 0.02, 0.00
```

Datele pentru aceste programe sunt stocate laolaltă cu altele în `/var/run/utmp` (informații despre utilizatorii conectați) și în sistemul de fișiere virtual `/proc`, de exemplu `/proc/loadavg` (informații despre gradul de solicitare a sistemului). Există tot felul de aplicații grafice care arată aceste date, precum Gnome System Monitor și *lavaps*. De la [Fresh Meat](#) și [SourceForge](#) veți putea instala zeci de aplicații care centralizează aceste informații laolaltă cu alte date specifice serverelor și jurnale de la mai multe servere pe un singur server (web), ceea ce vă permite monitorizarea întregii infrastructuri IT de la o singură stație de lucru.

Relațiile dintre procese pot fi afișate prin comanda **pstree**:

```
sophie:~> pstree
init+-amd
|-apmd
|-2*[artsd]
|-atd
|-crond
|-deskguide_apple
|-eth0
|-gdm---gdm+-X
|   `--gnome-session---Gnome
|       |-ssh-agent
|       `--true
|-geyes_applet
|-gkb_applet
|-gnome-name-serv
|-gnome-smproxy
|-gnome-terminal+-bash---vim
|   |-bash
|   |-bash---pstree
|   |-bash---ssh
|   |-bash---mozilla-bin---mozilla-bin---3*[mozilla-bin]
|   `--gnome-pty-helper
|-gpm
|-gweather
|-kapm-idled
|   `--true
|-keventd
|-khubd
|-5*[kjournald]
|-klogd
|-lockd---rpciod
|-lpd
|-mdrecoveryd
|-6*[mingetty]
|-8*[nfsd]
|-nscd---nscd---5*[nscd]
|-ntpd
|-3*[oafd]
|-panel
|-portmap
|-rhnsd
|-rpc.mountd
|-rpc.rquotad
|-rpc.statd
|-sawfish
|-screenshooter_a
|-sendmail
|-sshd---sshd---bash---su---bash
|-syslogd
|-tasklist_applet
|-vmnet-bridge
|-xfs
`--xinetd-ipv6
```

Opțiunile -u și -a oferă informații adiționale. Pentru mai multe opțiuni și ceea ce fac ele, consultați paginile Info.

În secțiunea următoare veți afla cum un proces poate crea altul.

4.1.5. Viața și moartea unui proces

4.1.5.1. Crearea unui proces

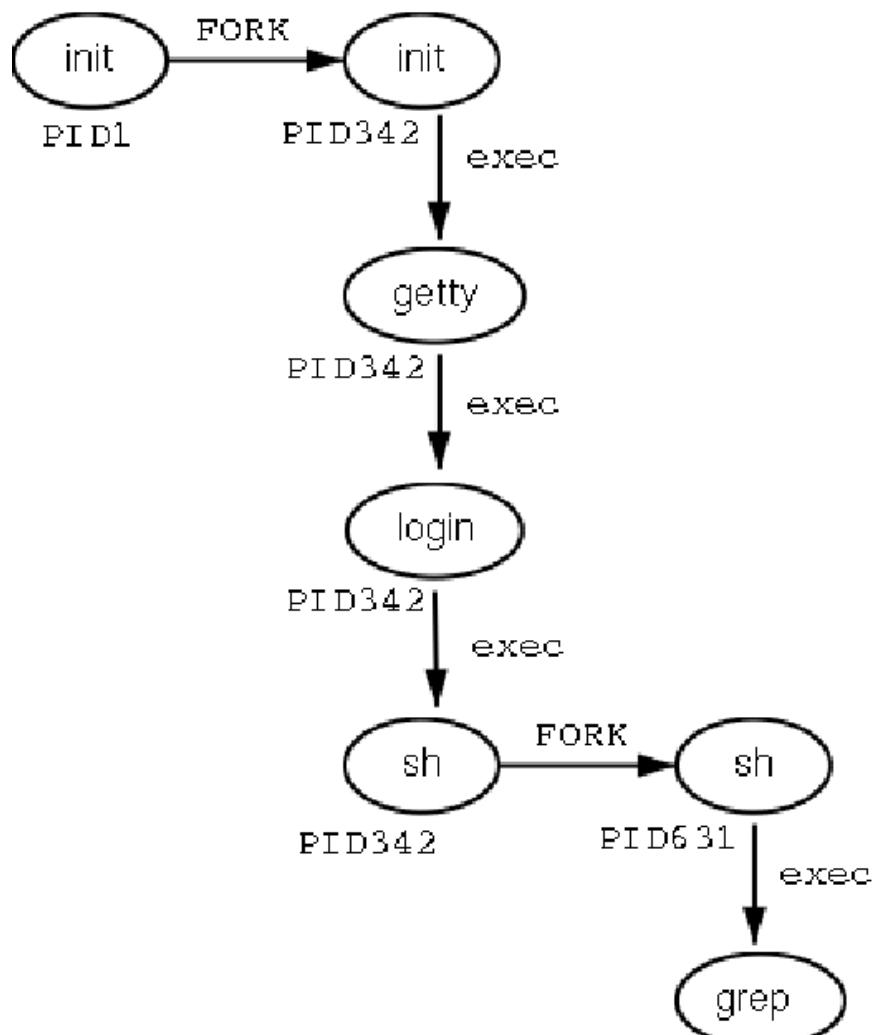
Un proces nou este creat deoarece un proces existent face o copie exactă a lui însuși. Acest proces copil dispune de același mediu ca părintele său, numai numărul ID este diferit. Această procedură este numită *forking (bifurcație)*.

După procesul de bifurcație, spațiul de adresare al procesului copil este suprascris cu datele noului proces. Acest lucru este realizat printr-o apelare exec către sistem.

Mecanismul fork-and-exec schimbă astădat o comandă veche cu una nouă, în timp ce mediul în care este executat noul program rămâne neschimbat, inclusiv configurația dispozitivelor de intrare și ieșire, variabilele de mediu și prioritățile. Acest mecanism este folosit pentru crearea tuturor proceselor UNIX, aplicându-se astfel și sistemelor de operare Linux. Chiar primul proces, **init**, care are numărul ID 1, este bifurcat pe parcursul procedurilor de pornire (boot) în aşa-numita procedură *bootstrapping*.

Schema următoare ilustrează mecanismul fork-and-exec. ID-ul proceselor sunt schimbate după procedura de bifurcație:

Figura 4-1. Mecanismul fork-and-exec



În câteva cazuri, **init** devine părinte al unui proces chiar dacă acest proces copil nu este pornit de **init**, după cum se vede în exemplul **pstree**. Multe programe, de exemplu, transformă (*daemonize*) procesele pornite de ele (procesele-copil) în *servicii*, pentru ca ele să poată rula și după ce procesul-părinte se oprește sau este oprit. Administratorul de ferestre este exemplul tipic; el pornește un proces **xterm** care generează o consolă în care puteți introduce comenzi. Apoi administratorul de ferestre respinge orice responsabilitate ulterioară și pasează procesul copil către **init**. Prin folosirea acestui mecanism, administratorii de ferestre pot fi schimbați fără îintreruperea aplicațiilor în desfășurare.

Câteodată lucrurile pot merge prost, chiar în familiile bune. În cazuri extrem de rare, un proces poate fi stopat înainte ca părintele care l-a pornit să aștepte completarea ciclului normal. Un astfel de proces terminat anormal, neîngropat, se numește *zombie*.

4.1.5.2. Terminarea proceselor

Când un proces se termină normal (nu este omorât, stopat – killed sau îintrerupt pe neașteptate), programul returnează părintelui semnalul de ieșire – *exit status*. Acest semnal este un număr returnat de program care conține informații despre execuția sarcinii. Sistemul prin care se returnează informații despre execuția sarcinilor are originile în limbajul de programare C, în care a fost scris UNIX-ul.

Codurile de ieșire returnate pot fi interpretate de procesele părinte sau în scripturi. Valorile acestor coduri sunt specifice fiecărui program. Aceste informații pot fi găsite de obicei în paginile man ale programelor în cauză, programul **grep** returnând, de exemplu, -1 dacă nu este găsită nici o potrivire, pe baza căruia este afișată linia care conține masajul „No files found” - „Nu am găsit nici un fișier”. Un alt exemplu este cel al comenzii proprii consolei Bash, **true**, care nu face nimic în afară de returnarea semnalului de ieșire 0, al cărui înțeles este încheierea cu succes.

4.1.5.3. Semnale

Procesele se termină deoarece ele primesc un semnal. Sunt multe semnale pe care le puteți trimite unui proces. Folosiți comanda **kill** pentru a trimite semnale proceselor. Comanda **kill -1** arată o listă care conține mai multe semnale. Cele mai multe sunt folosite de sistem pentru uzul intern sau de programatori atunci când scriu coduri. Ca utilizator, veți avea nevoie de următoarele semnale:

Tabelul 4-2. Semnale comune

Numele semnalului	Numărul semnalului	Înțelesul semnalului
SIGTERM	15	Termină procesele într-o manieră ordonată.
SIGINT	2	Întrerupe procesele. Un proces poate să ignore acest semnal.
SIGKILL	9	Întrerupe procesele. Un proces poate să ignore acest semnal.
SIGHUP	1	Pentru servicii: recitește fișierele de configurare.

Puteți citi mai multe despre acțiunile implicate care sunt aplicate unui proces prin trimiterea unui semnal în **man 7 signal**.

4.1.6. SUID și SGID

După cum am promis în capitolul anterior, vom discuta modurile speciale SUID și SGID mai în detaliu. Aceste moduri există pentru a oferi utilizatorului obișnuit posibilitatea de a executa sarcini pe care, în mod normal, nu ar putea să le îndeplinească din cauza sistemului de permisiuni asociate fișierelor pe sistemele asemănătoare cu UNIX-ul. În situația ideală, modurile speciale sunt utilizate cât mai rar, întrucât ele includ riscuri legate de securitatea sistemului. Dezvoltatorii Linux au încercat să le evite pe cât de mult posibil. Versiunea pentru Linux a comenzi **ps**, de exemplu, folosește informația stocată în fișierul **/proc**, care este accesibil tuturor, prin aceasta evitându-se expunerea datelor confidențiale și a resurselor. Înainte de asta și încă valabil pentru sistemele UNIX mai vechi, programul **ps** avea acces la fișiere precum **/dev/mem** și **/dev/kmem**, dezavantajul major fiind legat de permisiunile asociate lor și deținătorii acestor fișiere, după cum se poate observa din exemplul de mai jos:

```
rita:~> ls -l /dev/*mem
crw-r---- 1 root      kmem      1,   2 Aug 30 22:30 /dev/kmem
crw-r---- 1 root      kmem      1,   1 Aug 30 22:30 /dev/mem
```

În cazul versiunilor mai vechi ale comenzi **ps**, nu era posibilă pornirea programului ca utilizator obișnuit, decât dacă se aplicau moduri speciale.

Înțelegem faptul că este bine să evităm aplicarea modurilor speciale dar, câteodată, apare necesitatea aplicării unui SUID. Un exemplu bun este mecanismul schimbării parolelor. Desigur că utilizatorii vor dori să facă ei însăși acest lucru, în locul administratorului de sistem. După cum știm, numele de utilizatori și parolele sunt listate în fișierul **/etc/passwd**, care are asociate următoarele permisiuni și deținători:

```
bea:~> ls -l /etc/passwd
-rw-r--r-- 1 root      root     1267 Jan 16 14:43 /etc/passwd
```

Totuși, utilizatorii trebuie să poată schimba informațiile referitoare la propriul cont, conținute de acest fișier. Puteți obține acest lucru prin aplicarea unor permisiuni speciale programului **passwd**:

```
mia:~> which passwd
passwd is /usr/bin/passwd
```

```
mia:~> ls -l /usr/bin/passwd
-r-s--x--x 1 root      root    13476 Aug  7 06:03 /usr/bin/passwd*
```

Atunci când este apelat, programul **passwd** va rula folosind permisiunile asociate contului de *root*, prin aceasta asigurându-se utilizatorului obișnuit posibilitatea de a edita fișierul care conține parolele, care este deținut de administratorul de sistem.

Modul SGID al unui fișier nu apare la fel de des ca modul SUID, întrucât SGID implică adesea crearea unor grupuri noi. În anumite cazuri, totuși, trebuie să trecem prin aceste inconveniente în ideea de a pune la dispozitie o soluție elegantă (nu vă faceți prea multe griji – grupurile necesare sunt create pe parcursul instalării). Este cazul programelor **write** și **wall**, care sunt folosite pentru a trimite mesaje utilizatorilor altor terminale (ttys). Comanda **write** scrie un mesaj pentru un singur utilizator, în timp ce **wall** scrie tuturor utilizatorilor conectați.

Trimiterea de text către terminalul altui utilizator sau către altă interfață grafică nu este permisă în mod normal. Pentru a trece de această problemă, a fost creat un grup care deține toate dispozitivele terminal. Atunci când se alocă permisiuni SGID comenzilor **write** și **wall**, ele vor rula folosind drepturile de acces care sunt asociate acestui grup care le deține, în exemplul nostru fiind vorba despre *tty*. Deoarece acest grup are drepturi de scriere asupra terminalului de destinație, un utilizator care nu are nici o permisiune asociată cu terminalul de destinație, poate trimite mesaje către acesta.

În exemplul de mai jos, utilizatorul *joe* află în primul rând la care terminal este conectată corespondenta sa, folosind comanda **who**. După aceasta, îi trimit un mesaj folosind comanda **write**. Sunt ilustrate în imagine și drepturile de acces asociate programului **write**, cât și cele asociate terminalului ocupat de utilizatorarea care va primi mesajul lui *joe*: este clar că nimeni în afară de utilizatorul deținător nu are nici un fel de permisiune asupra dispozitivului, cu excepția grupului deținător, care are drepturi de scriere.

```
joe:~> which write
write is /usr/bin/write

joe:~> ls -l /usr/bin/write
-rwxr-sr-x 1 root      tty      8744 Dec  5 00:55 /usr/bin/write*

joe:~> who
jenny      ttym1      Jan 23 11:41
jenny      pts/1      Jan 23 12:21 (:0)
jenny      pts/2      Jan 23 12:22 (:0)
jenny      pts/3      Jan 23 12:22 (:0)
joe       pts/0      Jan 20 10:13 (lo.callhost.org)

joe:~> ls -l /dev/ttym1
crw--w---- 1 jenny  ttym  4,      1 Jan 23 11:41 /dev/ttym1
```

```
joe:~> write jenny tty1
hey Jenny, shall we have lunch together?
^C
```

Utilizatoarea *jenny* primește pe ecranul ei următoarele:

```
Message from joe@lo.callhost.org on ptys/1 at 12:36 ...
hey Jenny, shall we have lunch together?
EOF
```

După primirea unui mesaj, terminalul poate fi eliberat prin combinația de taste **Ctrl+L**. Pentru a nu primi deloc mesaje (cu excepția celor din partea administratorului de sistem), folosiți comanda **mesg**. Pentru a consulta lista cu utilizatorii care acceptă mesaje, folosiți **who -w**. Toate aspectele fiecărei comenzi sunt explicate în amănunt în paginile Info ale comenzi respective.



Numele grupurilor pot fi altele

Schema grupurilor este specifică fiecărei distribuții. Alte distribuții pot folosi alte nume sau alte soluții.

4.2. Procesul de pornire (boot), inițiere (init) și oprire (shut-down)

4.2.1. Introducere

Un aspect aparte al Linux-ului este metoda sa deschisă prin care pornește și oprește sistemul de operare. La pornire sunt încărcate programe specifice, folosind configurația particulară a acestora, permitându-vă să schimbați această configurație pentru a controla procesul de pornire, închiderea fiind făcută, de asemenea, într-o manieră organizată și elegantă.

În afară de aspectele legate de controlul proceselor de pornire și oprire, natura Linux-ului face mai ușoară determinarea exactă a surselor celor mai multor probleme asociate cu pornirea sau oprirea sistemului. O înțelegere de bază a acestor procese este benefică pentru oricine care utilizează un sistem Linux.

Multe sisteme Linux folosesc **lilo**, Linux Loader (încărătorul de sisteme al Linux-ului), pentru pornirea sistemului. Noi vom discuta doar despre GRUB, totuși, care este mai ușor de folosit și mai flexibil (*în ultimul timp, tendința de a folosi GRUB-ul câștigă teren în cazul majorității distribuțiilor de Linux^{n.t.}*). Dacă aveți nevoie de informații despre **lilo**, consultați paginile man și HOWTO-urile. Ambele sisteme pot gestiona instalări dual-boot (mai multe sisteme de operare pe același calculator^{n.t.}); ne vom referi la HOWTO-uri pe marginea acestui subiect pentru exemple practice și informații de bază.

4.2.2. Procesul de pornire (boot)

Când este pornit un calculator x86, procesorul caută în zona de la sfârșitul memoriei de sistem BIOS-ul (Basic Input/Output System) și îl rulează. Programul Bios este scris în memoria permanentă și este mereu gata de utilizare. BIOS-ul asigură interfața primară cu dispozitivele periferice și controlează primul pas al procesului de pornire.

BIOS-ul testează sistemul, caută și verifică perifericele și apoi caută un disc pentru a-l folosi la pornirea sistemului. De regulă caută prezența mediilor bootabile în unitatea de dischetă (sau în unitatea CD-ROM la sistemele mai noi), după care caută pe discul fix. Ordinea dispozitivelor folosite la pornire este controlată, de obicei, de un submeniu al BIOS-ului. Odată instalat Linux-ul pe discul fix al sistemului, BIOS-ul caută Master Boot Record (MBR-ul) care se află în primul sector al primului disc fix, îl încarcă în memorie și pasează controlul procesului acestuia din urmă.

MBR-ul conține instrucțiuni despre încărcarea GRUB-ului (sau a LILO), folosind un sistem de operare prestabilit. MBR-ul încarcă apoi încărătorul de sisteme, care va prelua procesul (dacă încărătorul de sisteme este instalat în MBR). În configurația implicită din distribuția Linux RedHat (și Ubuntu^{n.t.}), GRUB-ul folosește setările din MBR pentru afișarea opțiunilor de pornire într-un meniu. Când GRUB-ul a primit instrucțiunile corecte de pornire ale sistemului de operare ales, ori din linia de comandă, ori dintr-un fișier de configurare, găsește fișierele necesare pentru pornire și predă controlul mașinii aceluia sistem de operare.

4.2.3. Facilitățile GRUB-ului

Această metodă de pornire se numește încărcare directă (direct loading) din cauză că instrucțiunile sunt folosite pentru încărcarea nemijlocită a sistemului de operare, fără cod intermedier între încărcătorul de sisteme și fișierele principale ale sistemului de operare (cum ar fi kernelul). Procesul de pornire existent în cazul altor sisteme de operare diferă foarte puțin de cel pe care vi l-am prezentat mai sus. De exemplu, sistemele de operare de la Microsoft, DOS și Windows, rescriu complet MBR-ul la instalare, fără a încorpora nimic din configurația existentă pe MBR. Acest lucru distrugе orice informație stocată în MBR de alte sisteme de operare, precum Linux. Sistemele de operare Microsoft, ca și alte sisteme proprietare, sunt încărcate folosind o metodă de încărcare în lanț. Prin această metodă, MBR-ul țintește primul sector al partiiiei pe care se află sistemul de operare, unde găsește fișierele necesare pentru pornirea sistemului.

GRUB-ul poate gestiona ambele metode de pornire, permitându-vă să folosiți aproape orice sistem de operare, cele mai folosite sisteme de fișiere și aproape orice disc fix recunoscut de BIOS.

GRUB-ul are multe alte caracteristici; cele mai importante sunt:

- GRUB-ul asigură un mediu pre-sistem de operare, bazat pe linia de comandă, pentru mașinile x86, care permite flexibilitate maximă în procesul de încărcare al sistemelor de operare, având posibilitatea de a folosi anumite opțiuni și de a aduna informații despre sistem.
- GRUB-ul poate gestiona modul Logical Block Addressing (LBA), care este necesar pentru accesarea multor discuri fixe IDE și a tuturor discurilor fixe SCSI. Înainte de LBA, discurile fixe aveau o limită de 1024 de cilindri, impusă de faptul că peste aceasta BIOS-ul nu putea găsi un fișier.
- Fișierul de configurare al GRUB-ului este citit de pe disc de fiecare dată când sistemul este pornit, preîntâmpinând rescrierea MBR-ului de fiecare dată când modificați opțiunile de pornire.

Descrierea completă a GRUB-ului poate fi găsită prin introducerea comenzi info grub sau pe [saitul GRUB](#). Proiectul de documentare pentru Linux are un [Mini-CUM SĂ realizezi o instalare dual-boot cu ajutorul GRUB-ului](#).

4.2.4. Inițierea (Init)

Kernelul, odată încărcat, caută fișierul **init** din `sbin` și îl execută.

Când **init** pornește, el devine părintele sau bunicul tuturor proceselor care pornesc automat pe sistemul dumneavoastră Linux. Primul lucru pe care îl face **init** este să citească fișierul de inițializare, `/etc/inittab`. Acest fișier de inițializare dă instrucțiuni programului **init** să citească un script care conține o configurație inițială a mediului, prin care sunt stabilite traseele, se pornește memoria swap, se verifică sistemul de fișiere, și așa mai departe. Pe scurt, acest pas gestionează tot ceea ce trebuie făcut la inițializarea sistemului: potrivirea ceasului, pornirea porturilor seriale și altele.

Apoi **init** continuă să citească fișierul `/etc/inittab`, care descrie cum trebuie setat sistemul în fiecare nivel de rulare și stabilește *nivelul de rulare (default runlevel)* implicit. Nivelul de rulare este configurația unui proces. Toate sistemele asemănătoare UNIX-ului pot rula în diverse configurații ale proceselor, cum ar fi modul single user (un singur utilizator), care este cunoscut ca runlevel 1 sau runlevel S (ori s). În acest mod, doar administratorul de sistem se poate autentifica la sistem. Este folosit la ducerea la bun sfârșit a sarcinilor de întreținere, pentru a elimina alterarea datelor sistemului sau ale utilizatorilor. Normal, în acest mod nu se permite accesul utilizatorilor obișnuiți, așa că serviciile disponibile acestora sunt dezactivate. Alt nivel de rulare este cel de repornire (reboot runlevel) sau runlevel 6, care oprește conform procedurilor corecte toate serviciile care rulează și restartează sistemul.

Folosiți comanda **who** pentru a afișa nivelul de rulare curent:

```
willy@ubuntu:~$ who -r
      run-level 2 2006-10-17 23:22          last=S
```

Mai multe despre nivelurile de rulare vom vorbi în secțiunea următoare, [Secțiunea 4.2.5](#).

După ce determină nivelul de rulare implicit al sistemului dumneavoastră, **init** pornește toate procesele necesare sistemului, care rulează în fundal, prin citirea directorului `rc` potrivit nivelului de rulare desemnat. **init** rulează fiecare script de oprire (numele fișierelor acestora încep cu un K) cu un parametru de stopare. Apoi rulează toate scripturile de pornire (numele acestora încep cu un S) în directorul potrivit

nivelului de rulare, astfel încât toate serviciile și aplicațiile sunt pornite corect. Puteti executa aceleasi scripturi manual după ce sistemul termină procedurile de pornire printr-o comandă ca **/etc/init.d/httpd stop** sau **service httpd stop**, autentificați ca root, comandă referitoare, în acest caz, la oprirea serverului web.



Cazuri speciale

La pornirea sistemului, sunt executate, în mod normal, scripturile aflate în `rc2.d` și `rc3.d`. În acest caz, nu este oprit nici un serviciu (cel puțin, nu permanent). Există doar servicii care sunt pornite.

Nici unul din scripturile care pornesc și opresc de fapt serviciile nu se află în `/etc/rc<x>.d`. Mai degrabă toate fișierele din directorul `/etc/rc<x>.d` sunt legături simbolice care țințesc către scripturile propriu-zise, aflate în `/etc/init.d`. O legătură simbolică nu este altceva decât un fișier care țințește alt fișier și este folosită în acest caz deoarece poate fi creată și stearsă fără a modifica scriptul în sine care oprește sau pornește un serviciu. Legăturile simbolice către scripturi diferite sunt numerotate într-o ordine prestabilită, pentru ca acestea să fie pornite în acea ordine.

Puteti schimba ordinea în care sunt pornite sau oprite serviciile prin redenumirea legăturii simbolice care se referă la scriptul care controlează serviciul. Puteti folosi același număr de mai multe ori dacă doriti ca un anume serviciu să fie pornit sau oprit chiar înaintea sau după alt serviciu, precum în exemplul de mai jos, care listează conținutul directorului `/etc/rc5.d`, în care **cron** și **xfs** sunt pornite amândouă de o legătură al cărei nume începe cu „S90”. În acest caz, scripturile sunt pornite după ordinea alfabetică.

```
[jean@blub /etc/rc5.d] ls
K15httpd@   K45named@   S08ipchains@  S25netfs@    S85gpm@
K16rarpd@   K46radvd@   S08iptables@  S26apmd@    S90crond@
K20nfs@     K61ldap@   S09isdn@      S28autofs@   S90xfs@
K20rstatd@  K65identd@  S10network@   S30nscd@    S95anacron@
K20rusersd@ K74ntpd@   S12syslog@   S55sshd@    S95atd@
K20rwalld@  K74ypserv@  S13portmap@  S56rawdevices@ S97rhnsd@
K20rwhod@   K74ypxfrd@  S14nfsllock@ S56xinetd@  S99local@
K25squid@   K89bcm5820@ S17keytable@ S60lpd@     S80sendmail@
K34yppasswdd@ S05kudzu@  S20random@   S80sendmail@
```

După ce **init** a trecut prin nivelurile de rulare pentru a ajunge în cel implicit, scriptul `/etc/inittab` bifurcă un proces **getty** pentru fiecare consolă virtuală (prompterul de autentificare din modul text). **getty** deschide liniile `tty`, stabilește modul acestora, afișează prompterul de autentificare, obține numele de utilizator și apoi inițiază procesul de autentificare pentru acel utilizator. Astfel se permite autentificarea utilizatorilor la sistem și folosirea sistemului de către aceștia. Implicit, cele mai multe sisteme oferă 6 console virtuale, dar acest comportament este configurabil, după cum puteti afla din fișierul `inittab`.

`/etc/inittab` poate spune, de asemenea, programului **init** cum să gestioneze un utilizator care apasă taste **Ctrl+Alt+Del** în consolă. Pentru ca un sistem să fie oprit și restartat corespunzător, în loc să alegeti dintre modurile puse la dispoziție prin apăsarea pictogramei **îșișire**, îi puteți transmite lui **init** să execute comanda **/sbin/shutdown -t3 -r now**, de exemplu, atunci când apăsați tastele menționate. În plus, `/etc/inittab` determină comportamentul lui `init` în cazul întreruperii alimentării, dacă sistemul dumneavoastră are atașată o unitate UPS.

Pe cele mai multe sisteme bazate pe RPM, ecranul de autentificare în modul grafic este pornit în nivelul de rulare 5, în care `/etc/inittab` rulează un script numit `/etc/X11/prefdm`. Scriptul `prefdm` rulează administratorul grafic preferat, pe baza conținutului directorului `/etc/sysconfig/desktop`. Aceasta este de regulă **gdm** dacă folosiți GNOME sau **kdm** în cazul în care folosiți KDE, dar ele pot fi amestecate, existând chiar și **xdm**, care vine odată cu instalarea standard a serverului X.

Dar mai sunt și alte posibilități. În Debian, de exemplu, există câte un script de inițializare pentru fiecare administrator grafic și este folosit conținutul directorului `/etc/X11/default-displaymanager` pentru a determina care dintre acestea este folosit. Mai multe lucruri despre interfață grafică pot fi citite în [Secțiunea 7.3](#). Mai mult, documentația proprie sistemului dumneavoastră explică în detaliu aspectele legate de **init**.

Directoarele `/etc/default` și/sau `/etc/sysconfig` conțin intrări pentru mai multe funcții și servicii care sunt citite la pornirea sistemului. Locul în care se află directorul care conține setările implicate ale sistemului poate fi întrucâtva diferit, el depinzând de distribuția de Linux aleasă.

În afară de interfața grafică aleasă, multe alte servicii pot fi pornite. Dar dacă totul merge aşa cum trebuie, dumneavoastră veți avea în față prompterul de autentificare sau ecranul de autentificare la sfârșitul procesului de pornire.



Alte proceduri

Am explicat în ce mod SysV **init** lucrează pe calculatoarele compatibile x86. Procedurile de pornire pot fi diferite în cazul altor arhitecturi sau distribuții. Alte sisteme pot folosi stilul **init** BSD, în care fișierele de pornire nu sunt împărțite în mai multe directoare `/etc/rc<LEVEL>.d`. Este posibil ca sistemul dumneavoastră să folosească directorul `/etc/rc.d/init.d` în locul directorului `/etc/init.d`.

4.2.5. Nivelurile de rulare init

Ideea de a opera servicii diferite pe niveluri de rulare diferite rezidă, în principiu, din faptul că sisteme diferite pot fi folosite în moduri diferite. Anumite servicii nu pot fi folosite decât atunci când sistemul se află într-o stare particulară, sau *mod* (*mode*), de exemplu atunci când este pregătit pentru mai mulți utilizatori sau când este disponibilă legătura la Internet.

Există, de asemenea, momente în care doriți să folosiți sistemul într-un mod care solicită mai puțin resursele acestuia. Exemple sunt destule, precum repararea discurilor fixe în nivelul de rulare 1, pentru ca nici un alt utilizator să nu fie autentificat în sistem, sau punerea unui server în nivelul de rulare 3 fără ca să ruleze vreo sesiune a serverului X (modul grafic). În aceste cazuri, serviciile care depind de un mod superior al sistemului nu sunt rulate, deoarece nu are nici un sens, ele nefuncționând corect pe un nivel inferior de rulare. Prin faptul că fiecare serviciu este desemnat deja să pornească în momentul în care nivelul său de rulare este atins, obțineți un proces de pornire ordonat și puteți schimba repede modul în care se află calculatorul dumneavoastră fără să vă gândiți la ce servicii trebuie să porniți sau să opriți manual.

Nivelurile de rulare disponibile sunt descrise în general în `/etc/inittab`, un caz particular fiind prezentat mai jos:

```
#  
# inittab This file describes how the INIT process should set up  
#           the system in a certain run-level.  
  
# Default run level. The run levels are:  
#   0 - halt (Do NOT set initdefault to this)  
#   1 - Single user mode  
#   2 - Multiuser, without NFS  
#       (The same as 3, if you do not have networking)  
#   3 - Full multiuser mode  
#   4 - unused  
#   5 - X11  
#   6 - reboot (Do NOT set initdefault to this)  
#  
id:5:initdefault:  
<--cut-->
```

Puteți configura nivelurile de rulare nefolosite (în mod obișnuit este cazul nivelului 4) după cum doriti. Mulți utilizatori configurează aceste niveluri în moduri care li se potrivesc mai mult, lăsând nivelurile de rulare standard neschimbate. Acest lucru le permite să se mute în și din configurația lor particularizată fără să modifice aranjamentul implicit al nivelurilor de rulare standard.

Dacă sistemul dumneavoastră ajunge într-o stare în care nu poate porni datorită unui fișier `/etc/inittab` necorespunzător sau dacă nu vă puteți autentifica din cauză că fișierul `/etc/passwd` este distrus (sau dacă v-ați uitat pur și simplu parola), porniți sistemul în modul corespunzător unui singur utilizator.



Fără grafică?

Atunci când lucrați în modul text din cauză că nu vi s-a prezentat un ecran grafic de autentificare în consola calculatorului dumneavoastră, vă puteți muta în consola 7 sau mai sus pentru a obține un ecran grafic de autentificare. Dacă nu vă aflați în acest caz, verificați nivelul de rulare curent prin comanda `who -r`. Dacă este stabilit pe altceva decât cel desemnat implicit în fișierul `/etc/inittab`, există posibilitatea ca sistemul să nu pornească implicit în modul grafic. Luăți legătura cu administratorul de sistem sau citiți

paginile **man init**. Schimbarea nivelurilor de rulare este bine să fie făcută folosind comanda **telinit**; trecerea din modul text într-o consolă grafică sau invers nu implică schimbarea nivelului de rulare.

Discutarea nivelurilor de rulare, scripturilor și configurațiilor făcută în acest ghid este una care se dorește cât mai generală. Există multe variații. De exemplu, Gentoo Linux păstrează scripturile în `/etc/runlevels`. Alte sisteme pot trece mai întâi prin unul sau mai multe niveluri de rulare pentru a executa toate scripturile de acolo, înainte de a ajunge la nivelul de rulare final și a executa scripturile conținute de acesta. Apelați la documentația sistemului propriu pentru mai multe informații. Puteți, de asemenea, să citiți scripturile la care face referire `/etc/inittab` pentru a înțelege mai bine ceea ce se întâmplă cu sistemul dumneavoastră.

4.2.5.1. Uneltele

Utilitățile **chkconfig** și **update-rc.d**, când sunt instalate în sistemul dumneavoastră, asigură o unealtă simplă în linie de comandă, care este folosită la verificarea ierarhiei directorului `/etc/init.d`. Ele scutesc pe administratorul de sistem de la gestionarea manuală a numeroaselor legături simbolice care se află în subdirectoarele directoarelor `/etc/rc[x].d`.

În plus, anumite sisteme oferă unealtă **ntsysv**, care asigură o interfață în modul text; veți afla că aceasta este mai ușor de folosit decât interfața stil linie de comandă a **chkconfig**. În SuSE Linux, veți găsi uneltele **yast** și **insserv**. Pentru configurația adresată utilizatorilor obișnuiți, puteți încerca DarkConf, care, printre alte facilități, oferă posibilitatea schimbării nivelurilor de rulare între 3 și 5. În Mandriva, această utilitate devine Mandriva Linux Control Center.

Cele mai multe distribuții oferă o interfață grafică pentru configurarea proceselor, verificați aşadar documentația sistemului dumneavoastră.

Toate aceste utilități trebuie rulate ca administrator. Tot administratorul de sistem poate crea manual legătura potrivită în orice director de pe orice nivel de rulare, pentru a porni sau opri un serviciu pe un nivel de rulare anume.

4.2.6. Oprirea

UNIX-ul nu a fost proiectat să fie închis, dar dacă trebuie neapărat, folosiți comanda **shutdown**. După completarea procedurilor de închidere, opțiunea `-h` va opri sistemul, în timp ce `-r` îl va reporni.

Comenzile **reboot** și **halt** sunt disponibile pentru a apela comanda **shutdown** atunci când sistemul se află în nivelurile de rulare de la 1 la 5, asigurându-se prin ele oprirea corectă a sistemului, dar vă faceți un obicei prost, întrucât nu toate versiunile de UNIX/Linux dispun de aceste facilități.

Dacă sistemul dumneavoastră nu se oprește definitiv, nu trebuie să-l opriți decât după ce primiți un mesaj prin care vi se spune că sistemul este oprit sau că a terminat procedurile de oprire, pentru a da timp sistemului să demonteze partitiile. Dacă sunteți nerăbdător, puteți cauza pierderi de date.

4.3. Gestionarea proceselor

4.3.1. Muncă pentru administratorul de sistem

Cu toate că gestionarea resurselor sistemului, inclusiv a proceselor, este o sarcină care cade în grija administratorului de sistem, este bine ca utilizatorul obișnuit să aibă măcar o idee despre acest subiect, în special despre procesele proprii și despre condițiile optime de executare ale acestora.

Vom explica câteva aspecte din punct de vedere teoretic despre performanțele sistemului, dar nu vom atinge subiecte legate de optimizarea dispozitivelor sau alte proceduri avansate. În schimb, vom studia problemele zilnice cu care se confruntă utilizatorii obișnuiți și ceea ce pot face aceștia pentru folosirea optimă a resurselor disponibile. După cum veți învăța în secțiunea următoare, acest lucru este, în principal, o abordare de genul „gândește înainte de a acționa”.

Figura 4-2. Nu poți merge mai repede?



4.3.2. Cât durează?

Consola oferă o comandă proprie, **time**, care afișează timpul necesar pentru executarea unei comenzi. Răspunsul este foarte precis și o puteți folosi asupra oricărei comenzi. În exemplul care urmează este necesar un minut și jumătate pentru a face acea carte.

```
tilly:~/xml/src> time make
Output written on abook.pdf (222 pages, 1619861 bytes).
Transcript written on abook.log.

real    1m41.056s
user    1m31.190s
sys     0m1.880s
```

Comanda **time** din GNU care se află în /usr/bin (opusă versiunii proprii consolei) afișează mai multe informații care pot fi dispuse în mai multe moduri. Arată și semnalul de ieșire al comenzi, precum și timpul total care a fost necesar. Aceeași comandă dar folosind versiunea independentă **time** va genera

următoarele:

```
tilly:~/xml/src> /usr/bin/time make
Output written on abook.pdf (222 pages, 1595027 bytes).
Transcript written on abook.log.

Command exited with non-zero status 2
88.87user 1.74system 1:36.21elapsed 94%CPU
          (0avgtext+0avgdata 0maxresident)k
0inputs+0outputs (2192major+30002minor)pagefaults 0swaps
```

Pentru toate informațiile, apelați la paginile Info.

4.3.3. Performanță

Pentru utilizator, performanța înseamnă executarea rapidă a unei comenzi. Pentru un gestionar de sistem, pe de altă parte, înseamnă mult mai mult: administratorul de sistem trebuie să optimizeze performanțele sistemului pentru acel sistem ca întreg, inclusiv utilizatori, toate programele și serviciile. Performanțele sistemului pot depinde de mii de lucruri mărunte, care nu sunt contabilizate de comanda **time**:

- programul care este executat este scris prost sau nu folosește corect calculatorul
- accesul la discuri, controlere, monitor, alte dispozitive, etc.
- accesibilitatea sistemelor la distanță (performanțele rețelei)
- numărul utilizatorilor din sistem, numărul celor care lucrează simultan
- perioada efectivă din zi
- ...

4.3.4. Încărcarea (nivelul de solicitare a) sistemului

Pe scurt: încărcarea depinde de ceea ce este normal pentru sistemul dumneavoastră. Vechiul meu P133 care rulează ca firewall, server SSH, server de fișiere, router, server de mail, server proxy și care asigură alte câteva servicii, nu se plângе cu 7 utilizatori autentificați; nivelul de solicitare este tot 0, în medie. Anumite sisteme (multi-procesor) pe care le-am văzut suportau destul de bine un nivel de solicitare de 67. Puteți afla într-un singur mod – verificați periodic nivelul de solicitare, pentru a ști care este cel normal. Dacă nu o faceți, puteți măsura nivelul de solicitare a sistemului după răspunsul liniei de comandă, a cărei viteză este influențată de sute de factori.

Nu uitați că sisteme diferite se comportă aparte în cazul același nivel de solicitare. De exemplu, un sistem care are o placă grafică care asigură accelerare nu va avea probleme la randarea imaginilor 3D, în timp ce același sistem cu o placă VGA ieftină va fi încetinit substanțial la randare. Vechiul meu P133 va reacționa greoi la pornirea serverului X (modul grafic), în timp ce pe sistemele moderne veți observa cu greu diferențe în nivelul de solicitare a sistemului.

4.3.5. Ca utilizator, pot face ceva?

Un mediu stulos vă poate încetini. Dacă aveți stabilite multe variabile de mediu (în schimbul variabilelor consolei), traseele mai lungi de căutare care nu sunt optimizate (erori în stabilirea variabilei de mediu PATH) și alte asemenea reglaje făcute „din zbor”, sistemul va avea nevoie de mai mult timp pentru a căuta și să citească datele.

În X, administratorii de ferestre și mediul spațiului de lucru pot mânca mult din puterea procesorului. Un spațiu de lucru mai atractiv are un preț, chiar dacă îl puteți descărca gratuit, deoarece cele mai multe dintre ele pot integra tot felul de suplimente (add-ons). Modestia este o virtute atunci când nu sunteți dispusi să cumpărați un calculator nou în fiecare an.

4.3.5.1. Priorități

Prioritatea sau importanța unui serviciu este determinată de numărul său *amical*. Un program cu un număr amical mare este prietenos față de alte programe, alți utilizatori și față de sistem; nu este o sarcină importantă. Cu cât numărul amical este mai mic, cu atât este mai importantă sarcina și vor fi folosite mai multe resurse fără a le împărtăji cu alte programe.

Mărirea numărului amical al programelor pentru a le face mai prietenoase este folositoare pentru procesele care solicită mult procesorul (compilatoare, aplicații matematice și altele de acest fel). Procesele care folosesc intensiv timpii I/O sunt răsplătite automat de către sistem prin acordarea de prioritate (li se asociază un număr amical mai mic), un exemplu fiind cel al tastaturii, care are cea mai ridicată prioritate într-un sistem.

Definirea priorității unui program se face prin comanda **nice**.

Cele mai multe sisteme asigură și comanda BSD **renice**, care vă permite să schimbați *gradul de amicizia* (*niceness*) a unei comenzi care se află în execuție. Vă îndrumăm din nou către paginile de documentație specifică sistemului dumneavoastră.



Programe interactive

Nu vă sfătuim să schimbați prin **nice** sau **renice** numărul amical al programelor sau sarcinilor care rulează în prim-plan. Folosirea acestor comenzi este rezervată administratorului de sistem. Citiți paginile de manual sau cele Info despre funcțiile disponibile administratorului de sistem.

4.3.5.2. Resursele procesorului (CPU)

Pe orice sistem Linux, multe programe doresc să folosească procesorul în același timp, chiar dacă dumneavoastră sunteți singurul utilizator autentificat în sistem. Fiecare program are nevoie de un anumit număr de cicluri ale procesorului pentru a rula. Pot exista momente în care nu sunt disponibile suficiente cicluri deoarece procesorul este ocupat. Comanda **uptime** este mai degrabă orientativă (ea afișează doar media, pe când dumneavoastră aveți nevoie să cunoașteți situația normală), dar departe de a fi nefolositoare. Sunt câteva acțiuni pe care le puteți întreprinde dacă dumneavoastră credeți că procesorul este vinovat de faptul că sistemul nu vă răspunde la comenzi aşa cum ar trebui:

- Rulați programele care încarcă sistemul atunci când sistemul este mai puțin solicitat. Momentele în care sistemul este mai puțin solicitat se întâlnesc, de regulă, noaptea. Consultați secțiunea următoare, care prezintă programarea sarcinilor.
- Evitați ca sistemul să execute sarcini care nu sunt necesare: opriți serviciile și programele pe care nu le folosiți, folosiți comanda **locate** în loc de mai solicitanta **find**,...
- Rulați sarcinile mari consumatoare de resurse cu o prioritate scăzută.

Dacă nici una din aceste soluții nu este o opțiune viabilă în cazul sistemului dumneavoastră, ar trebui să luați în considerare schimbarea procesorului. Pe o mașină UNIX, acest lucru este o sarcină care cade în grijă administratorului de sistem.

4.3.5.3. Resursele de memorie

În situația în care procesele aflate în desfășurare au nevoie de mai multă memorie decât dispune fizic sistemul, un sistem Linux nu va îngheța; va începe să pageze, sau să facă ceea ce se numește *swapping*, adică procesul va folosi memorie de pe discul fix sau din spațiul swap, mutând conținutul memoriei fizice (fragmente din programul rulat sau tot programul în cazul swapping-ului) pe disc, permitând în acest mod memoriei fizice să gestioneze alte procese. În acest mod se încetinește destul de mult sistemul, deoarece accesul la disc este mai lent decât accesul la memorie. Comanda **top** poate fi folosită pentru afișarea memoriei folosite, fizice cât și swap. Sistemele care folosesc glibc oferă pentru afișarea memoriei în uz comenziile **memusage** și **memusagetest**.

Dacă aveți convingerea că este folosită multă memorie fizică și swap, puteți încerca:

- Terminarea, stoparea sau mărirea numărului amical al programelor care folosesc multă memorie.
- Adăugarea de memorie sistemului (în unele cazuri, mărirea spațiului swap).
- Îmbunătățirea performanțelor sistemului, lucru care nu face obiectul acestui ghid. Citiți lista din

4.3.5.4. Resursele I/O

În timp ce limitările impuse de I/O sunt cauza principală de stres pentru administratorii de sistem, Linux-ul nu oferă utilități foarte precise pentru măsurarea performanțelor I/O. Programele **ps**, **vmstat** și **top** dau anumite indicații despre cât de multe programe solicită resursele I/O; **netstat** afișează statistică ale interfeței de rețea, dar practic nu există unele care să măsoare răspunsul I/O la încărcarea sistemului, iar comanda **iostat** oferă doar o scurtă descriere a folosirii I/O în general. Există numeroase interfețe grafice care organizează datele afișate de aceste programe într-o formă inteligibilă pentru dumneavoastră.

Fiecare dispozitiv are problemele sale, dar lățimea de bandă disponibilă interfețelor de rețea și lățimea de bandă disponibilă discurilor sunt principalele cauze ale gâtuirilor care afectează performanțele I/O.

Problemele de rețea ale I/O:

- Supraîncărcarea rețelei: cantitatea de date transportată pe rețea este mai mare decât capacitatea rețelei, lucru care duce la executarea cu încetineală a sarcinilor care implică rețea, pentru toți utilizatorii. Pot fi rezolvate prin reducerea solicitării rețelei (care înseamnă, în principal, dezactivarea protocolelor și serviciilor de care nu aveți nevoie) sau prin reconfigurarea rețelei (de exemplu, folosirea rețelelor intranet, înlocuirea hub-urilor cu switch-uri, actualizarea programelor și modernizarea echipamentelor).
- Problemele cauzate de integritatea rețelei: apar atunci când datele sunt transferate incorect. Rezolvarea acestor probleme se face doar prin identificarea elementului defect și înlocuirea acestuia.

Problemele I/O cauzate de disc:

- rata de transfer per proces prea mică: viteza de citire sau scriere per proces este prea mică.
- rata de transfer globală prea mică: lățimea de bandă totală pe care o pune la dispoziție sistemul tuturor programelor care rulează la un moment dat este prea mică.

Acest fel de probleme sunt mai greu de depistat și au nevoie de echipamente suplimentare care să redistribuie fluxul de date pe magistrale, controller-e și discuri, dacă se confirmă că echipamentele suprasolicită sunt cauza problemelor. O soluție ar fi o configurație RAID, optimizată pentru acțiunile de intrare și ieșire. Astfel, vă puteți păstra același echipament. O trecere la magistrale, controller-e și discuri mai rapide este cealaltă cale prin care puteți rezolva problemele de acest gen.

Dacă supraîncărcarea nu este cauza problemelor, este posibil ca echipamentele dumneavoastră să nu mai funcționeze cum trebuie sau să nu fie conectate corect la sistem. Puteți să începeți cu verificarea contactelor, conectorilor și cuplajelor.

4.3.5.5. Utilizatorii

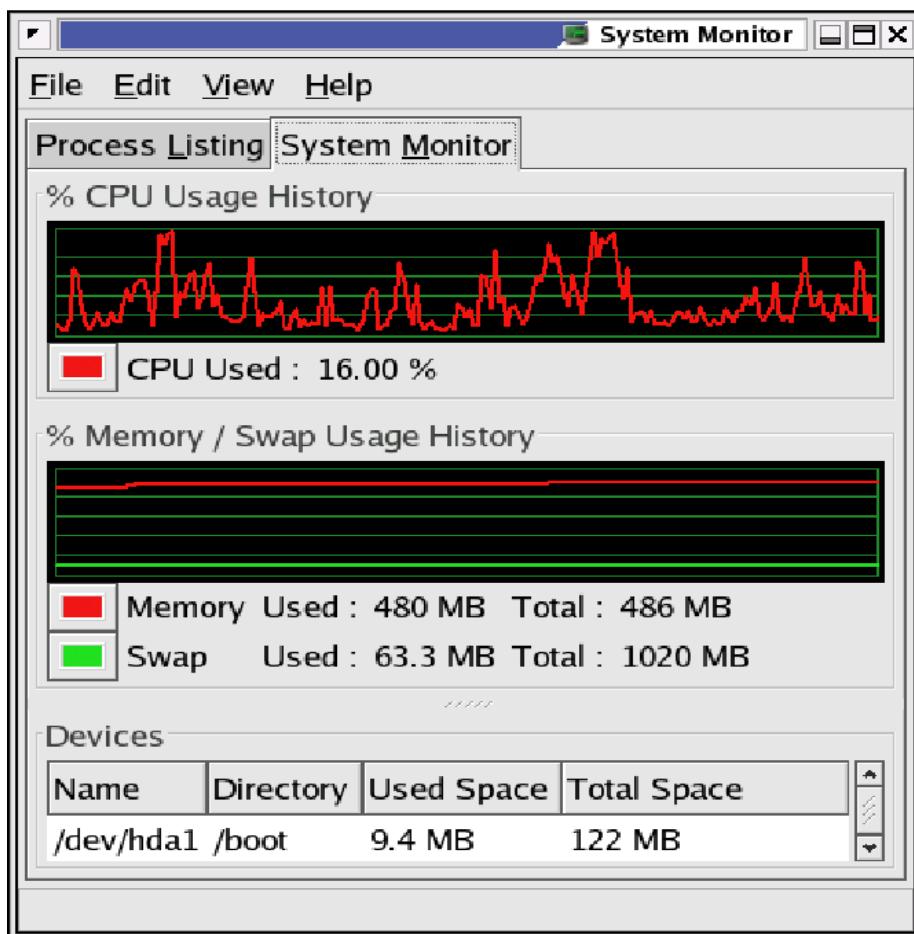
Utilizatorii pot fi împărțiți în câteva categorii, după comportamentul lor față de folosirea resurselor:

- Utilizatori care rulează un număr (mare) de sarcini care nu solicită foarte mult sistemul: dumneavoastră, utilizatorul începător de Linux, de exemplu.
- Utilizatori care rulează doar câteva sarcini, dar care solicită sistemul: cei care rulează simulări, calcule, emulatoare sau alte programe care au nevoie de multă memorie, acești utilizatori folosind, de regulă, și fișiere de date foarte mari.
- Utilizatori care rulează puține sarcini, dar folosesc foarte multe resurse ale procesorului (dezvoltatorii și cei cu preocupări asemănătoare).

Veți observa că cerințele de sistem diferă cu fiecare categorie de utilizatori și că a-i mulțumi pe toți este un lucru dificil. Dacă lucrați pe un sistem cu mai mulți utilizatori, vă este de folos (și este amuzant, totodată) să cunoașteți obiceiurile celorlalți și sistemul, pentru a folosi cât mai bine resursele pentru scopurile dumneavoastră.

4.3.5.6 Uinelte grafice

Pentru mediul grafic sunt disponibile destule unelte pentru monitorizare. Mai jos este o captură de ecran care arată Gnome System Monitor (Monitorul de sistem pentru mediul Gnome), care poate căuta și afișa informații despre procese și resursele sistemului:



Sunt disponibile, de asemenea, mai multe aplicații pe care le puteți instala în bara de sarcini, ca pictograme, care monitorizează discurile fixe, memoria și nivelul de solicitare a sistemului. **xload**, de exemplu, este o altă miniaplicație pentru monitorizarea sistemului. Căutați pe acelea care vă plac mai mult!

4.3.5.7. Întreruperea proceselor

Ca utilizator obișnuit, puteți influența doar procesele dumneavoastră. Am arătat deja cum puteți afișa procesele și cum puteți filtra pe acele care aparțin unui anume utilizator și am vorbit despre restricțiile pe care le puteți întâlni. Când realizați că unul din procesele dumneavoastră ocupă prea multe resurse ale sistemului, sunt două lucruri pe care le puteți face:

1. Asigurați-vă că procesul va folosi mai puține resurse fără a fi întrerupt;
2. Oprîți definitiv procesul.

În cazul în care doriți ca acel proces să ruleze în continuare, dar să acordați, în același timp, și altor procese o șansă, puteți oricând să modificați numărul amical (**renice**) al procesului care solicită sistemul. În afară de utilizarea comenziilor **nice** sau **renice**, **top** este întotdeauna o cale ușoară prin care puteți observa procesele care produc necazuri, pentru a le reduce prioritatea.

Identificați procesele problemă în coloana „NI” printre cele care au o prioritate negativă. Tastați **r** și introduceți ID-ul procesului căruia doriți să-i redefiniți prioritatea. Apoi introduceți valoarea numărului amical, de exemplu „20”. Acest lucru înseamnă că cel mult numai 1/5 din ciclurile procesorului vor fi folosite de acest proces.

Emulatoarele, mașinile virtuale, compilatoarele și altele de acest fel sunt exemple de procese pe care nu ați dorit să le oprîți, ci doar să le redefiniți prioritatea.

Dacă doriți să opriți un proces deoarece este înghețat sau pentru că solicită prea mult resursele I/O sau alte resurse ale sistemului, folosiți comanda **kill**. Dacă aveți posibilitatea, încercați să opriți procesul într-un mod mai puțin brutal, trimițând semnalul *SIGTERM*. Aceasta este o instrucțiune care cere oprirea în concordanță cu procedurile descrise în codul programului:

```
joe:~> ps -ef | grep mozilla
joe    25822      1  0 Mar11 ?    00:34:04 /usr/lib/mozilla-1.4.1/mozilla-
joe:~> kill -15 25822
```

În exemplul de mai sus, utilizatorul *joe* a opri Mozilla deoarece răspunde greu la comenzi.

De anumite procese scăpați mai greu. Dacă aveți răbdare, le puteți trimite semnalul *SIGINT* pentru a le opri. Dacă nici acesta nu reușește, folosiți cel mai puternic semnal, *SIGKILL*. În exemplul de mai jos, *joe* oprește Mozilla deoarece înghețase:

```
joe:~> ps -ef | grep mozilla
joe    25915      1  0 Mar11 ?    00:15:06 /usr/lib/mozilla-1.4.1/mozilla-
joe:~> kill -9 25915

joe:~> ps -ef | grep 25915
joe    2634 32273 0 18:09 pts/4    00:00:00 grep 25915
```

În astfel de cazuri, ați putea să verificați dacă procesul este cu adevărat opus, folosind filtrarea **grep** asupra PID. Dacă este returnat doar procesul **grep**, puteți fi siguri că ați reușit oprirea procesului.

Printre procesele care sunt greu de opus se numără și consola. Este un lucru bun: dacă ar fi ușor de terminat, ați fi pierdut consola de fiecare dată când ați fi apăsat accidental **Ctrl-C** în linia de comandă, întrucât combinația de taste menționată trimite semnalul *SIGINT*.



UNIX-ul fără conexiuni (pipes) este de neconcepție

Folosirea conexiunilor () în aşa fel încât ieșirea unei comenzi să devină intrare pentru altă comandă este explicată în capitolul următor, [Capitolul 5](#).

În mediul grafic, programul **xkill** este foarte ușor de folosit. Introduceți numele comenzi, apăsați **Enter** și alegeti fereastra aplicației pe care doriți să o opriți. Este o abordare periculoasă, deoarece comanda trimite implicit semnalul *SIGKILL*, așa că folosiți-o numai în cazurile în care aplicațiile nu mai răspund.

4.4. Programarea proceselor

4.4.1. Folosiți acel timp disponibil!

Un sistem Linux poate suferi din multe cauze, dar suferă, de regulă, doar pe parcursul programului de lucru. Fie că se află într-un mediu de birouri, ori camera serverelor sau acasă, cele mai multe sisteme Linux sunt în aşteptare dimineață, seara, noaptea și la sfârșitul săptămânii. Folosirea acestui timp disponibil se dovedește a fi mult mai ieftin decât cumpărarea unor echipamente de care aveți neapărată nevoie dacă doriți ca toate lucrurile să fie făcute în același timp.

Se disting trei tipuri de execuție întârziată:

- Sarcină pusă în aşteptare pentru puțin timp și apoi relansată în execuție, prin comanda **sleep**. Timpul de execuție depinde de timpul sistemului în momentul introducerii.
- Rularea unei comenzi la un moment de timp specificat, folosind comanda **at**. Executarea sarcinii depinde de timpul sistemului, nu de cel al introducerii.
- Rularea unei comenzi regulat, lunar, săptămânal sau din oră în oră, prin folosirea comenzi **cron**.

Secțiunea următoare va discuta fiecare posibilitate.

4.4.2. Comanda sleep

Paginile Info despre această comandă sunt, probabil, cele mai scurte din căte există. Tot ceea ce face **sleep** este să aștepte. Implicit, timpul de așteptare este exprimat în secunde.

Pentru ce există atunci? Iată câteva exemple practice:

Vă cauță cineva la telefon și îi spuneți „Voi fi la dumneavoastră într-o jumătate de oră”, dar sunteți absorbit de munca dumneavoastră și nu vreți să uitați de întâlnire:

(sleep 1800; echo „Întâlnire cu X...” &

Când nu puteți folosi comanda **at** din diverse motive, să zicem că este ora cinci, doriți să plecați acasă dar încă aveți de lucru și chiar în aceste momente cineva folosește resursele sistemului:

(sleep 10000; programul_meu) &

În aceste cazuri trebuie să aveți activată o modalitate de autodeautentificare și țineți minte să vă deautentificați, să încuați desktop-ul ori biroul sau rulați astfel de sarcini într-o sesiune **screen**.

Dacă aveți foarte multe lucrări de tipărit dar doriți ca și alții să poată tipări în acest timp:

lp foarte_mult_text; sleep 900; lp fișier_imens; sleep 900; lp alt_fișier_imens

Tipărirea fișierelor este discutată în [Capitolul 8](#).

Programatorii folosesc adesea comanda **sleep** pentru a opri execuția unui script sau a unui program pe o anume perioadă de timp.

4.4.3. Comanda at

Comanda **at** execută programele desemnate la timpul precizat, folosind consola dumneavoastră implicită, dacă nu precizați altă opțiune (consultați paginile man).

Opțiunile comenzi **at** sunt destul de prietenoase, după cum vă prezentăm în exemplul de mai jos:

```
steven@home:~> at tomorrow + 2 days
warning: commands will be executed using (in order) a) $SHELL
          b) login shell c) /bin/sh
at> cat reports | mail myboss@mycompany
at> <EOT>
job 1 at 2001-06-16 12:36
```

Apăsarea tastelor **Ctrl+D** determină ieșirea din utilitarul **at** și afișează mesajul „EOT”.

Utilizatorul *steven*, în exemplul următor, face un lucru mai puțin întâlnit, combinând două comenzi; vom studia aceste practici în [Capitolul 5](#), Redirectarea Intrărilor și Ieșirilor (Redirecting I/O).

```
steven@home:~> at 0237
warning: commands will be executed using (in order) a) $SHELL
          b) login shell c) /bin/sh
at> cd new-programs
at> ./configure; make
at> <EOT>
job 2 at 2001-06-14 02:00
```

Opțiunea **-m** trimite un mail utilizatorului la terminarea sarcinii sau oferă explicații în cazurile în care sarcina nu poate fi executată. Comanda **atq** listează sarcinile; introduceți această comandă înainte de programarea unor sarcini pentru a nu introduce în execuție sarcini care să pornească la același moment de timp. Prin comanda **atrm** puteți șterge sarcinile programate, dacă este necesar.

Vă sfătuim să nu programați sarcini la ore fixe, deoarece aceste momente sunt folosite adesea pentru execuția sarcinilor de sistem, după cum veți observa în secțiunea următoare, [Secțiunea 4.4.4](#). De exemplu, astfel de sarcini sunt rulate la ora 01.00 (unu dimineață – de pildă căutarea și indexarea de către

sistem a unei baze de date standard), aşa că programarea unei alte sarcini la aceeaşi oră poate duce la încetinirea sistemului, în loc să-l accelereze. Pentru a preveni situaţiile în care sarcinile sunt rulate în acelaşi timp, puteţi folosi şi comanda **batch**, care verifică procesele şi asigură executarea acestora de către sistem într-o manieră echilibrată, eliminând solicitarea maximă a resurselor sistemului. Consultaţi paginile Info pentru mai multe informaţii.

4.4.4. Cron și crontab

Sistemul cron este gestionat de serviciul (daemon) **cron**. Acest program adună informaţii despre programe şi despre momentele în care acestea sunt programate să ruleze, folosind intrările din fişierul crontab, generate de sistem sau de utilizatori. Numai administratorul de sistem are dreptul să acceseze fişierele crontab ale sistemului, utilizatorii fără privilegii speciale având propriile fişiere crontab în care pot introduce sarcini care vor fi executate la un anumit moment. Pe anumite sisteme, utilizatorii obişnuiţi pot să nu aibă deloc acces la facilităţile **cron**.

La pornirea sistemului, serviciul cron cauă în directorul `/var/spool/cron` intrări în fişierele crontab care sunt denumite după conturile existente în `/etc/passwd`, cauă în `/etc/cron.d/` şi `/etc/crontab`, apoi foloseşte aceste informaţii în fiecare minut, pentru a verifica dacă are ceva de făcut. Va executa comenziile cu drepturile asociate utilizatorului care deține fişierul crontab şi trimite mesaje care conţin ieşirile generate de programele executate acestor utilizatori.

Pe sistemele care folosesc Vixie, sarcinile care se repetă la fiecare oră, zilnic, săptămânal şi lunar au intrări separate, în directoarele `/etc`, pentru a putea fi trecute în revistă mai uşor, comportament care este opus funcţiilor cron standard din UNIX, care foloseşte un singur fişier, mai mare, pentru toate intrările.

Îată un exemplu de fişier crontab Vixie:

```
[root@blob /etc]# more crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
# commands to execute every hour
01 * * * * root run-parts /etc/cron.hourly
# commands to execute every day
02 4 * * * root run-parts /etc/cron.daily
# commands to execute every week
22 4 * * 0 root run-parts /etc/cron.weekly
commands to execute every month
42 4 1 * * root run-parts /etc/cron.monthly
```



Alternativă

Puteţi folosi oricând comanda **crontab -l** pentru afişarea fişierelor crontab.

Sunt stabilite câteva variabile, după care sunt afişate sarcinile programate, fiecare pe o linie separată, începând cu cinci câmpuri care arată ora şi data. Primul câmp conţine minutele (de la 0 la 59), următorul arată ora la care sarcina va fi rulată (0-23), cel de-al treilea câmp arată ziua din lună (1-31), apoi numărul lunii (1-12), ultimul câmp fiind pentru ziua săptămânnii (0-7, atât 0 cât şi 7 fiind duminica). Un asterisc prezent în aceste câmpuri reprezintă acceptarea tuturor valorilor aplicabile aceluiai câmp. Sunt permise şi liste; pentru a rula o sarcină de luni până vineri, introduceţi 1-5 în ultimul câmp, iar pentru a rula o sarcină luni, miercuri şi vineri, introduceţi 1,3,5.

Apoi este arătat utilizatorul pentru care trebuie să ruleze procesele, listat în ultima coloană. Exemplul de mai sus arată o configuraţie cron Vixie, în care administratorul rulează programul **run-parts** la intervale regulate, având ca opţiuni directoarele potrivite. În aceste directoare, sarcinile propriu-zise care trebuie să fie rulate în momentele stabilite, sunt stocate ca scripturi shell, precum următorul, care rulează zilnic, pentru a actualiza baza de date folosită de comanda **locate**:

```
billy@ahost cron.daily]$ cat slocate.cron
#!/bin/sh
renice +19 -p $$ >/dev/null 2>&1
/usr/bin/updatedb -f "nfs,smbfs,ncpfs,proc,devpts" -e \
"/tmp,/var/tmp, /usr/tmp,/afs,/net"
```

Utilizatorii sunt sfătuți să editeze fișierele crontab într-un mod adecvat, prin folosirea comenzi **crontab -e**. Ea preîntâmpină deschiderea a mai mult de o copie a fișierului crontab. Editorul implicit este **vi** (consultați [Capitolul 6](#)), dar puteți folosi orice editor de text, precum **gvim** sau **gedit** dacă vă place mai mult un editor propriu modului grafic.

La încheierea editării, sistemul vă va anunța că este instalat un nou fișier crontab.

Următoarea imagine prezintă o intrare crontab care amintește lui *billy* să meargă la clubul de sport în fiecare marți seară:

```
billy:~> crontab -l
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.20264 installed on Sun Jul 20 22:35:14 2003)
# (Cron version -- $Id: chap4.xml,v 1.28 2007/09/19 12:22:26 tille Exp $)
38 16 * * 3 mail -s "sports evening" billy
```

După adăugarea unei noi sarcini programate, sistemul vă va spune că este instalat un nou fișier crontab. Nu trebuie să restartați serviciul **cron** pentru ca schimbările să-și facă efectul. Mai jos, *billy* a adăugat o linie care întețește către un script care face copii de rezervă:

```
billy:~> crontab -e
45 15 * * 3 mail -s "sports evening" billy
4 4 * * 4,7 /home/billy/bin/backup.sh

<--write and quit-->

crontab: installing new crontab

billy:~>
```

Scriptul *backup.sh* este executat în fiecare joi și duminică. Consultați [Sectiunea 7.2.5](#) pentru noțiuni introductive în scriptingul shell. Rețineți că ieșirile unei comenzi (dacă există) sunt trimise prin mail deținătorului fișierului crontab. Dacă nu este configurat nici un serviciu de poștă electronică, puteți găsi aceste ieșiri generate de comenzi în directorul care păstrează mesajele mail locale, */var/spool/mail/<numele_dumneavoastră>*, ca fișiere text simplu.



Cine rulează comenziile mele?

Nu trebuie să specificați utilizatorul care trebuie să execute comenziile pe care le programați. Ele vor fi rulate implicit folosind permisiunile deținătorului.

4.5. Sumar

Linux-ul este un sistem de operare multi-utilizatori și multi-sarcini care gestionează procesele în stilul UNIX-ului. Viteza de executare a proceselor depinde de mii de lucruri mărunte. Printre altele, ați învățat multe comenzi noi care vă ajută la vizualizarea și gestionarea proceselor. Vă prezintăm o listă cu cele mai des folosite comenzi în gestionarea proceselor:

Tabelul 4-3. Comenzi noi în Capitolul 4: Procesele

Comanda	Înțelesul comenzi
at	Ordonează sarcinile pentru rularea programată.
atq	Arată sarcinile care așteaptă să fie rulate.
atrm	Șterge sarcinile care așteaptă să fie rulate, după numărul asociat lor.
batch	Execută sarcinile atunci când nivelul de solicitare a sistemului permite acest lucru.
crontab	Gestionează fișierele crontab pentru utilizatorii individuali.
halt	Oprește sistemul.
init run level	Inițializează controlul procesului.

jobs	Listează sarcinile care rulează în acel moment.
kill	Termină un proces.
mesg	Controlează trimiterea de mesaje către terminalul tău.
netstat	Afișează legăturile de rețea, tabela de routing, statisticile de rețea, conexiunile deghizate (masquerade) și difuzările multiple (multicast).
nice	Rularea unui program cu modificarea priorității acestuia.
pgrep	Afișează procesele.
ps	Arată statutul proceselor.
pstree	Afișează o imagine arborescentă a proceselor.
reboot	Oprește sistemul.
renice	Modifică prioritatea proceselor aflate în desfășurare.
shutdown	Oprește definitiv sistemul.
sleep	Determină o întârziere cu un anumit timp specificat.
time	Cronometreză o comandă sau afișează folosirea resurselor.
top	Afișează procesele care solicită procesorul.
uptime	Arată de când rulează sistemul.
vmstat	Afișează statistici referitoare la memoria virtuală.
w	Arată cine este autentificat și ce face.
wall	Trimite un mesaj terminalelor celorlalți utilizatori.
who	Arată cine este autentificat.
write	Trimite un mesaj altui utilizator.

4.6. Exerciții

Vă propunem câteva exerciții care vă vor ajuta să vă formați o idee despre procesele care rulează pe sistemul dumneavoastră.

4.6.1. Generalități

- Rulați comanda **top** într-un terminal și faceți-vă exercițiile într-altul.
- Rulați comanda **ps**.
- Citiți în paginile man cum puteți să afișați toate procesele.
- Rulați comanda **find** /. Ce efect are asupra nivelului de solicitare a sistemului? Oprîți această comandă.
- În modul grafic, porniți programul **xclock** în prim-plan. Apoi mutați-l în fundal. Oprîți programul prin comanda **kill**.
- Rulați comanda **xcalc** direct în fundal, pentru ca prompterul terminalului în care ați introdus comanda să fie eliberat.
- Ce face comanda **kill -9 -1**?
- Deschideți două terminale sau ferestre terminal și folosiți **write** ca să trimiteți mesaje dintr-unul în celălalt.
- Introduceți comanda **dmesg**. Ce vă spune?
- Cât durează rularea comenzi **ls** în directorul curent?
- Folosindu-vă de intrările proceselor din /proc, deținute de UID-ul dumneavoastră, cum puteți afla ce reprezintă de fapt aceste proceze?
- De cât timp este pornit sistemul dumneavoastră?
- Care este terminalul (TTY) pe care-l folosiți în acest moment?
- Numiți trei procese care nu au fost generate de **init** ca părinte inițial.

- Numiți trei comenzi care folosesc modul SUID. Explicați de ce.
 - Numiți comenzi care determină, în general, un nivel de solicitare mare în sistemul dumneavoastră.
-

4.6.2. Pornirea (booting), inițializarea (init) etc.

- Puteți restarta sistemul ca utilizator obișnuit? De ce?
 - În funcție de nivelul de rulare în care vă aflați, numiți pașii care sunt făcuți pe parcursul opririi.
 - Cum puteți schimba nivelul de rulare? Treceți din nivelul de rulare implicit în nivelul de rulare 1 și înapoi.
 - Faceți o listă cu toate serviciile și sarcinile care sunt rulate la pornirea sistemului.
 - Ce kernel este încărcat la pornire?
 - Să presupunem că trebuie să rulați un serviciu neobișnuit la pornirea sistemului. Până în acest moment, vă autentificați după pornirea sistemului și rulați acest serviciu manual, folosind un script numit `livrează_pizza` din directorul personal. Ce trebuie să faceți pentru ca acest serviciu să fie pornit automat în nivelul de rulare 4, pe care l-ați definit numai în acest scop?
-

4.6.3. Programarea

- Folosiți `sleep` pentru a crea un anunț care vă amintește că pastele dumneavoastră sunt gata în zece minute.
 - Creați o sarcină `at` care copiază toate fișierele din directorul personal în `/var/tmp` peste o jumătate de oră. Va trebui să creați un subdirector în `/var/temp`, dacă dorîți ca să păstrați ușor ordinea pe discul fix.
 - Creați o sarcină cron care face același lucru în fiecare săptămână, de luni până vineri, în timpul prânzului.
 - Verificați dacă funcționează.
 - Faceți o greșeală într-o intrare crontab, precum indicarea comenzi inexistente `copy`, în locul `cp`. Ce se întâmplă la executarea sarcinii?
-

Capitolul 5. Redirecționarea I/O

Acest capitol prezintă mai multe detalii despre mecanismul puternic al UNIX-ului prin care sunt redirecționate intrările, ieșirile și erorile. Aspectele discutate includ:

- ◆ Intrările standard, ieșirile și erorile.
- ◆ Operanții de redirecționare.
- ◆ Cum să folosiți ieșirea unei comenzi ca intrare pentru alta.
- ◆ Cum să puneti ieșirea unei comenzi într-un fișier pentru consultare ulterioară.
- ◆ Cum să adăugați ieșirile mai multor comenzi într-un fișier.
- ◆ Redirecționarea intrărilor.
- ◆ Gestionarea mesajelor standard de eroare.
- ◆ Combinarea redirecționărilor intrărilor, ieșirilor și fluxurilor de erori.
- ◆ Filtrarea ieșirilor.

5.1. Redirecționări simple

5.1.1. Ce sunt intrările standard și ieșirile standard?

Cele mai multe comenzi Linux citesc intrările, precum un fișier sau alte atrbute ale comenzi și scriu ieșirile. Implicit, intrările sunt generate de la tastatură, iar ieșirile sunt afișate pe ecranul dumneavoastră. Tastatura este *dispozitivul dumneavoastră standard de intrare* (stdin), iar ecranul, sau o anume fereastră terminal, *dispozitivul dumneavoastră standard de ieșire* (stdout).

Totuși, deoarece Linux-ul este un sistem flexibil, aceste opțiuni de configurare implicate nu sunt aplicate în toate cazurile. Ieșirea standard, de exemplu, în cazul unui server monitorizat îndeaproape, aflat într-un mediu dezvoltat, poate fi o imprimantă.

5.1.2. Operanții de redirecționare

5.1.2.1. Redirecționarea ieșirilor cu > și |

Câteodată trebuie să puneti ieșirea unei comenzi într-un fișier sau doriti ca ieșirea unei comenzi să devină intrare pentru altă comandă. Aceste lucruri sunt cunoscute sub numele de redirecționări ale ieșirilor. Redirecționările sunt făcute prin „>” (simbolul mai mare decât) sau prin folosirea operandului „|” (conexiune), care trimite ieșirea standard a unei comenzi ca intrare standard pentru altă comandă.

După cum am văzut, comanda **cat** conexează (înlănțuie) fișierele și le afișează la ieșirea standard. Prin redirectarea acestor ieșiri către un fișier, el va fi creat – sau suprascris dacă există deja, aşa că fiți foarte atenți cu această comandă.

```
nancy:~> cat test1
some words

nancy:~> cat test2
some other words

nancy:~> cat test1 test2 > test3

nancy:~> cat test3
some words

some other words
```



Nu suprascrieți!

Aveți grijă să nu suprascrieți fișierele (importante) existente atunci când redirecționați ieșirile. Multe

console, printre care și Bash, dispun de o facilitate care vă protejează de acest risc: **noclobber**. Consultați paginile Info pentru mai multe informații. În Bash, puteți adăuga comanda **set -o noclobber** la fișierul de configurare `.bashrc` pentru a preveni suprascrierea accidentală a unui fișier.

Redirecționarea a „nimic” către un fișier existent este același lucru cu ștergerea conținutului acelui fișier:

```
nancy:~> ls -l list
-rw-rw-r-- 1 nancy nancy 117 Apr 2 18:09 list

nancy:~> > list

nancy:~> ls -l list
-rw-rw-r-- 1 nancy nancy 0 Apr 4 12:01 list
```

Acest proces este numit *trunchiere* (*truncating*).

Același tip de redirecționare către un fișier inexistent va crea un nou fișier, gol, cu numele pe care l-ați introdus în comanda de redirecționare:

```
nancy:~> ls -l newlist
ls: newlist: No such file or directory

nancy:~> > newlist

nancy:~> ls -l newlist
-rw-rw-r-- 1 nancy nancy 0 Apr 4 12:05 newlist
```

[Capitolul 7](#) va oferi mai multe exemple despre cum să folosiți acest tip de redirecționare.

Îată câteva exemple despre folosirea conexiunilor în comenzi:

Pentru a găsi un cuvânt în cadrul unui text, afișați toate liniile care conțin „criteriu1” și excludeți liniile care conțin „criteriu2”:

grep criteriu1 fișier | grep -v criteriu2

Pentru a afișa conținutul unui director cu listarea paginilor una câte una:

ls -la | less

Pentru a căuta un fișier într-un director:

ls -l | grep nume_partial_fișier

5.1.2.2. Redirecționarea intrărilor

În alte cazuri, poate că doriți ca un fișier să devină intrare pentru o comandă care, în mod normal, nu ar accepta un fișier ca opțiune. Această redirecționare a intrărilor se face folosind ca operand simbolul „<” (mai mic decât).

Vă prezentăm un exemplu care arată cum puteți trimite un fișier cuiva, folosind redirecționarea intrărilor.

```
andy:~> mail mike@somewhere.org < to_do
```

Dacă utilizatorul *mike* există în sistem, nu trebuie să introduceți adresa întreagă. Dacă doriți să trimiteți fișierul cuiva pe Internet, introduceți întreaga adresă de poștă electronică ca argument al comenзii **mail**.

Poate că este mai dificil decât folosirea comenзii preferate de începători **cat fișier | mail cuiva**, dar este, fără îndoială, o modalitate elegantă prin care folosiți uneltele pe care le aveți la dispozitie.

5.1.2.3. Combinarea redirectionărilor

Următorul exemplu arată cum pot fi combinate redirectionările intrărilor și ieșirilor. Fișierul `text.txt` este verificat mai întâi din punct de vedere al ortografiei, iar ieșirea generată este trimisă la un fișier jurnal de erori:

```
spell <text.txt> error.log
```

Comanda următoare listează toate comenziile pe care le puteți folosi pentru examinarea fișierelor atunci când folosiți `less`:

```
mike:~> less --help | grep -i examine
:e [file]      Examine a new file.
:n            * Examine the (N-th) next file from the command line.
:p            * Examine the (N-th) previous file from the command line.
:x            * Examine the first (or N-th) file from the command line.
```

Oțjuna `-i` este folosită pentru căutările care nu țin cont de literele mari sau mici – amintiți-vă că sistemele UNIX sunt case-sensitive.

Dacă doriți să păstrați ieșirea acestei comenzi pentru a o folosi mai târziu, redirectați-o:

```
mike:~> less --help | grep -i examine > examine-files-in-less

mike:~> cat examine-files-in-less
:e [file]      Examine a new file.
:n            * Examine the (N-th) next file from the command line.
:p            * Examine the (N-th) previous file from the command line.
:x            * Examine the first (or N-th) file from the command line.
```

Ieșirea unei comenzi poate fi conexată (piped) altei comenzi ori de câte ori doriți, atât timp cât aceste comenzi vor citi intrările de la intrarea standard și vor afișa ieșirile la ieșirea standard. Câteodată nu fac acest lucru, dar în aceste cazuri există opțiuni speciale care spun acestor comenzi să se comporte în concordanță cu definițiile standard; citiți documentația (paginile man și Info) comenziilor pe care le folosiți în cazul în care întâmpinați dificultăți.

Vă atragem din nou atenția asupra folosirii numelor unor fișiere existente de care mai aveți nevoie. Redirecționarea ieșirilor către acestea va înlocui conținutul lor.

5.1.2.4. Operandul `>`

Pentru a nu suprascrie datele dintr-un fișier puteți adăuga text la un fișier existent, folosind două caractere `>>` (mai mare decât):

Exemplu:

```
mike:~> cat wishlist
more money
less work

mike:~> date >> wishlist

mike:~> cat wishlist
more money
less work
Thu Feb 28 20:23:07 CET 2002
```

Comanda `date` va pune, în mod normal, ultima linie pe ecran; acum ea este atașată fișierului `wishlist`.

5.2. Facilități avansate de redirecționare

5.2.1. Utilizarea descriptorilor de fișiere

Sunt trei tipuri de I/O, fiecare având propriul identificator, denumit descriptor de fișiere:

- intrarea standard: 0
- ieșirea standard: 1
- eroarea standard: 2

În descrierile următoare, dacă numărul descriptorului de fișiere este omis și primul caracter este operandul <, redirecționarea se referă la intrarea standard (descriptorul de fișiere 0). Dacă primul caracter este operandul >, redirecționarea se referă la ieșirea standard (descriptorul de fișiere 1).

Câteva exemple practice vor clarifica lucrurile:

ls > dirlist 2>&1

va direcționa atât ieșirile cât și erorile standard către fișierul dirlist, în timp ce comanda

ls 2>&1 >dirlist

va direcționa doar ieșirea standard către fișierul dirlist. Această opțiune este folositoare programatorilor.

Lucrurile se complică din acest moment, de exemplu nu confundați folosirea semnului „&” cu sensul pe care îl are acesta în [Sectiunea 4.1.2.1](#), unde era folosit pentru rularea în fundal a unui proces. Aici el servește ca indicator care arată faptul că numărul care urmează nu este un nume de fișier, ci un loc către care întreține fluxul de date. Vă atragem atenția asupra faptului că semnul „mai mare decât” nu este separat prin spații de numărul descriptorului de fișiere. Dacă ar fi separat, am fi întîrit din nou către ieșirea unui fișier. Exemplul de mai jos vă demonstrează acest lucru:

```
[nancy@asus /var/tmp]$ ls 2> tmp  
[nancy@asus /var/tmp]$ ls -l tmp  
-rw-rw-r-- 1 nancy nancy 0 Sept 7 12:58 tmp  
  
[nancy@asus /var/tmp]$ ls 2 > tmp  
ls: 2: No such file or directory
```

Prima comandă introdusă de *nancy* este corectă (nu este generată nici o eroare, astfel încât fișierul către care este redirecționată eroarea standard este gol). Cea de-a doua comandă consideră 2 ca fiind numele unui fișier, care, în cazul nostru, nu există, aşa că este afișată o eroare.

Toate aceste facilități sunt explicate în detaliu în paginile Info Bash.

5.2.2. Exemple

5.2.2.1. Analizarea erorilor

Dacă procesele generează multe erori, iată o cale prin care le puteți examina:

comandă 2>&1 | less

Este utilizată pe larg atunci când sunt realizate programe noi prin comanda **make**, precum în:

```
andy:~/newsoft> make all 2>&1 | less  
--output omitted--
```

5.2.2.2. Separarea ieșirilor standard de erorile standard

Construcții ca acestea sunt folosite adesea de programatori, pentru ca ieșirile să fie afișate într-o fereastră terminal, iar erorile în alta. Stabilitiți mai întâi ce pseudoterminal folosiți, prin comanda **tty**:

```
andy:~/newsoft> make all 2> /dev/pts/7
```

5.2.2.3. Scrierea simultană a ieșirilor în fișiere

Puteți folosi comanda **tee** pentru a copia intrările la ieșirea standard și într-unul sau mai multe fișiere într-un singur pas. Cu opțiunea **-a** aplicată lui **tee** intrările vor fi adăugate fișierului (fișierelor). Comanda este folositoare în cazurile în care doriti atât să consultați cât și să salvați ieșirile generate. Operanții > și >> nu permit executarea ambelor acțiuni simultan.

Unealta este apelată printr-o conexiune (), după cum vă arată exemplul următor:

```
mireille ~/test> date | tee file1 file2
Thu Jun 10 11:10:34 CEST 2004

mireille ~/test> cat file1
Thu Jun 10 11:10:34 CEST 2004

mireille ~/test> cat file2
Thu Jun 10 11:10:34 CEST 2004

mireille ~/test> uptime | tee -a file2
11:10:51 up 21 days, 21:21, 57 users, load average: 0.04, 0.16, 0.26

mireille ~/test> cat file2
Thu Jun 10 11:10:34 CEST 2004
11:10:51 up 21 days, 21:21, 57 users, load average: 0.04, 0.16, 0.26
```

5.3. Filtre

Realizarea de către un program a unei operații prin citirea intrării și afișarea rezultatului la ieșirea standard, se numește filtrare. Cea mai utilizată filtrare este cea prin care sunt restructurate ieșirile. Vom discuta câteva filtre mai importante în cele ce urmează.

5.3.1. Mai multe despre grep

După cum am arătat în [Secțiunea 3.3.3.4](#), **grep** scanează ieșirile, linie cu linie, în căutarea tiparelor. Toate liniile care conțin tiparul vor fi afișate la ieșirea standard. Comportamentul poate fi inversat prin opțiunea **-v**.

Câteva exemple: să presupunem că dorim să știm ce fișiere dintr-un anumit director au fost modificate în februarie:

```
jenny:~> ls -la | grep Feb
```

Comanda **grep**, ca majoritatea comenzilor, face deosebire între literele mari și cele mici. Folosiți opțiunea **-i** pentru ca această comandă să nu facă diferență între majuscule și literele mici. Multe extensii ale GNU sunt disponibile, precum **--color**, care vă ajută la evidențierea tiparelor de căutare în liniile prea lungi și **--after-context**, care afișează numărul liniilor după ultima linie care se potrivește. Puteți introduce o comandă **grep** recursivă care caută în toate subdirectoarele unui director prin opțiunea **-r**. Opțiunile pot fi combinate, ca în mai toate cazurile.

Expresiile regulate pot fi folosite pentru detalierea până la potrivirea exactă a caracterelor pe care doriti să le selectați din toate liniile de intrare. Cea mai bună cale de a începe în privința expresiilor regulate este citirea documentației **grep**. Un capitol excelent este inclus în paginile Info ale comenzii **grep**. Deoarece

discutarea intrărilor și ieșirilor expresiilor regulate ne-ar purta prea departe, vă sfătuim să începeți cu acel capitol dacă doriți să cunoașteți mai multe despre acest subiect.

Jucăți-vă puțin cu **grep**, deoarece merită efortul și timpul petrecut cu deprinderea acestei puternice comenzi de filtrare. Exercițiile de la sfârșitul acestui capitol vor fi un bun început, consultați aşadar [Sectiunea 5.5](#).

5.3.2. Filtrarea ieșirilor

Comanda **sort** aranjează implicit liniile în ordine alfabetică:

```
thomas:~> cat people-I-like | sort
Auntie Emmy
Boyfriend
Dad
Grandma
Mum
My boss
```

Dar **sort** poate face multe alte lucruri. Poate spune mărimea fișierului, de exemplu. Cu ajutorul acestei comenzi puteți sorta fișierele după mărime, de la cele mai mici la cele mai mari:

ls -la | sort -nk 5



Sintaxa veche

Este posibil să obțineți același rezultat prin folosirea comenzi **ls -la | sort +4n**, dar este o formă veche care nu este conformă cu standardele de azi.

Comanda **sort** poate fi folosită în combinație cu programul **uniq** (sau **sort -u**) pentru a sorta ieșirile și a filtra intrările duble:

```
thomas:~> cat itemlist
1
4
2
5
34

567
432
567
34
555

thomas:~> sort itemlist | uniq
1
2
34
4
432
5
555
567
```

5.4. Sumar

În acest capitol ați învățat cum puteți cupla comenzi și cum ieșirea unei comenzi poate fi folosită ca intrare pentru altă comandă.

Redirectionarea intrărilor/ieșirilor este o sarcină obișnuită pe sistemele UNIX și Linux. Acest mecanism puternic permite folosirea flexibilă a „cărămizilor”, părților din care este constituit UNIX-ul.

Tabelul 5-1. Comenzi noi în Capitolul 5: Redirecționarea I/O

Comanda	Înțelesul comenzi
date	Afișează informații despre oră și dată.
set	Configurează consola (opțiunile acesteia).
sort	Sortează liniile de text.
uniq	Exclude duplicatele liniilor dintr-un fișier sortat.

5.5. Exerciții

Exercițiile care urmează sunt exemple despre cum puteți combina comenzi. Scopul principal este folosirea cât mai redusă a tastei **Enter**.

Faceți toate exercițiile folosind un cont obișnuit și nu fiți surprinși dacă unele comenzi vor returna erori. Când le întâlniți, nu uitați să citiți paginile man referitoare la comenziile în cauză!

- Folosiți comanda **cut** pe ieșirea generată de listarea unui director care conține multe fișiere pentru a afișa doar permisiunile asociate fișierelor. Apoi realizați o conexiune cu **sort** și **uniq** pentru a filtra orice duplicate. Folosiți apoi **wc** pentru a număra permisiunile diferite din acel director.
- Trimiteti ieșirea comenzi **date** într-un fișier. Adăugați ieșirea comenzi **ls** aceluiași fișier. Trimiteti acest fișier către căsuța de poștă electronică locală (nu specificați nici un <@domeniu>, doar numele de utilizator va fi suficient). În cazul în care folosiți Bash, veți observa o notificare de mesaj nou în căsuța locală dacă ati efectuat corect exercițiul.
- Listați dispozitivele din **/dev** care sunt folosite de UID-ul dumneavoastră. Conexați prin **less** pentru a le viziona cum se cuvine.
- Introduceți următoarele comenzi ca utilizator obișnuit. Determinați intrarea, ieșirea și eroarea standard pentru fiecare comandă.
 - ◆ **cat fișier_inexistent**
 - ◆ **file /sbin/ifconfig**
 - ◆ **grep root /etc/passwd /etc/fișier_inexistent >rezultat_grep**
 - ◆ **/etc/init.d/sshd start > /var/tmp/output**
 - ◆ **/etc/init.d/cron start > /var/tmp/output 2>&1**
 - ◆ Verificați rezultatele prin introducerea din nou a același comenzi, redirectând ieșirea standard către fișierul **/var/tmp/output** și erorile standard către fișierul **/var/tmp/error**.
- Câte procese rulează în acest moment pe sistemul dumneavoastră?
- Câte fișiere ascunse sunt în directorul personal?
- Folosiți **locate** pentru a găsi documentația despre kernelul folosit.
- Spuneți ce fișier conține intrarea următoare:

```
root:x:0:0:root:/root:/bin/bash
```

Și aceasta:

```
system:          root
```

- Spuneți ce se întâmplă când introduceți comanda următoare:

> time; date >> time; cat < time
- Ce comandă veți folosi pentru a verifica care script din directorul **/etc/init.d** pornește un anume proces?

Capitolul 6. Editoare de text

În acest capitol vom discuta importanța stăpânirii unui editor de text. Ne vom concentra îndeosebi asupra editorului îmbunătățit **vi**.

După terminarea acestui capitol, veți putea să:

- ◆ Deschideți și închideți fișierele în modul text.
- ◆ Editați fișiere.
- ◆ Căutați în fișierele text.
- ◆ Reveniți la starea inițială (repararea erorilor).
- ◆ Alăturați fișiere.
- ◆ Recuperăți fișiere pierdute.
- ◆ Găsiți un program sau suită pentru birou.

6.1. Editoare de text

6.1.1. De ce trebuie să folosesc un editor de text?

Este foarte important să stăpâniți măcar un editor care lucrează în modul text. Deprinderea folosirii unui astfel de editor de text este primul pas către independență.

Va trebui să vă obișnuiji cu un editor până la începutul următorului capitol, deoarece va fi necesar să editați fișiere care influențează mediul sistemului dumneavoastră. Ca utilizatori avansați, poate că veți dori să scrieți scripturi sau cărți, să dezvoltați saituri web sau programe noi. Stăpânirea unui editor va îmbunătăți enorm productivitatea dumneavoastră, laolaltă cu abilitățile de care dispuneți.

6.1.2. Ce editor ar trebui să folosesc?

Vom vorbi mai multe despre editoarele de text, care pot fi folosite pe sisteme care nu dispun de interfață grafică sau a căror interfață grafică nu poate fi accesată, care lucrează și în ferestrele terminal. Un avantaj în plus care vă este asigurat de astfel de editoare este vizibil atunci când lucrați la distanță, pe alte sisteme. Întrucât nu este nevoie să transferați întregul mediu grafic pe rețea, lucrul cu editoarele de text îmbunătățește substanțial viteza rețelei.

Sunt, ca de obicei, mai multe căi prin care să abordați această problemă. Vă prezentăm cele mai întâlnite editoare:

6.1.2.1. Emacs GNU

Emacs este un editor extensibil, care poate fi personalizat, ce conține documentația necesară, cu afișare în timp real, cunoscut pe multe sisteme UNIX și non-UNIX. Textul care este editat este vizibil pe ecran și este actualizat automat în timp ce introduceți comenzi. Este cu afișare în timp real deoarece ecranul este actualizat foarte frecvent, de regulă imediat după introducerea unui caracter sau a unei perechi de caractere. Acest lucru minimizează cantitatea de informație pe care trebuie să o rețineți în timp ce editați. Este avansat deoarece asigură facilități care trec de simpla inserare și ștergere: controlează subprocesele; indentează automat programele; afișează două sau mai multe fișiere deodată; editează text formatat; gestionează după caracter, cuvinte, linii, propoziții, paragrafe și pagini, ca și după expresii și comentarii proprii cătorva limbaje de programare.

Documentația înseamnă că oricând tastăți anumite caractere, de regulă **Ctrl+H**, vi se prezintă opțiunile pe care le aveți. Poate fi personalizat pentru că puteți schimba anumite definiții ale comenzielor Emacs. De exemplu, dacă folosiți un limbaj de programare în care comentariile încep cu „`<**>`” și se termină cu „`**>`”, puteți spune editorului Emacs să folosească aceste siruri (tipare) în comenzi de gestionare ale comentariilor. Alt tip de personalizare este posibilitatea de rearanjare a introducerii comenziilor. De exemplu, dacă preferați ca cele patru direcții principale de mișcare ale cursorului (sus, jos, stânga și dreapta) să fie reprezentate pe tastatură într-un model de diamant, puteți realoca tastele în acest scop.

Extensibil înseamnă că puteți trece de simpla personalizare către scrierea de comenzi în întregime noi, programe în limbajul Lisp care sunt rulate de propriul interpretor Lisp al Emacs. Emacs este un sistem

extensibil *în timp real* deoarece este divizat în mai multe funcții care pot comunica, fiind posibilă redefinirea oricărei funcții chiar în mijlocul unei sesiuni de editare. Aproape orice parte a Emacs poate fi înlocuită fără a fi necesară realizarea unei copii separate a întregului Emacs. Cele mai multe comenzi de editare ale Emacs sunt scrise deja în Lisp; există câteva excepții în care comenzi care ar fi putut fi scrise în Lisp au fost scrise în limbajul C pentru eficiență. Cu toate că doar un programator poate scrie o extensie, oricine poate folosi după ce a fost scrisă.

Când rulează în interfață grafică – X Window System (pornit ca **xemacs**) Emacs furnizează un meniu propriu și disponibili convenabile ale butoanelor mausului. Dar Emacs asigură multe din beneficiile unei interfețe grafice chiar în mediul text. De exemplu, puteți afișa sau edita mai multe fișiere în același timp, puteți muta text dintr-un fișier în altele și puteți edita fișiere în timp ce rulați comenzi shell.

6.1.2.2. Vi(m)

Vim înseamnă „Vi IMproved” - Vi ÎMbunătățit. Era cunoscut drept „Vi IMitation” - Vi IMitat, dar există atât de multe îmbunătățiri, încât schimbarea de nume este justificată. Vim este un editor de text care include aproape toate comenzi din programul UNIX **vi**, precum și multe altele, noi.

Comenzi sunt introduse în editorul **vi** folosind numai tastatura, ceea ce are avantajul că puteți ține mâinile pe tastatură și ochii pe ecran, în loc să vă mutați în mod repetat mâna pe maus. Pentru cei care nu consideră introducerea doar de la tastatură a comenziilor ca fiind un avantaj, pot fi activate atât integrarea mausului, cât și o versiune grafică, care asigură meniu și bară de derulare.

Vom face referire la **vi** sau **vim** în această carte când vom vorbi despre editarea fișierelor, dumneavoastră având libertatea de a alege orice editor doriti. Totuși, noi vă recomandăm să asimilați măcar noțiuni elementare legate de **vi**, deoarece este editorul standard pe aproape toate sistemele UNIX, Emacs fiind un pachet optional. Pot exista diferențe mici între diverse calculatoare sau terminale, dar ideea principală este aceea că dacă puteți lucra cu **vi**, aveți toate şansele să vă descurcați cu orice sistem UNIX.

În afară de comenziile **vim**, pachetul **vim** poate conține **gvim**, versiunea Gnome a lui **vim**. Utilizatorii începători îl consideră mai ușor de folosit, deoarece meniurile le oferă ajutor în cazurile în care uită sau nu știu cum să îndeplinească o anumită sarcină folosind comenziile standard ale **vim**.

6.2. Utilizarea editorului Vim

6.2.1. Două moduri

Editorul **vi** este o unealtă puternică care dispune de un manual cuprinsător, care poate fi activat prin comanda :**help** când programul este pornit (în loc să folosiți **man** sau **info**, care nu conțin atât de multe informații). Vom discuta doar comenziile principale care vă permit să începeți să lucrăți cu **vim**.

Ceea ce face ca **vi** să fie confuz pentru începători este faptul că poate opera în două moduri: modul comandă și modul inserare. Modul comandă vă permite să navigați prin text, să căutați, să înlocuiți, să marcați porțiuni și să realizați alte sarcini de editare care pot să treacă editorul în modul inserare.

Acest lucru înseamnă că orice tastă are nu unul, ci mai degrabă două înțelesuri: poate ori să reprezinte o comandă a editorului atunci când vă aflați în modul comandă, ori un caracter pe care îl doriți inserat în text în cazul în care vă aflați în acest mod.



Pronunție

Se pronunță „vee-eye”.

6.2.2. Comenziile principale

6.2.2.1. Navigarea prin text

Navigarea prin text se face, de regulă, cu tastele săgeți. Dacă nu reușești, folosiți:

- **h** pentru a muta cursorul către stânga

- **I** pentru a-l muta către dreapta
- **k** pentru a-l muta în sus
- **j** pentru a-l muta în jos

SHIFT – G vă va duce la sfârșitul documentului.

6.2.2. Operații de bază

Iată câteva comenzi **vi** foarte des folosite:

- **n dd** va șterge n linii începând din poziția curentă a cursorului.
 - **n dw** va șterge n cuvinte din partea dreaptă a cursorului.
 - **x** va șterge caracterul asupra căruia este poziționat cursorul.
 - **:n** vă mutați pe linia n a fișierului
 - **:w** va aplica modificările – va salva – (scrive) fișierul.
 - **:q** veți ieși din editor.
 - **:q!** forțează ieșirea atunci când doriți să ieșiți dintr-un fișier care conține schimbări neaplicate.
 - **:wq** va aplica modificările și veți ieși din editor.
 - **:fișier_nou** va salva textul în fișierul **fișier_nou**.
 - **:wq!** trece peste permisiunile asociate fișierelor (dacă aveți, la rândul dumneavoastră, permisiunea să faceți acest lucru, de exemplu dacă utilizați un cont de *root*).
 - **/așir** (/astring) va căuta sirul în fișier și va poziționa cursorul la prima potrivire găsită, dedesubtul poziției sale.
 - **/** va repeta căutarea același sir, mutând cursorul la potrivirea următoare.
 - **:1, \$s/cuvânt/alt_cuvânt/g** va înlocui cuvânt cu alt_cuvânt în tot fișierul.
 - **yy** va copia un bloc de text.
 - **:recover** va recupera un fișier în cazul unei întreruperi neașteptate.
-

6.2.3. Calea mai ușoară

În loc să citiți tot textul, un lucru plăcitos, puteți folosi vimtutor pentru a învăța primele comenzi Vim. Este un tutorial de treizeci de minute care vă învață cele mai întâlnite comenzi Vim în opt exerciții ușoare. Nu puteți învăța chiar totul despre **vim** doar într-o jumătate de oră, dar veți cunoaște destul comenzi care vă ajută să folosiți Vim ca editor de text potrivit tuturor scopurilor.

În UNIX și MS Windows, dacă Vim este instalat cum trebuie, puteți porni programul din consolă sau linia de comandă, prin introducerea comenzi **vimtutor**. Ea va face o copie a fișierului tutore, aşa că o puteți edita fără teamă că distrugeti originalul. Există și câteva traduceri ale tutorialului. Pentru a afla dacă este disponibil și în limba dumneavoastră, folosiți cele două litere ale codului de limbă. Pentru franceză, introduceți **vimtutor fr** (dacă este instalat în sistem).

6.3. Linux la birou

6.3.1. Istorie

Pe parcursul ultimei decenii domeniul birotică a fost dominat fără săgădă de MS Office, și, să recunoaștem: formatele Microsoft Word, Excel și PowerPoint sunt standarde ale industriei cu care veți avea de-a face mai devreme sau mai târziu.

Această situație în care Microsoft are o poziție monopolistă s-a dovedit a fi un mare dezavantaj în atragerea de noi utilizatori către Linux, aşa că un grup de dezvoltatori germani au început proiectul StarOffice, care a fost, și încă este, gândit pentru a deveni o clonă a suitei MS Office. Compania lor, StarDivision, a fost achiziționată de Sun Microsystems pe la sfârșitul anilor 90, chiar înainte de lansarea versiunii 5.2. Sun continua dezvoltarea, dar restrânge accesul la surse. Însă dezvoltarea setului original de

surse este continuată de comunitatea Sursei deschise (Open Source community), care a trebuit să redenumească proiectul OpenOffice. OpenOffice este disponibil acum pentru mai multe platforme, inclusiv pentru MS Windows, Linux, MacOS și Solaris. V-am prezentat o imagine în [Sectiunea 1.3.2](#).

Aproape simultan, alte proiecte cunoscute acum, au fost demarate. O alternativă apreciată este Koffice, suita de birou care este foarte populară printre utilizatorii distribuției SuSE. La fel cu originalul, această clonă încorporează programe compatibile cu MS Word și Excel și multe altele.

Proiecte mai mici sunt destinate unor programe particulare ale suitei MS Office, precum Abiword și MS Wordview pentru compatibilitatea cu documentele MS Word și Gnumeric pentru vizualizarea și crearea foilor de calcul compatibile cu Excel.

6.3.2. Suite și programe

Distribuțiile curente vin cu toate uneltele necesare. Întrucât acestea asigură o documentație excelentă, indexată pentru căutări în meniurile de ajutor, nu le vom discuta în detaliu. Consultați aşadar documentația sistemului sau saiturile proiectelor, precum:

- <http://www.openoffice.org/>
 - <http://www.koffice.org/>
 - [Freshmeat](#) și [Sourceforge](#) pentru alte proiecte.
-

6.3.3. Remarci

6.3.3.1. Utilizarea generală a documentelor de birou

Încercați să limitați folosirea documentelor de birou la scopurile pentru care au fost create: biroul.

Iată un exemplu: dacă trimiteți un mail care spune cam aşa: „Salut, vreau să vă spun ceva, vezi atașamentul”, iar atașamentul se dovedește a fi un document compatibil MS Word precum: „Salut, prieteni, cum merge treaba la noul loc de muncă? Aveți timp pentru a lua prânzul împreună mâine?” este tocmai ceea ce trebuie ca un utilizator de Linux să devină extrem de furios. De asemenea este o idee proastă să atașați semnătura dumneavoastră unui astfel de fișier. Dacă dorîți să semnați mesajele sau fișierele, folosiți GPG, GNU Privacy Guard compatibil PGP sau certificate SSL (Secure Socket Layer).

Acești utilizatori nu sunt furioși pentru că nu pot citi aceste documente sau pentru că sunt îngrijorați deoarece aceste formate generează, de regulă, fișiere mai mari, ci pentru că trebuie să folosească MS Windows sau pentru că trebuie să pornească alte programe.

6.3.3.2. Fișierele de configurare ale sistemului și ale utilizatorilor.

În capitolul care urmează veți începe să configurați mediul sistemului dumneavoastră și acest lucru include editarea a tot felul de fișiere care determină modul în care se comportă programele.

Nu editați aceste fișiere cu nici un component al suitelor de birou!

Specificațiile formatului implicit al fișierelor vor determina programul să adauge câteva linii de cod, care definesc formatul fișierului și fonturile utilizate. Aceste linii nu vor fi interpretate corect de către programele de care depind, generând astfel erori sau blocări ale programului care citește aceste fișiere. În anumite cazuri, puteți salva fișierul ca text simplu, dar veți avea probleme dacă acest lucru devine un obicei.

6.3.3.3. Dar eu doresc un editor grafic de text!

Dacă insistați, încercați **gedit**, **kedit**, **kwrite** sau **xedit**; aceste programe sunt doar pentru fișiere text, exact ceea ce aveți nevoie. Dacă însă dorîți să faceți ceva serios, atunci oprîți-vă la un editor propriu modului text, precum **vim** sau **Emacs**.

O alternativă acceptabilă este **gvim**, versiunea Gnome a lui **vim**. Tot va trebui să folosiți comenzi **vim**, dar dacă întâmpinați dificultăți, aceste comenzi sunt disponibile prin meniu.

6.4. Sumar

În acest capitol ați învățat cum să folosiți un editor. Cu toate că preferințele individuale primează în alegerea editorului de text, este necesar să cunoașteți cum se folosește cel puțin un editor de text.

Editorul **vi** este disponibil pe orice sistem UNIX.

Cele mai multe distribuții Linux includ o suită pentru birou și un editor grafic de text.

6.5. Exerciții

Acest capitol are doar un exercițiu: porniți tutorialul Vim prin introducerea comenzi **vimtutor** într-o sesiune terminal și parcurgeți-l.

Puteți porni, alternativ, **emacs** și tasteți **Ctrl + H** pentru a lansa tutorialul propriu al Emacs.

Practica este singura cale!

Capitolul 7. Home sweet /home

Acest capitol tratează configurarea mediului dumneavoastră de lucru. Acum, pentru că știți cum se folosește un editor, puteți modifica tot felul de fișiere pentru a vă simți mai bine acasă. După terminarea acestui capitol, veți ști mai multe despre:

- ◆ Organizarea mediului dumneavoastră
 - ◆ Fișierele comune de configurare ale consolei
 - ◆ Configurarea consolei
 - ◆ Configurarea prompterului
 - ◆ Configurarea mediului grafic
 - ◆ Aplicații legate de sunet și video
 - ◆ Administratorii de ecran și ferestre
 - ◆ Cum lucrează sistemul X client-server
 - ◆ Opțiunile de configurare pentru limbă și fonturi
 - ◆ Instalarea de programe noi
 - ◆ Actualizarea pachetelor existente
-

7.1. Păstrarea ordinii și curățeniei

7.1.1. Introducere

După cum am mai arătat, este foarte ușor să aduceți dezordine în sistem. Nu vom putea sublinia îndeajuns necesitatea păstrării curățeniei în sistem. Dacă învățați acest lucru de la început, veți căpăta o deprindere folositoare care vă va economisi timp în cazul în care veți programa pe sisteme Linux sau UNIX sau în cazurile în care va trebui să executați sarcini care țin de gestionarea sistemului. Vă prezentăm câteva cai prin care puteți să vă ușurați munca:

- Creați un director bin pentru fișierele programelor dumneavoastră și pentru scripturi.
 - Organizați fișierele ne-executabile în directoare potrivite și faceți cât de multe directoare credeti de cuvînță. Exemplele includ directoare separate pentru imagini, documente, proiecte, fișiere descărcate de pe web, foi de calcul, fișiere personale, și aşa mai departe.
 - Restrângeți permisiunile alocate directoarelor. Comanda **chmod 700 nume_director** vă ajută în acest sens.
 - Numiți sugestiv fișierele, precum Plângere către primul-ministru 050302, în loc de scrisoare1.
-

7.1.2 Faceți loc

Pe anumite sisteme, programul **quota** vă forțează să ștergeți date de pe disc din timp în timp sau limitele fizice ale discului fix vă determină să eliberați spațiu chiar dacă nu rulați nici un program de monitorizare. Această secțiune vă prezintă câteva moduri în care puteți face rost de spațiu pe disc, altele decât comanda **rm**.

Introduceți comanda **quota -v** pentru a afișa spațiul disponibil pe discul fix.

7.1.2.1. Golirea fișierelor

Câteodată, conținutul unui fișier mai prezintă interes, dar doriți să păstrați numele acelui fișier ca marker (de exemplu, aveți nevoie doar de data și ora la care a fost creat fișierul respectiv, doriți să știți că acel fișier a existat sau poate că aveți nevoie de acesta în viitor). Redirecționarea ieșirii unei comenzi nule este metoda prin care goliti un fișier în consolele Bourne și Bash:

```
andy:~> cat wishlist > placeholder
```

```

andy:~> ls -la placeholder
-rw-rw-r-- 1 andy andy 200 Jun 12 13:34 placeholder

andy:~> > placeholder

andy:~> ls -la placeholder
-rw-rw-r-- 1 andy andy 0 Jun 12 13:35 placeholder

```

Procesul prin care se reduce un fișier existent la un fișier cu același nume dar care conține 0 octeți este numit trunchiere.

Pentru a crea un fișier nou, gol, același efect îl obțineți prin comanda **touch**. În cazul unui fișier existent, comanda **touch** va actualiza doar data și ora la care a fost creat. Consultați paginile Info ale comenzi **touch** pentru mai multe detalii.

Pentru a goli parțial un fișier utilizați comanda **tail**. Să presupunem că lista de dorințe a utilizatorului *andi* devine din ce în ce mai lungă pentru că el adaugă noi dorințe dar nu șterge niciodată lucrurile pe care le-a obținut. Acum dorește să păstreze doar ultimele cinci poziții:

```

andy:~> tail -5 wishlist > newlist
andy:~> cat newlist > wishlist
andy:~> rm newlist

```

7.1.2.2. Mai multe despre fișierele jurnal

Anumite programe din Linux îți morțis să scrie tot felul de date într-un fișier jurnal. De regulă sunt configurate să scrie doar erorile sau o cantitate minimă de informații, cum ar fi revenirea dintr-o eroare. Chiar și în aceste condiții, informațiile conținute pot fi neinteresante pentru dumneavastră. Vă prezentăm câteva metode prin care puteți scăpa de aceste fișiere jurnal sau, cel puțin, limita mărimea lor:

- Încercați să ștergeți fișierele jurnal atunci când programul nu rulează, dacă sunteți sigur că nu vă trebuie. Unele programe, atunci când sunt repornite, constată lipsa fișierului jurnal și nu vor mai scrie date jurnaliere.
- Dacă ștergeți fișierul jurnal iar programul îl recreează, citiți documentația acelui program pentru a identifica opțiunea care spune programului să nu mai creeze fișiere jurnal.
- Încercați să realizați fișiere jurnal mai mici, prin configurarea scrierii informațiilor care sunt relevante pentru dumneavastră sau a celor cu adevărat importante.
- Încercați înlocuirea fișierelor jurnal cu legături simbolice către /dev/null; dacă aveți noroc, programul nu se va plângă. Nu faceți acest lucru pentru programele care rulează la pornirea sistemului sau a programelor care sunt rulate prin cron (consultați [Capitolul 4](#)). Aceste programe pot înlocui legătura simbolică cu un fișier mic, dar care va începe să crească în timp.

7.1.2.3. Poșta electronică

Curățați cu regularitate căsuța de poștă electronică (mailbox), creați subdirecțoare și redirecționați automat folosind **procmail** (consultați paginile Info) sau filtrele programului dumneavastră favorit de citire a mesajelor de poștă electronică. Dacă aveți un director pentru coșul de gunoi, goliți-l regulat.

Pentru redirecționarea mesajelor folosiți fișierul `.forward` din directorul personal. Serviciul de poștă electronică al Linux-ului cauță acest fișier de fiecare dată când trebuie să livreze mesaje. Conținutul fișierului spune serviciului de poștă electronică ce să facă cu mesajele dumneavastră. Poate conține o singură linie care definește o adresă de E-mail validă. În acest caz, sistemul va trimite toate mesajele la această adresă. De exemplu, când închiriați spațiu pentru un sait web, pentru a nu irosi spațiu pe disc, puteți redirecționa mesajele de poștă electronică adresate webmaster-ului către contul dumneavastră. În acest caz, directorul `.forward` poate arăta precum următorul exemplu:

```

webmaster@www ~/> cat .forward
mike@pandora.be

```

Folosirea redirecționării mesajelor este utilă și atunci când aveți mai multe adrese de poștă electronică, evitând în acest mod verificarea pe rând a tuturor acestor adrese. Puteți face ca fiecare adresă să înainteze mesajele către un singur cont centralizator, ușor accesibil.

Puteți cere administratorului de sistem să definească o redirecționare în fișierul local de mail, în același mod în care un cont este închis dar E-mail-ul rămâne activ pentru un timp.

7.1.2.4. Economisiți spațiu cu o legătură

Atunci când mai mulți utilizatori au nevoie de același fișier sau program, când numele fișierului original este prea lung sau greu de ținut minte, folosiți o legătură simbolică în loc de o copie separată pentru fiecare utilizator sau scop.

Legături simbolice multiple pot avea nume diferite, de exemplu o legătură se poate numi `fișierul_meu` într-un director personal al unui utilizator și `legătura_mea` în altul. Legături multiple (cu nume diferite) către același fișier pot exista chiar în același director. Acest lucru se întâmplă în directorul `/lib`: dacă introduceți comanda

`ls -l /lib`

vei observa că directorul este plin de legături care țințesc către același fișier. Ele sunt create pentru ca programele care caută un nume să nu înghețe, ele fiind direcționate către numele corect/current al librăriilor de care au nevoie.

7.1.2.5. Limitarea mărimii fișierelor

Consola dispune de o comandă proprie prin care puteți limita mărimea fișierelor, `ulimit`, care poate fi folosită și la afișarea limitărilor resurselor sistemului:

```
cindy:~$ ulimit -a
core file size (blocks)          0
data seg size (kbytes)          unlimited
file size (blocks)              unlimited
max locked memory (kbytes)      unlimited
max memory size (kbytes)        unlimited
open files                      1024
pipe size (512 bytes)           8
stack size (kbytes)              8192
cpu time (seconds)              unlimited
max user processes              512
virtual memory (kbytes)         unlimited
```

Cindy nu este dezvoltatoare de programe și nu-i pasă de fișierele care conțin informații pentru depanarea programelor (core dumps). Dacă doriți să beneficiați de informațiile conținute de aceste fișiere, puteți stabili mărimea lor prin comanda `ulimit`. Citiți paginile Info ale consolei (bash) pentru explicații detaliate.



Fișiere de depanare?

Un fișier de depanare sau *renunțare* (core file, core dump) este un fișier generat atunci când ceva nu merge aşa cum ar trebui în timpul executării unui program. El conține o copie a memoriei sistemului, aşa cum era în momentul în care a apărut eroarea.

7.1.2.6. Fișiere comprimate

Fișierele comprimate sunt folosite deoarece ele ocupă mai puțin spațiu pe discul fix. Alt avantaj este acela că aceste fișiere ocupă necesită o lățime de bandă mai mică în cazurile în care sunt transferate pe rețea. Multe fișiere, inclusiv paginile de manuale, sunt stocate într-un format comprimat în sistemul dumneavoastră. Însă despachetarea acestora pentru a consulta informațiile și comprimarea lor din nou sunt acțiuni care consumă timp. Nu doriți să despachetați o pagină de manual, pentru a citi despre opțiunea unei comenzi, apoi să comprimați din nou pagina man respectivă. Cei mai mulți oameni vor uita probabil să facă curățenie după ce găsesc informațiile de care au nevoie.

Există însă unele care acționează asupra fișierelor comprimate, despachetându-le doar în memorie. Fișierul comprimat de pe discul fix nu este modificat. Cele mai multe sisteme pot lucra cu **zgrep**, **zcat**, **bzless** și alți membri ai familiei z pentru prevenirea decomprimărilor și comprimărilor inutile. Consultați directorul binar al sistemului și paginile Info.

[Capitolul 9](#) tratează comprimarea fișierelor și vă oferă exemple despre construirea arhivelor.

7.2. Mediul dumneavoastră text

7.2.1 Variabilele de mediu

7.2.1.1. Generalități

Am prezentat deja câteva variabile de mediu, precum PATH și HOME. Până la acest moment, am prezentat exemple în care aceste variabile serveau anumitor scopuri ale consolei. Însă există alte utilități ale Linux-ului care solicită informații despre dumneavoastră pentru a-și face treaba.

Ce alte informații solicită programele, în afară de cele conținute de directoarele variabilei path și home?

Multe programe au nevoie să cunoască ce fel de terminal folosiți; această informație este stocată în variabila TERM. În modul text, este vorba despre emulatorul de terminal *linux*, în mediul grafic probabil că veți folosi *xterm*. Multe programe vor să cunoască care este editorul dumneavoastră favorit, în cazul în care trebuie să pornească un editor solicitat de un anume subprocess. Consola pe care o folosiți este stocată în variabila SHELL, tipul sistemului de operare în OS, și aşa mai departe. O listă a tuturor variabilelor definite pentru sesiunea dumneavoastră curentă poate fi obținută prin introducerea comenzi **printenv**.

Variabilele de mediu sunt gestionate de consolă. În opozitie cu variabilele obișnuite ale consolei, variabilele de mediu sunt moștenite de oricare program pe care îl porniți, inclusiv o altă consolă. O copie a acestor variabile este furnizată noilor procese, pe care o pot citi, modifică și transmite proceselor pe care ele le generează (procese-copil).

Nu este nimic special în numele variabilelor, cu excepția faptului că există convenția potrivit căreia cele comune sunt scrise cu litere mari. Puteti scorsi orice nume, deși există și variabile standard care sunt destul de importante pentru a se regăsi pe orice sistem Linux, precum PATH și HOME.

7.2.1.2. Exportarea variabilelor

Conținutul unei variabile individuale poate fi afișat prin comanda **echo**:

```
debby:~> echo $PATH  
/usr/bin:/usr/sbin:/bin:/sbin:/usr/X11R6/bin:/usr/local/bin  
  
debby:~> echo $MANPATH  
/usr/man:/usr/share/man/:/usr/local/man:/usr/X11R6/man
```

Dacă dorîți să schimbați conținutul unei variabile pentru a folosi celorlalte programe, trebuie să exportați noile valori din mediul dumneavoastră în mediul în care rulează acele programe. Un exemplu des întâlnit este exportarea variabilei PATH. Pentru a putea folosi în condiții optime simulatorul de zbor care se găsește în /opt/FlightGear/bin trebuie să declarați următoarele:

```
debby:~> PATH=$PATH:/opt/FlightGear/bin
```

Această linie dă instrucțiuni consolii să caute nu doar în variabila curentă, \$PATH, ci și în directorul /opt/FlightGear/bin.

Totuși, atât timp cât noua valoare a variabilei PATH nu este cunoscută mediului, programele nu vor lucra:

```
debby:~> runfgfs  
bash: runfgfs: command not found
```

Exportarea variabilelor se face prin comanda proprie consolei **export**:

```
debby:~> export PATH  
debby:~> runfgfs  
--flight simulator starts--
```

În consola Bash, puteți face acest lucru într-un singur pas:

```
export VARIABLE=value
```

Aceeași tehnică o puteți folosi în cazul variabilei MANPATH, care spune comenzi **man** unde să caute pentru paginile arhivate man. Dacă adăugați noi programe sistemului în directoare noi sau nepotrivate, documentația acestuia se poate găsi tot în aceste directoare. Dacă dorîți să citiți paginile de manual ale acestor programe, extindeți variabila MANPATH:

```
debby:~> export MANPATH=$MANPATH:/opt/FlightGear/man  
debby:~> echo $MANPATH  
/usr/man:/usr/share/man:/usr/local/man:/usr/X11R6/man:/opt/FlightGear/man
```

Puteți evita reintroducerea acestei comenzi în fiecare fereastră pe care o deschideți prin adăugarea ei unui fișier de configurare a consolei, consultați [Secțiunea 7.2.2.](#)

7.2.1.3. Variabile rezervate

Tabelul următor trece în revistă cele mai întâlnite variabile predefinite:

Tabelul 7-1. Variabile comune de mediu

Numele variabilei	Informația stocată
DISPLAY	Folosită de sistemul X Window pentru identificarea serverului grafic
DOMAIN	Numele domeniului
EDITOR	Stochează editorul dumneavoastră favorit
HISTSIZE	Dimensiunea fișierului arhivă al consolei, în număr de linii
HOME	Traseul către directorul dumneavoastră personal
HOSTNAME	Numele gazdei (calculatorului)
INPUTRC	Localizarea fișierului de definire a dispozitivelor de intrare, precum tastatura
LANG	Limba preferată
LD_LIBRARY_PATH	Traseul de căutare a librăriilor
LOGNAME	Numele de autentificare
MAIL	Localizarea directorului de poștă electronică
MANPATH	Traseul de căutare a paginilor de manuale
OS	Șir care descrie sistemul de operare
OSTYPE	Mai multe informații despre versiune, etc
PAGER	Folosit de programe ca man care au nevoie să știe ce trebuie să facă în cazul în care ieșirea generată este adresată la mai mult de o fereastră terminal
PATH	Caută traseele comenziilor
PS1	Prompterul primar
PS2	Prompterul secundar
PWD	Directorul de lucru curent

SHELL	Consola curentă
TERM	Tipul de terminal
UID	Identitatea utilizatorului
USER (NAME)	Numele utilizatorului
VIZUAL	Editorul dumneavoastră favorit în modul full-screen
XENVIRONMENT	Localizarea configurărilor personale ale comportamentului serverului X
XFILESEARCHPATH	Traseele de căutare ale librăriilor grafice

Multe alte variabile nu sunt doar predefinite, ci și preconfigurate, prin fișierele de configurare. Le vom discuta în secțiunea următoare.

7.2.2. Fișierele de configurare ale consolei

Atunci când introduceți comanda **ls -al** pentru a obține lista lungă (cuprinzătoare) a tuturor fișierelor, inclusiv a celor care încep cu un punct, din directorul dumneavoastră personal, veți observa unul sau mai multe fișiere care încep cu un punct (.) și se termină în *rc*. În cazul consolei **bash** acesta este *.bashrc*. Acest fișier este oglinda fișierului de configurare lărgită a sistemului */etc/bashrc*.

La autentificarea într-o consolă interactivă, **login** va realiza autentificarea, va configura mediul și va porni consola. În cazul **bash**, pasul următor este citirea profilului general *profile* din */etc*, dacă acest fișier există. **bash** cauță apoi fișierele *~/.bash_profile*, *~/.bash_login* și *~/.profile*, în această ordine, și citește și execută comenziile din primul fișier care le conține, dacă sunt valide. Dacă ele nu există în nici un fișier, se aplică cele din */etc/bashrc*.

Când consola primește comanda de ieșire, **bash** citește și execută comenziile din fișierul *~/.bash_logout*, dacă există.

Procedura este explicată în detaliu în paginile man ale **login** și **bash**.

7.2.3. Înfățișarea tipică a fișierelor de configurare

7.2.3.1. Exemplul */etc/profile*

Vom arunca o privire în cele ce urmează asupra câtorva dintre aceste fișiere de configurare. Primul este citit */etc/profile*, în care sunt configurate variabile importante precum *PATH*, *USER* și *HOSTNAME*:

```
debby:~> cat /etc/profile
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# Path manipulation
if [ `id -u` = 0 ] && ! echo $PATH | /bin/grep -q "/sbin" ; then
    PATH=/sbin:$PATH
fi

if [ `id -u` = 0 ] && ! echo $PATH | /bin/grep -q "/usr/sbin" ; then
    PATH=/usr/sbin:$PATH
fi

if [ `id -u` = 0 ] && ! echo $PATH | /bin/grep -q "/usr/local/sbin"
    then
        PATH=/usr/local/sbin:$PATH
fi

if ! echo $PATH | /bin/grep -q "/usr/X11R6/bin" ; then
    PATH="$PATH:/usr/X11R6/bin"
fi
```

Aceste linii marchează configurarea traseelor: dacă *root* deschide o consolă (ID-ul de utilizator 0), se verifică dacă /sbin, /usr/sbin și /usr/local/sbin sunt în variabila PATH. Dacă nu, ele sunt adăugate. Pentru orice utilizator se verifică dacă /usr/X11R6/bin se regăsește în trasee.

```
# No core files by default  
ulimit -S -c 0 > /dev/null 2>&1
```

Tot cunoit merge către /dev/null dacă utilizatorul nu schimbă această cale.

```
USER=`id -un`  
LOGNAME=$USER  
MAIL="/var/spool/mail/$USER"  
  
HOSTNAME=`/bin/hostname`  
HISTSIZE=1000
```

Aici le sunt alocate variabilelor generale valorile potrivite.

```
if [ -z "$INPUTRC" -a ! -f "$HOME/.inputrc" ]; then  
    INPUTRC=/etc/inputrc  
fi
```

Dacă variabila INPUTRC nu este configurată și nu există nici un fișier .inputrc în directorul personal al utilizatorului, este încărcat fișierul implicit de control.

```
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE INPUTRC
```

Toate variabilele sunt exportate, pentru a fi disponibile programelor care au nevoie de informații despre mediul dumneavoastră.

7.2.3.2. Directorul profile.d

```
for i in /etc/profile.d/*.sh ; do  
    if [ -r $i ]; then  
        . $i  
    fi  
done  
unset i
```

Toate scripturile shell valide din directorul /etc/profile.d sunt citite și executate. Ele fac lucruri precum activarea *color-ls*, atribuirea aliasului **vi** lui **vim**, configurarea localizării, și altele. Variabila temporară *i* nu este luată în considerare pentru a preveni alterarea comportamentului consolei.

7.2.3.3. Exemplul .bash_profile

Apoi **bash** caută în directorul personal al utilizatorului fișierul .bash_profile:

```
debby:~> cat .bash_profile  
#####  
#  
#   .bash_profile file  
#  
#   Executed from the bash shell when you log in.  
#  
#####  
  
source ~/.bashrc  
source ~/.bash_login
```

Acest fișier de direcționare dă instrucțiuni consolei să citească prima dată ~/.bashrc și apoi ~/.bash_login. Veți întâlni comanda proprie consolei **source** cu regularitate dacă lucrați în mediul consolei: este folosită la modificarea configurației mediului curent.

7.2.3.4. Exemplul .bash_login

Fișierul `~/.bash_login` definește nivelul implicit de protecție prin configurarea valorii **umask**, consultați [Sectiunea 3.4.2.2](#). Fișierul `~/.bashrc` este folosit pentru definirea mai multor aliasuri și funcții specifice utilizatorului precum și a variabilelor personale de mediu. Este citit în primul rând `/etc/bashrc`, care descrie prompterul implicit (`PS1`) și valoarea implicită `umask`. După aceasta puteți adăuga propriile configurații. Dacă nu există fișierul `~/.bashrc`, este citit implicit `/etc/bashrc`.

7.2.3.5. Exemplul /etc/bashrc

Fișierul dumneavoastră `/etc/bashrc` ar putea arăta precum acesta:

```
debby:~> cat /etc/bashrc
# /etc/bashrc

# System wide functions and aliases
# Environment stuff goes in /etc/profile

# by default, we want this to get set.

# Even for non-interactive, non-login shells.
if [ `id -gn` = `id -un` -a `id -u` -gt 99 ]; then
    umask 002
else
    umask 022
fi
```

Aceste linii configurează valoarea **umask**. Apoi, în funcție de tipul consolei pe care o folosiți, este configurația prompterului:

```
        stty erase `tput kbs`
    fi
    fi
    fi
case $TERM in
    xterm*)
        if [ -e /etc/sysconfig/bash-prompt-xterm ]; then
            PROMPT_COMMAND=/etc/sysconfig/bash-prompt-xterm
        else
            PROMPT_COMMAND='echo -ne "\033]0;${USER}@${HOSTNAME%%.*}:\
${PWD/$HOME/~}\007"'
        fi
    ;;
    *)
        [ -e /etc/sysconfig/bash-prompt-default ] && PROMPT_COMMAND=\
/etc/sysconfig/bash-prompt-default
    ;;
esac
[ "$PS1" = "\s-\v\\\$ " ] && PS1="\u@\h \W]\\$ "

if [ "x$SHLVL" != "x1" ]; then # We're not a login shell
    for i in /etc/profile.d/*.sh; do
        if [ -x $i ]; then
            . $i
        fi
    done
fi
```

7.2.3.6. Exemplul .bash_logout

La deautentificare, sunt executate comenzi din `~/.bash_logout`, care pot, de exemplu, curăța terminalul, pentru ca dumneavoastră să aveți o fereastră goală la ieșirea dintr-o sesiune la distanță sau la părăsirea consolei sistemului:

```
debby:~> cat .bash_logout
# ~/.bash_logout

clear
```

Modul în care lucrează aceste scripturi vă este prezentat în secțiunea care urmează. Trebuie să vă spunem că veți avea nevoie de comanda **info bash**.

7.2.4. Prompterul Bash

7.2.4.1. Introducere

Prompterul Bash poate face mai multe lucruri decât să afișeze informațiile simple ca numele de utilizator, numele calculatorului și indicațiile despre directorul de lucru curent. Puteți adăuga alte informații, precum data și ora curentă, numărul utilizatorilor autentificați, etc.

Înainte de a începe, totuși, faceți o copie de rezervă a prompterului curent într-altă variabilă de mediu:

```
[jerry@nowhere jerry]$ MYPROMPT=$PS1

[jerry@nowhere jerry]$ echo $MYPROMPT
[\u@\h \W]\$

[jerry@nowhere jerry]$
```

În cazul în care schimbați prompterul, de exemplu prin comanda **PS1="->"**, puteți recăpăta oricând prompterul original dacă introduceți comanda **PS1=\$MYPROMPT**. Îl veți avea înapoi, de asemenea, la reautentificare, atât timp cât v-ați jucat cu prompterul doar în linia de comandă, evitând scrierea lui într-un fișier de configurare al consolei.

7.2.4.2. Câteva exemple

Pentru a înțelege aceste promptere și secvența de ieșire folosită, apelați la paginile Info sau man.

- **export PS1="\t \j"**

Afișează ora și numărul sarcinilor care rulează.

- **export PS1="\d \u@\h \w : "**

Afișează data, numele utilizatorului, al calculatorului și directorul curent de lucru. Opțiunea \W afișează doar numele de bază al directorului curent.

- **export PS1="\! "**

Afișează date despre folosirea fiecărei comenzi.

- **export PS1="\[\033[1;35m\] \u@\h \[\033[0m\] "**

Afișează șirul care conține numele utilizatorului și cel al calculatorului în roz.

- **export PS1="\[\033[1;35m\] \u \[\033[0m\] \[\033[1;34m\] \w \[\033[0m\] "**

Afișează numele utilizatorului în roz și directorul de lucru curent în albastru.

- **export PS1="\[\033[1;35m\] \u \[\033[0m\] \[\033[1;34m\] \w \[\033[0m\] "**

Prompter pentru persoanele care au dificultăți în remarcarea diferențelor dintre prompter și textul introdus de la tastatură.

- **export PS1="\[\033[4;34m\] \u@\h \w \[\033[0m\] "**

Prompter subliniat.

- **export PS1="\[\033[7;34m\] \u@\h \w \[\033[0m\] "**

Caractere albe pe fundal albastru.

● **export PS1="[\033[3;35m]\u@\[\w \ [\033[0m]\a"**

Prompter roz care își schimbă strălucirea pentru a vă atenționa despre terminarea comenzi.

● **export PS1=...**

Variabilele trebuie exportate, pentru ca mediul de lucru să fie cunoscut de comenzi ce vor fi executate. Dacă dorîți să păstrați o configurație particulară a prompterului, cel mai bine este ca această configurație să fie pusă în fișierul de configurare al consolei, ~/.bashrc.

Dacă dorîți, prompterele pot executa scripturi shell și se pot comporta diferit în condiții deosebite. Puteți configura prompterul să cânte o melodie de fiecare dată când introduceți o comandă, chestie de care însă vă veți plăcisi repede... Mai multe informații pot fi găsite în [Bash-Prompt HOWTO](#).

7.2.5. Scripturi shell

7.2.5.1. Ce sunt scripturile?

Un script shell este, după cum ați observat din exemplele care trătau configurarea consolei, un fișier text care conține comenzi pentru aceasta. Când este folosit un asemenea fișier ca prim argument fără opțiuni ce invocă Bash și nu sunt prezente nici opțiunile -c sau -s, Bash citește și execută comenzi din fișier și apoi se închide. Acest mod de operare creează o consolă non-interactivă. Când Bash rulează un script de shell, asociază parametrul special 0 numelui de fișier, în detrimentul numelui consolei, iar parametrul de poziționare (tot ceea ce urmează numelui scriptului) este asociat argumentelor disponibile, dacă sunt furnizate. Dacă nu sunt furnizate argumente adiționale, parametrii de poziționare rămân nestabiliți.

Un script de shell poate fi făcut executabil folosind comanda **chmod** care determină luarea în considerare a bit-ului care marchează scriptul ca executabil. Atunci când Bash găsește un astfel de fișier în timp ce caută comenzi în PATH, deschide o subconsolă pentru rularea acestuia. Cu alte cuvinte, executarea

nume_fișier ARGUMENTE

este echivalentă cu executarea

bash nume_fișier ARGUMENTE

dacă „nume_fișier” este un script shell executabil. Această subconsolă se reinitializează, astfel că efectul este același cu invocarea unei noi console care să interpreteze scriptul, cu excepția că localizarea comenziilor executate de consola părinte este reținută și de consola copil (citiți despre **hash** în paginile Info).

Cele mai multe versiuni de UNIX includ acest lucru în mecanismul de executare a comenziilor propriu sistemului de operare. Dacă prima linie a unui script începe cu două caractere „#!”, restul liniei specifică interpreterul aceluia program. Astfel, puteți specifica **bash**, **awk**, **perl** sau alt interpreter sau consolă și puteți scrie restul scriptului în acel limbaj de programare.

Argumentele pentru interpreter constau de fapt într-un singur argument optional poziționat după numele interpreterului, în prima linie a scriptului, urmat de restul argumentelor. Bash realizează aceste acțiuni pe sistemele care nu gestionează ele însele acest gen de activități. Scripturile Bash încep, de regulă, cu

```
#! /bin/bash
```

(presupunem aici că Bash este instalată în /bin), deoarece această expresie spune că acel script va fi interpretat de Bash, chiar dacă este executat într-o altă consolă.

7.2.5.2. Câteva exemple simple

Înălță un script foarte simplu, care constă într-o singură comandă, care spune hello utilizatorului care-l execută:

```
[jerry@nowhere ~] cat hello.sh
#!/bin/bash
echo "Hello $USER"
```

Se observă că scriptul conține doar comanda **echo**, care folosește valoarea variabilei de mediu (\$USER pentru a afișa un sir personalizat pentru fiecare utilizator care îl rulează.

Vă prezentăm un altul, tot de o singură linie, folosit pentru afișarea utilizatorilor autentificați:

```
#!/bin/bash  
who | cut -d " " -f 1 | sort -u
```

Mai jos este un script care conține mai multe linii folosit pentru realizarea de copii ale tuturor fișierelor dintr-un director. Scriptul listează mai întâi toate fișierele din directorul curent și pune această listă în variabila LIST. Apoi stabilește numele copiei fiecărui fișier, copiind în cele din urmă fișierele. Este afișat un mesaj pentru fiecare fișier prelucrat:

```
tille:~> cat bin/makebackupfiles.sh  
#!/bin/bash  
# make copies of all files in a directory  
LIST=`ls`  
for i in $LIST; do  
    ORIG=$i  
    DEST=$i.old  
    cp $ORIG $DEST  
    echo "copied $i"  
done
```

Prin simpla introducere a unei linii ca **mv **.old** nu veți obține nici un rezultat, lucru ușor de verificat pe câteva fișiere de test. Comanda **echo** a fost adăugată pentru afișarea desfășurării activităților. **echo** este folositoare în general în cazurile în care un script nu lucrează cum trebuie: introduceți câte una după fiecare pas despre care aveți dubii și veți afla eroarea imediat.

Directorul /etc/rc.d/init.d conține multe exemple de scripturi. Vă prezentăm pe acela care controlează serverul fictiv ICanSeeYou:

```
#!/bin/sh  
# description: ICanSeeYou allows you to see networked people  
  
# process name: ICanSeeYou  
# pidfile: /var/run/ICanSeeYou/ICanSeeYou.pid  
# config: /etc/ICanSeeYou.cfg  
  
# Source function library.  
. /etc/rc.d/init.d/functions  
  
# See how (with which arguments) we were called.  
case "$1" in  
    start)  
        echo -n "Starting ICanSeeYou: "  
        daemon ICanSeeYou  
        echo  
        touch /var/lock/subsys/ICanSeeYou  
        ;;  
    stop)  
        echo -n "Shutting down ICanSeeYou: "  
  
        killproc ICanSeeYou  
        echo  
        rm -f /var/lock/subsys/ICanSeeYou  
        rm -f /var/run/ICanSeeYou/ICanSeeYou.pid  
        ;;  
    status)  
        status ICanSeeYou  
        ;;  
    restart)  
        $0 stop  
        $0 start  
        ;;  
    *)  
        echo "Usage: $0 {start|stop|restart|status}"  
        exit 1  
esac  
  
exit 0
```

În primul rând, prin comanda . (punct) sunt încărcate anumite funcții ale consolei, folosite de aproape fiecare script din /etc/rc.d/init.d. Apoi este executată comanda **case**, care definește patru moduri în care poate fi executat scriptul. Unul dintre acestea poate fi **I Can See You Start**. Alegerea unuia din aceste moduri se face prin citirea argumentului (prim) al scriptului, cu expresia \$1.

Dacă nu se regăsește nici o intrare, este aplicat cazul implicit, cel marcat cu un asterisc, situație în care scriptul generează un mesaj de eroare. Listarea **case** este încheiată cu declarația **esac**. În cazul **start** programul este pornit ca serviciu, fiindu-i alocat un ID de proces și blocat (încuiat). În cazul **stop**, procesul care a fost pornit este căutat și opus, înlăturându-se blocajul (încuietoarea) și PID-ul. Opțiunile, ca, de exemplu, opțiunea **daemon** și **funcțiile**, ca, de exemplu, **killproc**, sunt definite în fișierul /etc/rc.d/init.d/functions. Configurația prezentată este specifică distribuției alese pentru acest exemplu. Scripturile rulate la inițierea sistemului dumneavoastră pot folosi alte funcții, definite în alte fișiere, sau nici unul.

Scriptul returnează codul de ieșire 0 către părintele său dacă a rulat fără probleme.

Acest script este un exemplu foarte bun în privința folosirii funcțiilor, care fac ca scriptul să fie citit mai ușor, munca fiind terminată mai repede. Observați că a fost folosit **sh** în loc de **bash**, pentru ca acest script să fie folositor pe mai multe sisteme. Într-un sistem Linux, rularea **bash** ca **sh** are ca rezultat o consolă care lucrează în modul compatibil POSIX.

Paginile de manuale referitoare la **bash** conțin multe informații despre combinarea comenziilor, ciclurile **for**- și **while**- și expresiile regulate, precum și exemple. Un curs de bază despre Bash, pentru administratorii de sisteme și utilizatori care doresc mai multe puteri în relația cu sistemul, scris de autoarea prezentului ghid, poate fi găsit la <http://telle.garrels.be/training/bash/>. Descrieri detaliate ale facilităților Bash și aplicații ale acestora puteți găsi în ghidul [Advanced Bash Scripting](#).

7.3. Mediul grafic

7.3.1. Introducere

Utilizatorului de nivel mediu poate să nu-i pese foarte mult de configurările mediului de lucru, dar Linux-ul oferă o varietate însemnată de ferestre și administratori ai spațiului de lucru (desktop managers) care sunt folosiți în mediul X, mediul grafic. Folosirea și configurarea gestionarelor de ferestre și a spațiilor de lucru este directă și poate fi asemănătoare cu spațiile de lucru standard MS Windows, Apple sau UNIX CDE, deși mulți utilizatori de Linux preferă spații de lucru și administratori de ferestre mai elaborați. Nu vom discuta aici configurările specifice utilizatorilor. Dumneavoastră puteți experimenta după citirea documentației, folosind funcțiile integrate de ajutor care sunt integrate în aceste gestionare și vă asigurăm că vă veți descurca de minune. Totuși, noi vă vom prezenta sistemul în linii generale.

7.3.2. Sistemul X Window

Sistemul X Window este un sistem de tip client-server care rulează pe o gamă largă de calculatoare și dispozitive grafice. Serverul X Window rulează pe calculatoare cu grafică tip hartă de pixeli (imagină raster – bitmap). Serverul X distribuie intrările de la utilizatori și acceptă solicitările ieșirilor generate de mai multe programe client prin mai multe canale de comunicație interprocese. Deși cazul cel mai des întâlnit este acela în care programele client rulează pe aceeași calculatoare cu serverul, ele pot rula foarte bine și pe alte calculatoare (inclusiv calculatoare cu arhitecturi și sisteme de operare diferite). Vom prezenta aceste aspecte în [Capitolul 10](#), în care tratăm rețele și aplicații la distanță.

Sistemul X integrează suprapunerea ierarhică a sub-ferestrelor, operațiilor text și grafice, atât în sistemele monocrome cât și în cele color. Numărul programelor client X care folosesc serverul grafic X este foarte mare. Iată câteva din programele furnizate ca parte a distribuției Consorțiului X:

- **xterm**: un emulator de terminal
- **twm**: un gestionar de ferestre minimalist
- **xdm**: administrator de ecran
- **xconsole**: un program de redirectare a consolei
- **bitmap**: editor pentru harta de pixeli

- **xauth, xhost și iceauth**: programe pentru controlul accesului
- **xset, xmodmap** precum și altele de acest tip: programe pentru configurații personale
- **xclock**: un ceas
- **xlsfonts** și altele: un afișor de fonturi, utilități care listează informații despre fonturi, ferestre și ecrane
- **xfs**: server pentru fonturi
- ...

Vă îndrumăm din nou către paginile de manuale ale acestor comenzi pentru informații detaliate. Mai multe explicații asupra funcțiilor disponibile puteți găsi în manualul conținut de distribuția dumneavoastră X, *Xlib – C language X Interface*, în specificațiile protocolului Sistemului de ferestre X și în alte manuale ale componentelor acestui sistem grafic. Directorul `/usr/share/doc` conține documentație asupra acestui subiect.

Multe alte programele, administratori de ferestre, jocuri, unelte și altele sunt incluse în distribuțiile Consorțiuului X ca programe realizate de utilizatori contributori sau se găsesc pe saiturile anonime FTP. Locuri în care puteți începe căutările sunt <http://www.x.org> și <http://www.xfree.org>.

Mai mult, toate aplicațiile dumneavoastră grafice, precum navigatorul de internet, programul de poștă electronică, programul de vizualizare a fotografiilor, uneltele pentru redarea sunetului și aşa mai departe, sunt clienti ai serverului X. În cazul unei utilizări obișnuite a modului grafic, clientii X și serverul X se află pe același calculator.

7.3.2.1. Numele ecranelor

Din punctul de vedere al utilizatorului, fiecare server X are un nume de ecran în forma:

gazdă: numărul afișajului.numărul ecranului

Această informație este folosită de aplicații pentru determinarea modului în care se conectează la serverul X și ce ecran vor folosi implicit (pe sistemele cu mai multe monitoare):

- **numele gazdei**: numele gazdei este numele calculatorului client la care ecranul este conectat fizic. Dacă acest nume nu este dat, este folosită cea mai eficientă modalitate de comunicare cu un serviciu de pe același calculator.
- **numărul afișajului**: cuvântul „afișaj” este folosit pentru a descrie un grup de monitoare care împart aceeași tastatură și dispozitive indicatoare (maus, tabletă, etc.). Cele mai multe stații de lucru au, în mod normal, o singură tastatură și un singur monitor. Sistemele mai mari, multi-utilizatori, au totuși mai multe afișaje, pentru ca lucrul în modul grafic să fie posibil pentru mai multe persoane. Pentru evitarea situațiilor confuze, fiecare afișaj al acestui tip sistem are asociat un **număr de afișaj** (care începe cu 0) la pornirea serverului X. Numărul afișajului trebuie să fie mereu prezent în numele acestuia.
- **numărul de ecran**: anumite afișaje împart o singură tastatură și un singur dispozitiv indicator la două sau mai multe monitoare. Întrucât fiecare dintre ele are propriul set de ferestre, fiecărui ecran îi este dat un **număr de ecran** (care pornește de la 0) în momentul pornirii serviciului grafic. Dacă acest număr nu este alocat, va fi folosit afișajul 0.

Pe sistemele POSIX, numele ecranului implicit este stocat în variabila de mediu DISPLAY. Această variabilă este configurată automat de către emulatorul de terminal **xterm**. În cazul conectării la alt calculator, prin rețea, poate fi necesar să configurați această variabilă manual pentru a indica afișajul dumneavoastră, consultați [Sectiunea 10.4.3.2](#).

Mai multe informații pot fi găsite în paginile de manual ale serverului X.

7.3.2.2. Administratorii de ferestre și de ecran

Afișarea ferestrelor pe ecranul monitorului este controlată de programe speciale, numite **administratori de ferestre**. Deși mulți administratori de ferestre se vor descurca cu specificațiile geometrice existente, alții pot ignora aceste specificații (ele vor solicita utilizatorului desenarea explicită a regiunii ferestrei pe ecran, folosind un dispozitiv indicator, de exemplu).

Întrucât administratorii de ferestre sunt, în general (deși complexe), programe client, pot fi construite numeroase interfețe cu utilizatorul. Distribuția Consorțiului X conține un administrator de ferestre numit **twm**, dar mulți utilizatori preferă ceva mai sofisticat, atunci când configurația sistemelor permite acest lucru. Sawfish și Enlightenment sunt exemple apreciate, care permit fiecărui utilizator să dispună de un spațiu de lucru în concordanță cu stilul și personalitatea sa.

Un administrator de ecran (spațiu de lucru) folosește un administrator de ferestre pentru a aranja spațiul de lucru grafic într-un mod convenabil, cu bare de meniu, meniuri contextuale, mesaje informative, ceas, administrator de programe, administrator de fișiere, etc. Printre cei mai cunoscuți administratori de ecran menționăm Gnome și KDE, care rulează pe aproape orice distribuție de Linux și pe multe sisteme UNIX.



Aplicații KDE în Gnome/aplicații Gnome în KDE

Nu este necesar să porniți spațiul de lucru în KDE pentru a rula aplicațiile specifice KDE. Dacă aveți instalate librăriile KDE (pachetul kdelibs), puteți rula aceste aplicații din meniul Gnome sau dintr-un terminal Gnome.

Rularea aplicațiilor Gnome în mediul KDE este puțin mai deosebită, deoarece nu există un singur set de librării de bază ale Gnome-ului. Totuși, prin instalarea dependințelor și a unor pachete extra, care vor fi cerute, veți putea face acest lucru.

7.3.3. Configurarea serverului X

Distribuția care este folosită de obicei în Linux, *Xfree86*, folosește fișierul de configurare *XF86Config* pentru configurarea implicită. Acest fișier configurației placa video și poate fi găsită în mai multe locuri, cel mai des fiind */etc/X11*.

Dacă observați că fișierul */etc/X11/XF86Config* este prezent în sistemul dumneavoastră, descrierea completă a acestuia poate fi găsită în paginile Info sau man ale *XF86Config*.

Din cauza aspectelor legate de licențiere în cazul *Xfree86*, sistemele mai noi vin cu distribuția de server X și unelte pentru acesta a *X.Org*. Fișierul principal de configurare este *xorg.conf*, pe care-l puteți găsi în */etc/X11*. Fișierul conține o serie de secțiuni care pot fi dispuse în orice ordine. Aceste secțiuni conțin informații despre monitor, placă video, configurațiile afișajului, tastatură, etc. Ca utilizator, nu trebuie să vă bateți capul prea mult cu ceea ce este în acest fișier, întrucât este configurat la momentul instalării sistemului.

Dacă trebuie să schimbați configurațiile serverului grafic, puteți rula uneltele de configurare sau edita fișierele de configurare care mențin infrastructura serverului *XFree86*. Consultați paginile man pentru mai multe informații; în unele cazuri, distribuția folosită de dumneavoastră are propriile unelte de configurare. Deoarece o configurație greșită poate duce la afișarea, în modul grafic, a unor gunoaie de neînțeles, vă sfătuim să realizați o copie de siguranță a fișierelor de configurare pe care doriti să le editați, pentru a putea fi restaurate.

7.4. Configurații regionale specifice

7.4.1. Configurarea tastaturii

Configurarea modelului de tastatură se face prin comanda **loadkeys** în cazul consolii text. Folosiți una altă specifică distribuției dumneavoastră sau editați secțiunea *Keyboard (Tastatură)* din *XF86Config* (sau *xorg.conf* cum ar fi în cazul Ubuntu^{n.t.}), pentru configurația tastaturii pentru modul grafic. În configurația implicită, este posibil să aveți configurații următorul caz:

```
XkbLayout      "us"
```

După cum am spus, acesta ar putea fi cazul implicit. Schimbarea aspectului tastaturii în concordanță cu regiunea în care vă aflați se face prin înlocuirea valorii dintre ghilimele cu oricare alta listată în subdirectoarele directorului *keymaps*. Dacă nu-l puteți găsi, încercați afișarea de către sistem a locului acestuia prin comanda următoare:

locate keymaps

Este posibilă combinarea aspectului tastaturii, după cum vă prezintă următorul exemplu:

```
Xkblayout      "us, ru"
```

Faceți o copie a fișierului /etc/X11/XF86Config (sau, după cum am mai arătat, a /etc/X11/xorg.conf) înainte de a-l edita! Trebuie să folosiți contul de root pentru acest lucru.

Deconectați-vă și autentificați-vă din nou pentru a folosi noile configurări ale serverului X.

Applet-ul pentru tastatură din Gnome asigură schimbarea imediată a aspectului tastaturii; nu este nevoie de permisiuni speciale pentru folosirea acestui program. KDE are o unealtă similară pentru schimbarea aspectului tastaturii.

7.4.2. Fonturi

Folosiți unealta **setfont** pentru a încărca fonturile în mediul text. Cele mai multe sisteme vin cu un fișier standard **inputrc** care permite combinarea caracterelor, cum ar fi franțuzescul "é" (metacaractere). Administratorul de sistem ar trebui să adauge linia

```
export INPUTRC="/etc/inputrc"
```

la fișierul /etc/bashrc.

7.4.3. Data și zona de timp

Configurarea informațiilor legate de timp se face la instalare. După aceea, pot fi actualizate folosind un client **NTP** (Network Time Protocol). Cele mai multe sisteme Linux folosesc implicit **ntpd**:

```
debby:~> ps -ef | grep ntpd
ntp      24678      1  0 2002 ?        00:00:33 ntpd -U ntp
```

Puteți rula **ntpdate** manual pentru configurarea ceasului, dacă, evident, puteți accesa un server de timp. Serviciul **ntpd** nu trebuie să ruleze atunci când folosiți **ntpdate** pentru ajustarea ceasului. Folosiți un server de timp ca argument al comenzii:

```
root@box:~# ntpdate 10.2.5.200
26 Oct 14:35:42 ntpdate[20364]: adjust time server 10.2.5.200 offset
-0.008049 sec
```

Consultați manualul sistemului dumneavoastră și documentația conținută de pachetul NTP. Cei mai mulți administratori ai spațiului de lucru au unelte proprii pentru configurarea timpului sistemului, cu solicitarea privilegiilor speciale.

Pentru configurarea corectă a zonei de timp, puteți folosi comenziile **tzconfig** sau **timezone**. Zona de timp este și ea configurață la instalarea sistemului pe calculatorul dumneavoastră. Multe sisteme au unelte specifice distribuției pentru configurarea acestor opțiuni, aşa că vă îndreptăm către documentația specifică sistemului pe care îl folosiți.

7.4.4 Limba

Dacă primiți mesaje de la sistem în olandeză sau franceză, și nu dorîți acest lucru, vă sfătuim să configurați variabilele **LANG** și **LANGUAGE**, acestea asigurând infrastructura pentru limba dorită, configurările locale și fonturile legate de caracterele speciale ale respectivei limbi.

În cazul celor mai multe ecrane grafice de autentificare, cum ar fi **gdm** sau **kdm**, aveți posibilitatea de a configura limba folosită în sistem înainte de autentificare.

Pe cele mai multe sisteme, en_US.UTF-8 trebuie să fie alegerea implicită în zilele noastre. Acest

lucru nu reprezintă o problemă, deoarece sistemele pe care această configurare este implicită, vin cu toate programele pentru această codare. Astfel, **vi** poate edita toate fișierele din sistem, **cat** nu se va comporta ciudat, etc.

Necazurile încep atunci când vă conectați la un sistem mai vechi care nu se descurcă cu acest sistem de codare, sau atunci când deschideți un fișier codat în *UTF-8* pe un sistem care înțelege numai caractere ale fonturilor pe 1 bit. Utilitarul **recode** se poate dovedi folositor în aceste cazuri, cu ajutorul lui fiind posibilă conversia dintr-un set de caractere în altul. Citiți paginile man pentru o vedere de ansamblu a facilităților care le aveți la îndemână. O altă soluție ar fi lucrul temporar cu o altă definiție a codării, prin configurația variabilei **LANG**:

```
debby:~> acroread /var/tmp/51434s.pdf
Warning: charset "UTF-8" not supported, using "ISO8859-1".
Aborted

debby:~> set | grep UTF
LANG=en_US.UTF-8

debby:~> export LANG=en_US

debby:~> acroread /var/tmp/51434s.pdf
<--new window opens-->
```

Consultați saitul [Mozilla](#) pentru a afla cum puteți avea Firefox în limba dumneavoastră. Saitul [OpenOffice.org](#) are informații despre localizarea suitei de birou OpenOffice.

7.4.5. Informații specifice pentru alte țări

[Lista Cum să](#) următoare cuprinde referințe despre localizarea în limbile bangla, bielorusă, chineză, esperanto, finlandeză, franceză, greacă, lituaniană, poloneză, sârbă, slovacă, slovenă, spaniolă, thai și turcă.

7.5. Instalarea de noi programe

7.5.1. Generalități

Foarte mulți oameni sunt surprinși de faptul că, după instalarea Linux-ului, au un calculator care rulează bine, și, mai mult decât atât, gata de folosire; cele mai multe distribuții conțin drivere pentru plăcile video sau cele de rețea, monitoare și alte dispozitive externe, astfel încât, de obicei, nu mai trebuie să instalați alte drivere. De asemenea, unele comune, cum ar fi suite pentru birou, navigatoare de Internet, clienți de poștă electronică și alți clienți de rețea sunt incluse în distribuțiile principale. Chiar și aşa, o instalare implicită poate să nu acopere necesitățile dumneavoastră.

Dacă nu puteți găsi ceea ce aveți nevoie, poate că acel lucru nu este instalat în sistemul dumneavoastră. Sau poate că aveți programul necesar, dar acesta nu face ceea ce ar trebui să facă. Amintiți-vă că Linux-ul progresează rapid iar programele sunt îmbunătățite zi de zi. Nu vă irosiți timpul căutând rezolvarea unor probleme pentru care au muncit alții.

Vă puteți actualiza sistemul și adăuga pachete oricând doriti. Cele mai multe programe vin în pachete. Programe în plus față de instalarea implicită pot fi găsite pe CD-ul de pe care ați instalat sistemul sau pe Internet. Saitul web al distribuției dumneavoastră este un loc bun în care puteți începe căutarea de programe adiționale care conțin și instrucțiunile despre cum se face această instalare pe tipul de Linux pe care-l folosiți, consultați [Appendix A](#). Întotdeauna citiți documentația care este conținută de programele noi, precum și orice indicații referitoare la instalare, conținute de acel pachet. Toate programele vin cu un fișier **README**, pe care vă sfătuim să-l citiți neapărat.

7.5.2. Tipuri de pachete

7.5.2.1. Pachete RPM

7.5.2.1.1. Ce este RPM?

RPM, Administratorul de pachete RedHat, este un administrator de pachete puternic pe care-l puteți folosi la instalarea, actualizarea și ștergerea pachetelor de programe. Vă permite să căutați pachete și să urmăriți fișierele care sunt incluse de fiecare pachet. Are conținut un sistem de verificare a autenticității pachetelor descărcate de pe Internet. Utilizatorii avansați pot să-și construiască propriile pachete cu ajutorul RPM.

Un pachet RPM consistă într-o arhivă de fișiere și meta-date folosite la instalarea și ștergerea fișierelor arhivate. Meta-datele includ scripturi ajutătoare, atrbute ale fișierelor și informații care descriu pachetul în sine. Pachetele pot exista în două varietăți: pachete binare, care încapsulează programele de instalat, și pachete sursă, care conțin codul sursă și rețeta necesară producării pachetelor binare.

Multe alte distribuții folosesc pachete RPM, în afară de mai cunoscutele RedHat Enterprise Linux, Mandriva (fostul Mandrake), Fedora Core și SuSE Linux. În afară de sfaturile specifice distribuției alese, puteți citi **man rpm**.

7.5.2.1.2. Exemple de pachete RPM

Cele mai multe pachete sunt instalate cu opțiunea de upgrade, **-U**, chiar dacă un pachet în sine este instalat sau nu. Pachetele RPM conțin versiunea completă a programului, care suprascrie versiunea existentă sau este instalat ca pachet nou. Utilizarea tipică este următoarea:

```
rpm -Uvh /path/to/rpm-package(s)
```

Opțiunea **-v** generează mai multe date de ieșire, iar **-h** afișează o bară de stare care arată progresul acțiunii:

```
[root@jupiter tmp]# rpm -Uvh totem-0.99.5-1.fr.i386.rpm
Preparing...                                           #####
1:totem                                         #####
[100%]
[100%]
[root@jupiter tmp]#
```

Pachetele care conțin un kernel nou vă sfătuim să le instalați cu opțiunea **-i**, pentru a nu suprascrie versiunea existentă a pachetului. Astfel, veți putea să porniți calculatorul cu vechiul kernel, dacă cel nou nu-și face treaba.

Puteți folosi **rpm** și pentru a verifica dacă un pachet este instalat în sistem:

```
[david@jupiter ~] rpm -qa | grep vim
vim-minimal-6.1-29
vim-X11-6.1-29
vim-enhanced-6.1-29
vim-common-6.1-29
```

Sau puteți afla care pachet conține un anumit fișier sau un executabil:

```
[david@jupiter ~] rpm -qf /etc/profile
setup-2.5.25-1

[david@jupiter ~] which cat
cat is /bin/cat

[david@jupiter ~] rpm -qf /bin/cat
coreutils-4.5.3-19
```

Nu trebuie să accesați cu privilegiile de administrator comanda **rpm** pentru a căuta în baza de date RPM. Aceste privilegii sunt necesare numai în cazul în care adăugați, modificați sau ștergeți pachete.

Vă prezentăm mai jos un ultim exemplu, care ilustrează modul în care puteți șterge un pachet gestionat de **rpm**:

```
[root@jupiter root]# rpm -e totem
[root@jupiter root]#
```

Ștergerea pachetelor nu afișează implicit date de ieșire, este normal să nu observați nimic. Când aveți dubii, folosiți **rpm -qa** din nou pentru a verifica dacă un anume pachet a fost șters.

RPM poate face multe alte lucruri, în afara funcțiilor de bază pe care le-am prezentat în această introducere. [Cum să-ți](#) despre RPM conține multe alte informații.

7.5.2.2. Pachetele DEB (.deb)

7.5.2.2.1. Ce sunt pachetele Debian?

Acest tip de pachete este cel implicit pe sistemele Debian GNU/Linux, în care **dselect** și, mai nou, **aptitude**, sunt uneltele standard pentru administrarea lor. Sunt folosite la selectarea pachetelor pe care le doriți instalate sau actualizate, dar vor rula și pe parcursul instalării unui sistem Debian pentru a vă ajuta să definiți metoda de acces folosită, pentru listarea pachetelor disponibile și pentru configurarea lor. (*Un exemplu de sistem bazat pe Debian este Ubuntu Linux, care, în modul grafic, dispune de doi administratori de pachete deb foarte bune, Synaptic în cazul Gnome și Adept în cazul KDE, care permit consultarea bazei de date, instalarea, actualizarea și ștergerea pachetelor într-un mod simplu, intuitiv^{n.t.}*).

Saitul [Debian](#) conține toate informațiile de care aveți nevoie, inclusiv o documentație adresată începătorilor pentru dselect ("dselect Documentation for Beginners").

În ultima vreme, pachetele Debian câștigă tot mai mult teren, devenind din ce în ce mai populare. La momentul scrierii acestei cărți, le foloseau cinci distribuții majore din topul celor zece. De asemenea, **apt-get** (consultați [Secțiunea 7.5.3.2](#)) este foarte folosit, chiar și pe sisteme non-DEB.

7.5.2.2.2. Exemple de unelte DEB

Verificați dacă un pachet este instalat sau nu cu **dpkg**. De exemplu, dacă doriți să știți ce versiune a programului Gallery este instalată pe sistemul dumneavostră:

```
nghtwsh@gorefest:~$ dpkg -l *gallery*
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Installed/Config-files/Unpacked/Failed-config/Half-installed
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=bad)
||/ Name          Version       Description
+++=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
ii  gallery        1.5-1sarge2    a web-based photo album written in php
```

Prefixul „ii” înseamnă că pachetul este instalat. Ar trebui să vedeați afișat prefixul „un” în cazul în care pachetul nu este instalat, dar există în lista gestionată de calculatorul dumneavostră.

Aflați cărui pachet aparține un fișier prin opțiunea **-S** a lui **dpkg**:

```
nghtwsh@gorefest:~$ dpkg -S /bin/cat
coreutils: /bin/cat
```

Mai multe informații găsiți în paginile Info ale **dpkg**.

7.5.2.3. Pachete sursă

Cea mai mare parte a programelor pentru Linux sunt Libere/Sursă deschisă, aşa că pachete care conțin sursa programelor sunt disponibile, ele fiind destul de răspândite. Fișierele sursă sunt necesare pentru compilarea versiunii dumneavoastră de program. Sursele unui program pot fi descărcate de pe site-ul său propriu, adesea ca arhivă compresată tarball (`program-versiune.tar.gz` sau similar). Pentru distribuțiile bazate pe RPM, sursa este adesea furnizată în `program-versiune.src.rpm`. Debian și cele mai multe distribuții bazate pe acesta asigură sursa adaptată care poate fi obținută folosind comanda **apt-get source**.

Cerințe specifice, dependințe și instrucțiuni de instalare se găsesc în fișierul `README`. Veți avea nevoie de un compilator C, **gcc**. Acest compilator C GNU este inclus în cele mai multe sisteme Linux și poate fi portat pe multe alte platforme.

7.5.3. Automatizarea administrării pachetelor și a actualizărilor

7.5.3.1. Generalități

Primul lucru pe care îl faceți după instalarea unui sistem nou este actualizarea acestuia; este un fapt care se aplică tuturor sistemelor de operare și Linux-ul nu este diferit.

Actualizările pentru cele mai multe sisteme Linux se găsesc de cele mai multe ori pe un sait apropiat, un sait-oglindă al distribuției dumneavoastră. O listă a saiturilor care oferă acest serviciu poate fi găsită pe saitul principal al distribuției dumneavoastră, consultați [Appendix A](#).

Actualizările trebuie aplicate cu regularitate, zilnic chiar dacă este posibil – dar și două săptămâni este un termen rezonabil. Trebuie să încercați să aveți cea mai recentă versiune a distribuției, deoarece Linux-ul se află într-o continuă schimbare. După cum am mai spus, facilități noi și rezolvări ale unor probleme sunt puse la dispoziție într-un ritm constant și, câteodată, sunt vizate aspecte importante legate de securitate.

Vestea bună este aceea că cele mai multe distribuții de Linux asigură unele care vă scutesc de actualizarea a zeci de pachete manual, zi de zi. Secțiunea următoare trece în revistă *administratorii administratorilor de pachete*. Sunt foarte multe de discutat pe marginea acestui subiect, chiar actualizări regulate a pachetelor sursă fiind administrate automat; vom lista doar cele mai întâlnite sisteme. Întotdeauna apelați la documentația specifică sistemului ales de dumneavoastră pentru înțelegerea corectă a procedurilor.

7.5.3.2. APT

The Advanced Package Tool este un sistem pentru pachetele de programe. Unealta pentru linia de comandă este **apt-get**, care dispune de pagini man excelente, care descriu cum să instalați și să actualizați pachetele și cum să instalați ultima versiune a unui pachet sau a întregii distribuții. APT își are rădăcinile în distribuția Debian GNU/Linux, în care este administratorul implicit pentru pachetele Debian. APT a fost portat și pe sistemele care folosesc pachete RPM. Principalul avantaj al APT este acela că este liber și flexibil în utilizare. Vă permite să configurați sisteme într-un mod asemănător cu unele unele specifice anumitor distribuții (în unele cazuri, comerciale), listate în secțiunea următoare.

În general, la prima utilizare a **apt-get**, trebuie să obțineți un index al pachetelor disponibile. Acest lucru se face prin comanda

apt-get update

După aceasta, puteți folosi **apt-get** pentru actualizarea sistemului dumneavoastră:

apt-get upgrade

Faceți acest lucru des, este o cale simplă prin care vă mențineți sistemul la zi și în siguranță.

În afară de acest mod general de utilizare, **apt-get** este foarte rapid în instalarea pachetelor individuale. Iată cum lucrează:

```
[david@jupiter ~] su - -c "apt-get install xsnow"
Password:
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
  xsnow

0 packages upgraded, 1 newly installed, 0 removed and 3 not upgraded.
Need to get 33.6kB of archives.
After unpacking 104kB of additional disk space will be used.
Get:1 http://ayo.freshrpms.net redhat/9/i386/os xsnow 1.42-10 [33.6kB]
Fetched 33.6kB in 0s (106kB/s)
Executing RPM (-Uvh) ...
Preparing...
  1:xsnow          ##### [100%]
                                         ##### [100%]
```

Observați opțiunea **-c** a comenții **su**, care indică consolii root să execute doar această comandă,

apoi să se revină la mediul utilizatorului. În acest mod nu puteți spune că ați uitat să părăsiți contul de root.

Dacă există dependințe față de alte pachete, **apt-get** va descărca și instala și aceste pachete necesare.

Mai multe informații puteți găsi în [APT HOWTO](#).

7.5.3.3. Sisteme care folosesc pachete RPM

Update Agent, care a fost disponibil inițial doar pentru pachetele RedHat RPM, este acum portat pentru o gamă largă de programe, inclusiv arhive (repozitoare) non-RedHat. Această unealtă asigură un sistem complet pentru actualizarea pachetelor RPM pe sistemele RedHat și Fedora Core. În linia de comandă, tastează **up2date** pentru a actualiza sistemul dumneavoastră. Pe spațiul de lucru este activată implicit o pictogramă (iconă) care vă spune dacă există sau nu actualizări disponibile pentru sistemul dumneavoastră.

Yellowdog`s Updater Modified (**yum**) este o altă unealtă care este din ce în ce mai apreciată. Este un program interactiv dar automat pentru instalarea, actualizarea și ștergerea pachetelor RPM. Este unealta implicită pe sistemele Fedora.

Pe sistemele SuSE Linux, totul este făcut cu ajutorul YaST, Yet another Setup Tool, care integrează mai multe sarcini de administrare a sistemului, printre care actualizarea pachetelor RPM. Începând cu SuSE Linux 7.1, puteți actualiza sistemul folosind o interfață web și YOUI, Yast Online Update.

Mandrake Linux și Mandriva asigură o unealtă numită URPMI, un set de programe strânse laolaltă, care au rolul de a simplifica instalarea de noi programe. Aceste unelte lucrează împreună cu RPMDrake și Mandrake Update pentru asigurarea celor necesare pentru instalarea și ștergerea în bune condiții a pachetelor de programe. MandrakeOnline oferă o gamă extinsă de servicii și poate notifica automat administratorii de sisteme atunci când sunt disponibile actualizări pentru distribuția Mandrake. Consultați **man urpmi**, printre altele, dacă dorîți mai multe informații.

De asemenea, administratorii spațiului de lucru KDE și Gnome dispun de propriile versiuni (grafice) ale administratorilor de pachete (*Adept și Synaptic, menționate mai sus^{n.t.}*).

7.5.4. Actualizarea kernelului

Cele mai multe instalări ale Linux-ului sunt în regulă dacă actualizați periodic distribuția pe care ați ales-o. Prin aceste actualizări de distribuție se va instala un kernel nou atunci când este nevoie și sunt făcute toate schimbările necesare în sistemul dumneavoastră. Comparați și instalați un kernel nou numai în cazul în care aveți nevoie de facilități care nu sunt cuprinse în kernelul implicit al distribuției Linux alese.

Indiferent dacă folosiți un kernel compilat și optimizat de dumneavoastră sau dacă îl folosiți pe cel pus la dispoziție de distribuție, instalați un nou kernel în coexistență cu cel vechi până în momentul în care sunteți siguri că funcționează aşa cum trebuie.

Creați un sistem dual boot care vă permite să alegeti ce kernel să fie încărcat prin configurarea încărcătorului dumneavoastră de sisteme, cu ajutorul fișierului **grub.conf**. Iată un exemplu simplu:

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making config changes.
# NOTICE: You have a /boot partition. This means that
#          all kernel and initrd paths are relative to /boot/, e.g.
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/hde8
#          initrd /initrd-version.img
#boot=/dev/hde
default=0
timeout=10
splashimage=(hd0,0)/grub/splash.xpm.gz
title Red Hat Linux new (2.4.9-31)
    root (hd0,0)
    kernel /vmlinuz-2.4.9-31 ro root=/dev/hde8
    initrd /initrd-2.4.9-31.img
title old-kernel
    root (hd0,0)
    kernel /vmlinuz-2.4.9-21 ro root=/dev/hde8
    initrd /initrd-2.4.9-21.img
```

După ce noul kernel se dovedește a fi stabil și fiabil, puteți șterge liniile care definesc kernelul vechi din fișierul de configurare al GRUB-ului, deși vă sfătuim să așteptați măcar câteva zile, pentru a fi siguri.

7.5.5. Instalarea de pachete aflate pe CD-ul de instalare

7.5.5.1. Montarea unui CD

Se face, în principiu, în același mod în care instalați manual pachetele, cu excepția faptului că trebuie să adăugați sistemul de fișiere al CD-ului la calculatorul dumneavoastră pentru a-l face accesibil. Pe cele mai multe sisteme, este un proces automat, lansat la inserarea CD-ului în unitatea optică, deoarece serviciul **automount** este pornit la inițierea sistemului. Dacă CD-ul dumneavoastră nu este disponibil automat, introduceți comanda **mount** într-o fereastră terminal. În funcție de configurația sistemului, o linie similară cu cea care urmează face acest lucru:

```
mount /dev/cdrom /mnt/cdrom
```

Pe unele sisteme, doar folosind contul de *root* puteți monta medii de stocare externe; este un lucru care depinde de configurația sistemului.

Pentru automatizarea acestor procese, unitatea optică are o intrare în directorul */etc/fstab*, care listează sistemele de fișiere și punctele lor de montare, care formează sistemul arborescent de fișiere. Iată o astfel de linie:

```
[david@jupiter ~] grep cdrom /etc/fstab
/dev/cdrom      /mnt/cdrom      iso9660      noauto,owner,ro 0 0
```

Ea indică faptul că sistemul înțelege comanda **mount /mnt/cdrom**. Opțiunea *noauto* înseamnă că pe acest sistem, CD-urile nu sunt montate automat.

Puteți încerca un clic dreapta pe pictograma CD-ului aflată pe spațiul de lucru pentru a monta CD-ul, dacă administratorul de fișiere nu face acest lucru pentru dumneavoastră. Comanda **mount** fără argumente verifică dacă acțiunea a avut succes:

```
[david@jupiter ~] mount | grep cdrom
/dev/cdrom on /mnt/cdrom type iso9660 (ro,nosuid,nodev)
```

7.5.5.2. Folosirea CD-ului

După montarea unui CD, puteți schimba directorul, de regulă în punctul de montare */mnt/cdrom*, unde puteți accesa conținutul CD-ului. Folosiți aceleași comenzi pe care le utilizați în lucrul cu fișierele aflate pe discul fix.

7.5.5.3. Ejectarea CD-ului

Pentru a scoate CD-ul din unitate, după ce ați terminat, sistemul de fișiere al CD-ului trebuie să figureze ca nefolosit. Chiar dacă vă aflați doar într-unul din subdirectoarele punctului de montare, în cazul nostru, */mnt/cdrom*, CD-ul figurează ca „folosit” față de sistem, aşa că trebuie mai întâi să ieșiți de acolo. Faceți acest lucru prin introducerea comenзii **cd** fără argumente, ceea ce vă duce în directorul dumneavoastră personal. După aceasta, puteți folosi comanda

```
umount /mnt/cdrom
```

sau

```
eject cdrom
```



Dispozitive blocate

NICIODATĂ să nu forțați unitățile optice. Trucul cu agrafa pentru hârtie este o idee proastă, deoarece vă scoate, într-adevăr CD-ul, dar sistemul va crede că acesta este încă acolo, deoarece nu au fost urmate procedurile corecte. De cele mai multe ori, doar repornirea sistemului vă ajută să aduceți sistemul

într-o stare normală.

Dacă primiți mesajul „device busy” (dispozitiv ocupat), verificați în primul rând dacă toate sesiunile consolei au părăsit fișierele CD-ului sau dacă acesta este folosit de aplicațiile grafice. Dacă aveți îndoieți, folosiți unealta **lsof** pentru a găsi procesele care folosesc resursele afilate pe CD.

7.6. Sumar

Când toate sunt în locul potrivit, jumătate de treabă este făcută.

Păstrarea ordinii este importantă și la fel de important este să vă organizați bine mediul directorului personal, în modul text sau grafic. Mediul text este controlat de fișierele de configurare a consolei. Mediul grafic depinde în primul rând de configurarea serverului X, în care sunt integrate mai multe aplicații, cum ar fi administratorii de ferestre și ai spațiului de lucru, fiecare având propriile fișiere de configurare. Trebuie să citiți documentația specifică sistemului și programelor dumneavoastră pentru a afla modul corect de configurare a acestora.

Configurările regionale, cum ar fi cele ale tastaturii, instalarea fonturilor potrivite și limba folosită de sistem sunt, cel mai bine, făcute la instalarea sistemului.

Programele sunt administrate în mod automat sau manual prin pachetele de programe.

Următoarele comenzi au fost tratate în acest capitol:

Tabelul 7-2. Comenzi noi în Capitolul 7: Faceți-vă comod acasă

Comandă	Înțelesul comenzi
aptitude	Administrează pachete Debian.
automount	Montează automat sisteme de fișiere nou inserate.
dpkg	Administratorul de pachete Debian.
dselect	Administrează pachete Debian.
loadkeys	Încarcă configurația tastaturii.
lsof	Identifică procesele.
mount	Montează sisteme noi de fișiere.
ntpdate	Configurează timpul sistemului folosind un server de timp.
quota	Afișează informații despre folosirea spațiului pe discul fix.
recode	Convertește fișierele la un alt set de caractere.
rpm	Administrează pachetele RPM.
setfont	Alege un font.
timezone	Configurează zona de timp.
tzconfig	Configurează zona de timp.
ulimit	Configurează sau afișează limitarea resurselor.
up2date	Administrează pachete RPM.
urpmi	Administrează pachete RPM.
yum	Administrează pachete RPM.

7.7. Exerciții

7.7.1. Mediul consolei

- Afişați mediul consolei. Ce variabilă poate fi folosită pentru stocarea tipului de procesor al

calculatorului dumneavoastră?

- Realizați un script care poate afișa ceva de genul unei linii „hello, world”. Acordați-i permisiunile necesare pentru a-l rula. Testați acest script.
 - Realizați un director în directorul dumneavoastră personal și mutați acest script în el. Adăugați directorul creat la traseul de căutare al sistemului, în mod permanent. Testați dacă scriptul poate fi executat fără a-i da traseul complet al locului în care se află.
 - Realizați subdirectoră în directorul dumneavoastră personal pentru păstrarea de fișiere diverse, de exemplu un director **muzică** pentru păstrarea fișierelor audio, un director **documente** pentru notele dumneavoastră, și aşa mai departe. Folosiți-le!
 - Configurați un prompter personalizat.
 - Afişați limitele aplicate resurselor. Le puteți modifica?
 - Încercați să citiți paginile de manuale fără a le decompresa.
 - Realizați un alias **III** care să execute de fapt **ls -la**.
 - De ce nu funcționează comanda **tail testfile > testfile**?
 - Montați un CD cu date, cum ar fi discul de instalare al distribuției dumneavoastră de Linux și explorați-l. Nu uitați să-l ejectați când nu mai aveți nevoie de acesta.
 - Scriptul care se găsește la [Sectiunea 7.2.5.2](#) nu este perfect. El generează erori pentru fișierele care sunt directoare. Adaptați scriptul astfel încât să selecteze doar fișierele simple pentru copiere. Folosiți **find** pentru selecție. Nu uitați să faceți scriptul executabil înainte de a-l rula.
-

7.7.2 Mediul grafic

- Încercați toate butoanele mausului în regiuni diferite (terminal, fundal, barele de meniu).
 - Explorați meniul.
 - Personalizați-vă fereastra terminal.
 - Folosiți butoanele mausului pentru copierea de text dintr-un terminal într-altul.
 - Configurați administratorul de ferestre; încercați diferite spații de lucru (ecrane virtuale).
 - Adăugați applet-uri, cum ar fi un monitor de sistem, barei de meniu.
 - Schimbați tema.
 - Activăți opțiunea prin care ferestrele sunt activate prin mișcarea cursorului mausului pe deasupra acesteia, astfel încât nu mai este necesar să faceți clic pe fereastră pentru ca aceasta să fie activă (*sloppy focus*).
 - Schimbați administratorul de ferestre.
 - Deautentificați-vă și selectați un alt tip de sesiune, de exemplu KDE dacă ați folosit Gnome. Repetați pasul.
-

Capitolul 8. Imprimante și tipărire

În acest capitol vom prezenta lucruri legate de imprimante și tipărirea fișierelor. După citirea acestei părți, veți putea să:

- ◆ Formatați documente.
 - ◆ Vizualizați documentele înainte de a le trimite la tipărire.
 - ◆ Alegeți o imprimantă bună, care să fie recunoscută de sistemul dumneavoastră Linux.
 - ◆ Tipăriți fișiere și verifica statutul imprimantei.
 - ◆ Rezolvați problemele legate de imprimante.
 - ◆ Găsiți documentația necesară instalării unei imprimante.
-

8.1. Tipărirea fișierelor

8.1.1. Tipărirea în linie de comandă

8.1.1.1. Trimiterea fișierului la imprimantă

Tipărirea din interiorul unei aplicații este simplă, cu ajutorul opțiunii Print (Tipărește) din meniu.

Din linia de comandă, folosiți comenziile **lp** sau **lpr**.

lp fișier (e)

lpr fișier (e)

Aceste comenzi pot citi o conexiune (pipe), astfel încât puteți tipări ieșirea unei comenzi folosind **comandă | lp**

Aveți la dispoziție multe opțiuni pentru configurarea aspectului paginii, numărului de copii, imprimanta pe care dorîți să o folosiți, dacă există mai multe, alegerea tipului hârtiei, tipărirea pe o față sau pe ambele fețe, dacă imprimanta asigură acest lucru, configurarea marginilor, etc. Citiți paginile manualelor pentru mai multe informații.

8.1.1.2. Statutul lucrărilor trimise la tipărire

Odată ce fișierul este acceptat în coada de așteptare pentru tipărire, îi este alocat un număr de identificare:

```
davy:~> lp /etc/profile
request id is blob-253 (1 file(s))
```

Pentru a verifica stadiul lucrărilor, folosiți comenziile **lpq** sau **lpstat**. La introducerea acestora fără argumente, vor afișa conținutul implicit:

```
davy:~> lpq
blob is ready and printing
Rank  Owner   Job    File(s)        Total Size
active  davy    253    profile      1024 bytes
davy:~> lpstat
blob-253      davy    1024    Tue 25 Jul 2006 10:20_01 AM CEST
```

8.1.1.3. Statutul imprimantei

Care este imprimanta implicită folosită de un sistem care are acces la mai multe imprimante?

lpstat -d

```
davy:~> lpstat -d
system default destination: blob
```

Care este statutul imprimantei mele?

lpstat -p

```
davy:~> lpstat -p
printer blob now printing blob-253. enabled since Jan 01 18:01
```

8.1.1.4. Ștergerea din coada de așteptare a lucrărilor trimise la imprimat

Dacă nu vă place ceea ce vedeți prin comenzi care afișează stadiul imprimării lucrărilor, folosiți **lprm** sau **cancel** pentru a șterge sarcinile legate de tipărire documentelor.

```
davy:~> lprm 253
```

În mediul grafic, poate să apară o fereastră care vă spune că o anumită sarcină legată de tipărire a fost anulată.

În medii mai mari, comanda **lpq** este folosită pentru controlarea mai multor imprimante. Consultați paginile Info ale fiecărei comenzi.

Aveți la dispoziție mai multe interfețe grafice pe care le puteți folosi pentru comanda **lp**, iar cele mai multe aplicații grafice au o funcție de tipărire care folosește **lp**. Căutați în paginile de ajutor integrate și în documentația specifică fiecărui program pentru mai multe informații.



De ce există două comenzi pentru fiecare sarcină legată de tipărire?

Tipărirea pe sistemele UNIX și pe cele asemănătoare are o istorie lungă. Au existat două abordări: stilul de tipărire BSD și stilul SystemV. Din motive de compatibilitate, serviciul CUPS al Linux-ului lucrează cu ambele. De asemenea, **lp** nu se comportă exact ca **lpr**, **lpq** are încrucișări de opțiuni diferite de cele ale **lpstat** și **lprm** este aproape același lucru, dar nu exact același ca **cancel**. Nu este important pe care-l folosiți, alegeți comenzi care vă sunt mai la îndemână sau pe cele cu care sunteți obișnuit de pe alte sisteme asemănătoare UNIX-ului.

8.1.2. Formatarea

8.1.2.1. Unelte și limbaje

Dacă dorîți să obțineți ceva anume de la imprimantă, fișierele trebuie mai întâi formatare. În afara programelor pentru formatare, care există din abundență, Linux-ul conține uneltele de bază ale UNIX-ului pentru formatare și limbaj.

Sistemele moderne Linux asigură tipărirea directă, fără o formatare prealabilă executată de către utilizator, pentru o gamă largă de fișiere: text, PDF, PostScript și pentru câteva formate de imagini precum PNG, JPEG, BMP și GIF.

Pentru acele fișiere care au nevoie de formatare, Linux-ul integrează unelte de formatare precum comenziile **pdf2ps**, **fax2ps** și **a2ps**, care convertește alte formate în PostScript. Aceste comenzi pot crea fișiere care apoi pot fi folosite pe alte sisteme care nu dispun de uneltele de conversie.

În afară de aceste comenzi ale liniei de comandă aveți la dispoziție multe unelte grafice pentru procesarea textului. Există numeroase suite pentru birou, multe dintre acestea fiind libere și gratuite. Acestea realizează automat formatarea în momentul inițierii unei sarcini pentru tipărire. Pentru a numi doar câteva: OpenOffice.org, KOffice, AbiWord, WordPerfect, etc.

Iată câteva limbaje comune în contextul sarcinilor pentru tipărire:

- **groff**: versiunea GNU a comenzi UNIX **roff**. Este o interfață a sistemului de formatare groff. În mod

normal, rulează comanda **troff** și un post-procesor convenabil pentru dispozitivul selectat. Permite generarea fișierelor PostScript.

- TeX și pachetul macro LaTeX: unul dintre cele mai răspândite limbaje de marcare pe sistemele UNIX. Invocat de regulă ca **tex**, formatează fișiere și le trimit către un dispozitiv corespondent o reprezentare independentă de tipul documentului.

Lucrurile care țin de domeniul tehnic sunt făcute cu LaTeX deoarece acesta integrează formule matematice, fiind depuse eforturi de către [W3C](#) (Consorțiul World Wide Web) pentru includerea acestei facilități și în alte aplicații.

- SGML și XML: sunt disponibile formate libere pentru UNIX și Linux. XML este noua generație SGML, care formează baza pentru DocBook XML, un sistem pentru documente (această carte este scrisă în XML, de exemplu – cartea originală^{n.t.}).



Tipărirea documentației

Paginile de manuale conțin date pre-formatate **troff** care trebuie să fie formatare înainte de a le trimite la imprimantă. Tipărirea se face folosind opțiunea **-t** pentru comanda **man**:

```
man -t command > man-command.ps
```

Apoi tipăriți fișierul PostScript. Dacă este configurață o destinație implicită pentru tipărire în cazul sistemului/contului dumneavoastră, puteți introduce comanda **man -t comandă** pentru a trimite pagina formatată direct la imprimantă.

8.1.2.2. Vizualizarea fișierelor formatare

Orice trimiteți la imprimantă poate fi afișat și pe ecran. În funcție de formatul fișierului, puteți folosi una din comenziile de mai jos:

- Fișiere PostScript: prin comanda **gv** (GhostView).
- Fișiere TeX dvi: prin **xdvi** sau **kdv**, propriu KDE-ului.
- Fișiere PDF: **xpdf**, **kpdf**, **gpdf** sau vizualizatorul Adobe, **acroread**, care este disponibil gratuit dar nu este un program liber. Adobe's reader este destinat versiunilor PDF 1.6, celelalte funcționează numai pentru versiunile până la 1.5. Versiunea unui PDF poate fi afișată prin comanda **file**.
- Din interiorul unor aplicații, cum ar fi Firefox sau OpenOffice, puteți alege opțiunea Vizualizare înainte de tipărire (Print Preview) din meniurile acestora.

8.2. Serviciul în sine

8.2.1. Generalități

Până acum câțiva ani, alegerea utilizatorilor de Linux era una simplă: toată lumea rula același vechi program pentru tipărire, Net-2 code al BSD. Apoi LPRng a devenit mai folosit, dar în zilele noastre, cele mai moderne distribuții de Linux folosesc CUPS, Common UNIX Printing System (Sistemul de tipărire comun UNIX). CUPS este o implementare a Internet Printing Protocol (IPP), un protocol standard RFC, asemănător HTTP-ului, înlocuitor al venerabilului (și demodatului) protocol LPD. CUPS este distribuit sub licență publică GNU. CUPS este și sistemul pentru tipărire implicit pe calculatoarele MacOSX.

8.2.2. Configurarea imprimantei în modul grafic

Cele mai multe distribuții disponă de o interfață grafică pentru configurarea imprimantelor locale (pe porturile paralele sau USB) sau de rețea. Vă permite să alegeți tipul imprimantei dintr-o listă pusă la dispoziție și chiar puteți testa acea imprimantă. Nu aveți de ce să vă bateți capul cu sintaxa comenzi sau cu localizarea fișierelor de configurare. Întotdeauna, înainte de instalarea unei imprimante, verificați documentația sistemului dumneavoastră.

CUPS poate fi configurat cu ajutorul unei interfețe web care rulează pe portul 631 al calculatorului dumneavoastră. Pentru a verifica dacă această opțiune este disponibilă, încercați adresa localhost:631/help sau localhost:631/.

8.2.3. Cumpărarea unei imprimante pentru Linux

Tot mai mulți producători de imprimante pun la dispozitie drivere pentru CUPS astfel încât acest serviciu se va conecta cu ușurință la o imprimantă care folosește un port serial, paralel sau USB ori la o imprimantă de pe rețea. CUPS va asigura o prezentare uniformă pentru toate tipurile de imprimante.

Imprimantele care vin doar cu drivere pentru Win9x pot pune probleme dacă nu au un driver pentru Linux scris pentru acestea. Verificați întotdeauna la adresa <http://linuxprinting.org/> atunci când aveți îndoieți.

În trecut, cea mai bună alegere ar fi fost o imprimantă care are integrată în firmware facilitatea PostScript, deoarece aproape toate programele UNIX sau Linux care generează ieșiri pentru tipărire o fac în PostScript, limbajul de control ales de industria imprimantelor. Imprimantele PostScript sunt puțin mai scumpe, dar sunt dispozitive independente, care folosesc un limbaj de programare liber și puteți fi siguri 100% că vor funcționa. În zilele noastre, importanța acestei reguli de bun simț este trecută cu vederea.

8.3. Probleme legate de tipărire

În această secțiune vom discuta despre ce puteți face ca utilizator atunci când ceva merge prost. Nu ne vom referi la problemele care au de-a face cu partea de serviciu în sine a sistemului de tipărire, întrucât acest lucru este o sarcină pentru administratorii de sistem.

8.3.1. Fișier greșit

Dacă este tipărit un fișier greșit, puteți anula cu ajutorul comenzi **Iprm IDsarcină**, unde IDsarcină este afișat sub forma *numele_imprimantei-numărul_sarcinii_de_tipărit* (poate fi găsit printre informațiile afișate de **Ipq** și **Ipstat**). Această abordare este posibilă numai în cazul în care fișierul dumneavoastră așteaptă în coada pentru tipărire. Trebuie să fiți îndeajuns de rapizi dacă sunteți singurul care utilizează imprimanta, întrucât documentele sunt trimise la imprimantă în câteva secunde. Odată ce au ajuns la imprimantă, este prea târziu să ștergeți sarcina folosind uneltele Linux-ului.

Tot ceea ce puteți încerca în aceste cazuri sau în cele în care ați configurat un driver greșit și din imprimantă ies doar gunoaie, este să opriți imprimanta. Nu este chiar o metodă foarte bună, deoarece poate cauza blocarea hârtiei și alte nereguli.

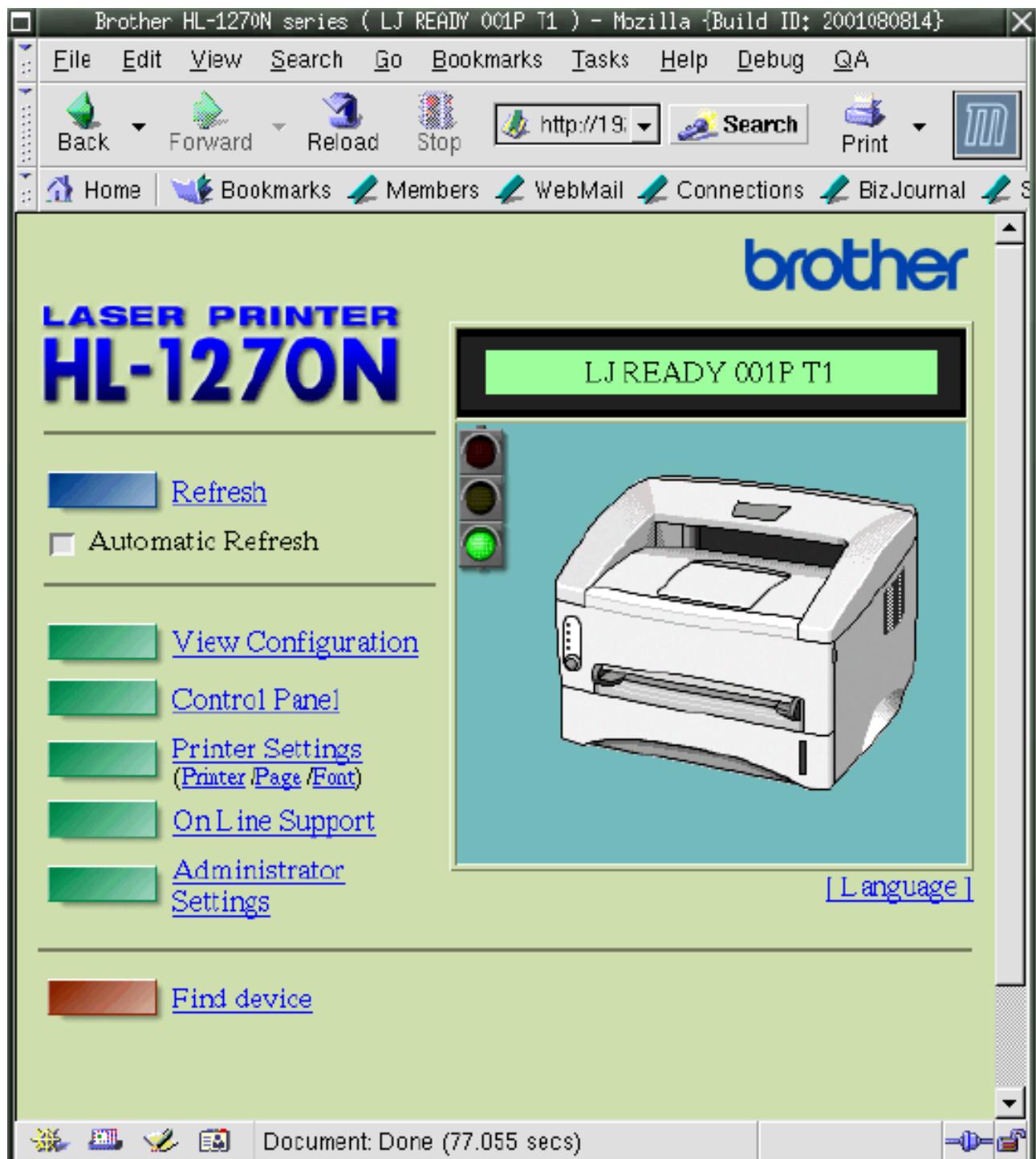
8.3.2. Tipăritura nuiese din imprimantă

Folosiți comanda **Ipq** pentru a găsi sarcina care nu este finalizată:

```
elly:~> ipq
Printer: lp@blob
Queue: 2 printable jobs
Server: pid 29998 active
Unspooler: pid 29999 active
Status: waiting for subserver to exit at 09:43:20.699
Rank   Owner/ID          Class Job Files      Size Time
1      elly@blob+997     A    997 (STDIN)    129 09:42:54
2      elly@blob+22       A     22 /etc/profile  917 09:43:20
```

Foarte multe imprimante dispun de interfețe web, care pot afișa informații despre statutul imprimantei, prin introducerea adresei imprimantei în navigatorul dumneavoastră de Internet. Pe pagina care urmează vă prezentăm o astfel de interfață.

Figura 8-1. Statutul imprimantei afișat de o interfață web.



Interfață web CUPS versus interfață web a imprimantei

Rețineți că ceea ce aveți prezentat în imaginea de mai sus nu este interfața web CUPS, ci o interfață web specifică unei imprimante care integrează această facilitate. Verificați documentația imprimantei pe care o folosiți pentru a afla dacă poate lucra în acest mod.

Dacă ID-ul sarcinii dumneavoastră nu este de găsit nici la imprimantă, contactați administratorul de sistem. Dacă ID-ul este listat, verificați dacă imprimanta tipărește. Dacă da, așteptați, va sosi și rândul documentelor dumneavoastră. Dacă imprimanta nu tipărește nimic, verificați dacă are hârtie, verificați conexiunile fizice cu rețeaua de electricitate și cablul de date. Este posibil să aibă nevoie de o repornire. Întrebați și administratorul de sistem dacă aveți frecvent probleme de acest gen.

În cazul unei imprimante de rețea, încercați să tipăriți de pe alt calculator. Dacă imprimanta este găsită de calculatorul dumneavoastră (consultați [Capitolul 10](#) în care se vorbește despre utilitatea **ping**),

Încercați să trimiteți un fișier formatat, cum ar fi un `fișier.ps` în cazul unei imprimante PostScript, folosind un client FTP. Dacă imprimanta lucrează, înseamnă că sistemul dumneavoastră de tipărire nu este configurat aşa cum trebuie. Dacă nu, imprimanta nu înțelege formatul în care ați realizat documentul trimis la tipărire.

Saitul [GNU/Linux Printing](#) conține mai multe sfaturi pe marginea acestui subiect.

8.4. Sumar

Serviciul de tipărire din Linux integrează o serie de unelte pentru tipărire bazate pe uneltele standard UNIX LPD, fie că vorbim despre implementarea System V sau cea BSD. Vă prezentăm mai jos mai multe comenzi legate de tipărire:

Tabelul 8-1. Comenzi noi în Capitolul 8: Tipărirea

Comanda	Înțelesul comenzi
lpr sau lp	Tipărește un fișier
lpq sau lpstat	Afișează documentele trimise la tipărire, aflate în așteptare
lprm sau cancel	Șterge documentele trimise în așteptare pentru tipărire
acroread	Vizualizator PDF
groff	Unealtă pentru formatare
gv	Vizualizator PostScript
printconf	Configurează imprimantele
xpdf	Vizualizator DVI
xps	Vizualizator PDF
*2ps	Convertește fișierele în PostScript

8.5. Exerciții

Configurarea și testarea imprimantei implică existența uneia și accesul la contul de *root*. Dacă sunt îndeplinite aceste condiții, puteți încerca următoarele:

- Instalați imprimanta folosind interfața grafică (GUI) a sistemului dumneavoastră.
- Tipăriți o pagină de test folosind aceeași interfață grafică.
- Tipăriți o pagină de test folosind comanda **lp**.
- Tipăriți din interiorul unei aplicații, de exemplu Mozilla sau OpenOffice, selectând Fișier -> Tipărește (File -> Print) din meniu.
- Deconectați imprimanta de la rețea sau de la calculator/server de imprimare. Ce se întâmplă când încercați să tipăriți ceva?

Următoarele exerciții pot fi făcute fără imprimantă sau accesarea contului de administrator.

- Încercați să faceți fișiere PostScript din diferite fișiere sursă (cum ar fi HTML, PDF, pagini man). Testați rezultatul cu vizualizatorul **gv**.
- Verificați dacă serviciul de tipărire rulează.
- Tipăriți oricum fișierul. Ce se întâmplă?
- Faceți un fișier PostScript folosind Mozilla. Testați-l cu **gv**.
- Convertiți acel fișier în format PDF. Testați-l cu **xpdf**.
- Cum ați putea tipări un fișier GIF din linia de comandă?
- Folosiți **a2ps** pentru a tipări fișierul /etc/profile într-un fișier de ieșire. Testați-l din nou cu **gv**. Ce se întâmplă dacă nu specificați un fișier de ieșire?

Capitolul 9. Tehnici fundamentale pentru crearea copiilor de siguranță

Mai devreme sau mai târziu, se pot întâmpla accidente. În acest capitol vom discuta despre cum să trimiteți datele într-un loc sigur folosind un alt calculator, dischete, CD-uri și benzi. Vom discuta și despre cele mai folosite programe de comprimare și comenzi de arhivare.

La terminarea acestui capitol, veți ști cum să:

- ◆ Creați, explorați și despachetați fișiere arhivate.
- ◆ Lucrați cu dischetele și cum să creați o dischetă de pornire pentru sistem.
- ◆ Scrieți CD-uri.
- ◆ Realizați copii de siguranță incrementale.
- ◆ Creați arhive Java.
- ◆ Căutați documentația necesară pentru utilizarea altor dispozitive și programe pentru copiile de siguranță.
- ◆ Criptați (cifrați) datele dumneavoastră.

9.1. Introducere

Cu toate că Linux-ul este unul din cele mai sigure sisteme de operare existente, fiind proiectat să lucreze non-stop, mai pot fi pierdute date. Pierderile de date sunt de cele mai multe ori o consecință a erorilor în utilizare, rareori fiind generate de o disfuncționalitate a sistemului, cum ar fi o cădere de tensiune sau defectarea discului fix, aşa că păstrarea unor copii ale datelor importante este o idee foarte bună.

9.1.1. Pregătirea datelor

9.1.1.1. Arhivarea cu tar

De cele mai multe ori, datele care trebuie arhivate sunt colectate într-un singur fișier, pe care îl veți comprima mai târziu. Procesul de arhivare implică concatenarea tuturor fișierelor listate și scoaterea spațiilor goale. În Linux acest lucru este făcut prin comanda **tar**. Comanda **tar** a fost creată la început pentru arhivarea datelor pe benzi, dar poate crea și arhive, cunoscute ca *tarballs*.

tar are multe opțiuni, cele mai importante fiind prezentate mai jos:

- **-v**: generează informații.
- **-t**: testează, arată conținutul unei arhive tarball.
- **-x**: extrage arhiva.
- **-c**: creează arhivă.
- **-f dispozitiv_de_arhivare**: folosește `dispozitiv_de_arhivare` ca sursă/destinație a arhivei, implicit fiind primul dispozitiv cu bandă (de regulă `/dev/st0` sau similar).
- **-j**: filtrează prin **bzip2**, consultați [Sectiunea 9.1.1.2](#).

În mod normal, prefixul dash (-) este păstrat în opțiunile **tar**, după cum veți observa din exemplele care urmează.



Pentru compatibilitate, folosiți comanda tar GNU

Arhivele create cu versiunea proprietară a **tar** pe un anume sistem pot fi incompatibile cu altă versiune proprietară **tar** aflată pe alt sistem. Acest lucru vă poate da multe bătăi de cap, legate de necesitatea decomprimării arhivei pe un sistem care nu mai există. Folosiți pe toate sistemele versiunea GNU a programului **tar**, pentru a scuti administratorul de sistem de probleme în plus. Linux-ul folosește în totdeauna programul tar GNU. Atunci când lucrăți pe sisteme UNIX, introduceți comanda **tar --help** pentru a verifica cu ce versiune lucrăți. Luăți legătura cu administratorul de sistem dacă nu observați niciunde acronimul GNU.

În exemplul de mai jos este creată și despachetată o arhivă:

```
gaby:~> ls images/
mettux.jpg nimf.jpg

gaby:~> tar cvf images-in-a-dir.tar images/
images/
images/nimf.jpg
images/mettux.jpg

gaby:~> cd images

gaby:~/images> tar cvf images-without-a-dir.tar *.jpg
mettux.jpg
nimf.jpg

gaby:~/images> cd

gaby:~> ls */*.tar
images/images-without-a-dir.tar

gaby:~> ls *.tar
images-in-a-dir.tar

gaby:~> tar xvf images-in-a-dir.tar
images/
images/nimf.jpg
images/mettux.jpg

gaby:~> tar tvf images/images-without-dir.tar
-rw-r--r-- gaby/gaby 42888 1999-06-30 20:52:25 mettux.jpg
-rw-r--r-- gaby/gaby 7578 2000-01-26 12:58:46 nimf.jpg

gaby:~> tar xvf images/images-without-a-dir.tar
mettux.jpg
nimf.jpg

gaby:~> ls *.jpg
mettux.jpg nimf.jpg
```

Exemplul ilustrează, de asemenea, diferența dintre un director arhivat și o adunătură de fișiere arhivate. Vă sfătuim să arhivați directoare, pentru ca fișierele să nu se împrăștie peste tot la despachetarea arhivei tarball (vă folosește foarte mult în momentul despachetării pe alte sisteme, în cazul cărora nu cunoașteți ce fișiere au existat pe acele sisteme și ce fișiere ați despachetat din arhivă).

Dacă există o unitate cu bandă conectată la calculator și configurată de administratorul de sistem, fișierele care se termină în .tar sunt înlocuite cu numele dispozitivului cu bandă, de exemplu:

```
tar cvf /dev/tape mail/
```

Directorul mail și toate fișierele care sunt conținute de acesta sunt comprimate într-un fișier care este scris imediat pe o bandă. Este listat și conținutul acestuia pentru că a fost folosită opțiunea verbose.

9.1.1.2. Copii de siguranță incrementale cu ajutorul tar

Programul **tar** poate să creeze copii de siguranță incrementale prin introducerea opțiunii -N. Cu această opțiune puteți specifica o dată, iar **tar** verifică modificările fișierelor asociate. Dacă aceste fișiere au fost modificate după data introdusă, ele vor fi incluse în copia de siguranță. Exemplul de mai jos folosește o notă de timp aplicată unei arhive ca valoare pentru dată. În primul rând, este creată arhiva inițială și este arătată nota de timp. Apoi, la apariția unui fișier nou, va fi făcută o copie de siguranță nouă, care conține doar acest fișier, nou apărut.

Imaginea care urmează ilustrează situația descrisă mai sus:

```
jimmy:~> tar cvpf /var/tmp/javaproggies.tar java/*.java
java/btw.java
java/error.java
java/hello.java
java/income2.java
java/income.java
java/inputdevice.java
java/input.java
java/master.java
java/method1.java
java/mood.java
java/moodywaitress.java
java/test3.java
java/TestOne.java
java/TestTwo.java
java/Vehicle.java

jimmy:~> ls -l /var/tmp/javaproggies.tar
-rw-rw-r-- 1 jimmy   jimmy  10240 Jan 21 11:58 /var/tmp/javaproggies.tar

jimmy:~> touch java/newprog.java

jimmy:~> tar -N /var/tmp/javaproggies.tar \
-cvp /var/tmp/incremental1-javaproggies.tar java/*.java 2> /dev/null
java/newprog.java

jimmy:~> cd /var/tmp/

jimmy:~> tar xvf incremental1-javaproggies.tar
java/newprog.java
```

Erorile standard sunt redirecționate către `/dev/null`. Dacă nu faceți acest lucru, **tar** va afișa un câte mesaj pentru fiecare fișier care nu a suferit modificări, spunându-vă că acestea nu vor fi incluse.

Acest mod de lucru are dezavantajul că se caută notele de timp ale fișierelor. Să spunem că descărcați o arhivă într-un director care conține copiile dumneavoastră de siguranță, iar arhiva conține fișiere care au fost create cu doi ani în urmă. La compararea notei de timp asociate acestor fișiere cu nota de timp a arhivei initiale, noile fișiere vor apărea pentru **tar** ca fiind vechi, și nu vor fi incluse în copia de siguranță incrementală realizată cu opțiunea `-N`.

O alegere mai bună ar fi opțiunea `-g`, prin care se generează o listă a fișierelor ce vor fi incluse în copia de siguranță. La realizarea copiilor de siguranță incrementale, sunt verificate fișierele din această listă. Iată cum lucrează opțiunea în cauză:

```
jimmy:~> tar cvpf work-20030121.tar -g snapshot-20030121 work/
work/
work/file1
work/file2
work/file3

jimmy:~> file snapshot-20030121

snapshot-20030121: ASCII text
```

În ziua următoare, utilizatorul *jimmy* modifică fișierul `file3` și creează fișierul `file4`. La sfârșitul zilei, el face o nouă copie de siguranță:

```
jimmy:~> tar cvpf work-20030122.tar -g snapshot-20030121 work/
work/
work/file3
work/file4
```

V-am prezentat câteva comenzi simple, dar le puteți folosi într-o sarcină programată (cronjob - consultați [Sectiunea 4.4.4](#)), în care să specificați, de exemplu, un fișier întă pentru copia de siguranță săptămânală și un altul pentru copia de siguranță zilnică. Fișierele întă vor fi înlocuite la realizarea unei copii de siguranță integrale.

Mai multe informații puteți găsi în documentația **tar**.



De reținut

După cum probabil ati observat, **tar** este de un real folos atunci vorbim despre directoare normale, adică un set de fișiere de același fel. Există însă și unelte care sunt mai ușor de administrat atunci când dorîti să arhivați partiții întregi, discuri sau proiecte mai mari. Am vorbit despre **tar** deoarece este foarte utilizat în distribuirea arhivelor. Se întâmplă adesea să vă loviți de instalarea unui program care vine sub forma așa numitor „arhive tarball”. Consultați [Secțiunea 9.3](#) pentru căi mai ușoare de realizare a copiilor de siguranță regulate.

9.1.1.3. Arhivarea și despachetarea cu gzip sau bzip2

Datele, inclusiv arhivele tarball, pot fi compresate folosind uneltele zip. Comanda **gzip** va adăuga sufixul .gz la numele fișierului și va șterge fișierul original.

```
jimmy:~> ls -la | grep tar
-rw-rw-r-- 1 jimmy jimmy 61440 Jun  6 14:08 images-without-dir.tar

jimmy:~> gzip images-without-dir.tar

jimmy:~> ls -la images-without-dir.tar.gz
-rw-rw-r-- 1 jimmy jimmy 50562 Jun  6 14:08 images-without-dir.tar.gz
```

Despachetați fișierele folosind opțiunea **-d** pentru comanda **gzip**.

bzip2 lucrează într-un mod asemănător, dar folosește un algoritm complex de comprimare, rezultând fișiere mai mici. Consultați paginile info **bzip2** pentru mai multe informații.

Pachetele de programe pentru Linux sunt distribuite adesea în arhive tarball create cu gzip. După despachetarea unor astfel de arhive, primul lucru pe care ar trebui să-l faceți este să citiți fișierul README. El conține, în general, informații despre instalarea programului în cauză.

Comanda **tar** GNU recunoaște fișierele create cu gzip. Folosiți comanda

tar zxvf fișier.tar.gz

pentru despachetarea fișierelor .tar.gz sau .tgz. Folosiți

tar jxvf fișier.tar.bz2

pentru despachetarea arhivelor **tar** care au fost împachetate cu **bzip2**.

9.1.1.4. Arhivele Java

Proiectul GNU pune la dispoziție unealta **jar** pentru crearea arhivelor Java. Este o aplicație Java care combină mai multe fișiere într-un singur fișier arhivat JAR. În timp ce poate fi folosită în scopuri generale, legate de arhivare, ca unealtă de comprimare, bazată pe formatele ZIP și ZLIB, **jar** a fost gândit în principal ca o unealtă care să șureze împachetarea codului Java, a applet-urilor și/sau aplicațiilor într-un singur fișier. Combinată într-o singură arhivă, componentele unei aplicații Java pot fi descărcate mult mai repede.

Spre deosebire de **tar**, **jar** arhivează implicit, independent de alte unelte - deoarece este, de fapt, versiunea Java a **zip**. În plus, permite ca intrările individuale să fie semnate de autor, astfel încât pot fi autentificate originile arhivei.

Sintaxa este aproape identică cu cea a comenzi **tar**, pentru diferențele specifice consultați [info jar](#).



tar, jar și legăturile simbolice

O notă importantă care nu este inclusă în documentația standard este aceea că **jar** urmărește

legăturile simbolice. Datele spre care țințesc aceste legături vor fi incluse în arhive. Comportamentul implicit al **tar** este să copieze doar legătura simbolică, dar poate fi schimbat cu opțiunea **-h** a aceleiași comenzi **tar**.

9.1.1.5. Transportarea datelor

Păstrarea copiilor datelor dumneavoastră pe un alt calculator (sau gazdă) este un mod simplu și precis de realizare a copiilor de siguranță. Consultați [Capitolul 10](#) pentru mai multe informații despre **scp**, **ftp** și altele.

În secțiunea următoare vom discuta despre dispozitivele locale de stocare.

9.2. Mutarea datelor pe un dispozitiv de stocare

9.2.1. Realizarea copiilor pe dischetă

9.2.1.1. Formatarea dischetei

Pe cele mai multe sisteme Linux, utilizatorii pot folosi discheta. Numele dispozitivului diferă, în funcție de tipul și numărul unităților de dischetă, dacă nu sunteți sigur, întrebați administratorul de sistem. Pe unele sisteme, există o legătură **/dev/floppy** care țințește dispozitivul în cauză, probabil **/dev/fd0** (dispozitiv detectat automat) sau **/dev/fd0H1440** (configurat pentru dischetele de 1,44 MO).

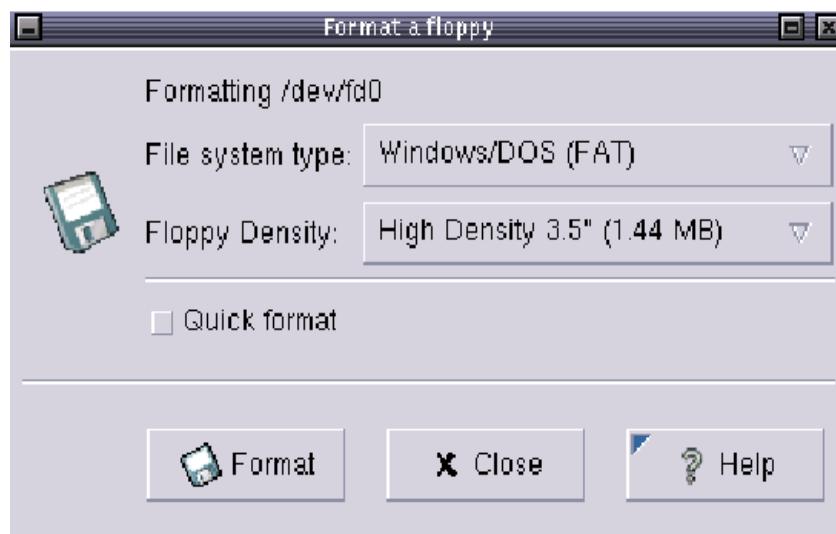
fdformat este unealta de bază pentru formatarea discheteelor. Ca opțiune are numele unității de dischetă. **fdformat** va afișa o eroare dacă discheta este protejată la scriere.

```
emma:~> fdformat /dev/fd0H1440
Double-sided, 80 tracks, 18 sec/track. Total capacity 1440 kB.
Formatting ... done
Verifying ... done
emma:~>
```

Comanda **mformat** (din pachetul **mtools**) este folosită pentru crearea discheteelor compatibile DOS care pot fi accesate prin comenziile **mcopy**, **mdir** și altele.

Sunt disponibile și unelte grafice.

Figura 9-1. Unealta grafică de formatare a discheteelor



După ce o dischetă este formatată, poate fi montată în sistemul de fișiere și accesată ca un director normal, mai mic, de regulă via **/mnt/floppy**.

Dacă aveți nevoie, instalați utilitarul **mkbootdisk**, care creează o dischetă de pe care sistemul

poate porni.

9.2.1.2. Folosirea comenzi **dd** pentru manevrarea datelor

Comanda **dd** poate fi folosită pentru mutarea datelor pe o dischetă sau la transferarea acestora înapoi pe discul fix, în funcție de dispozitivul de intrare și cel de ieșire folosit. Iată un exemplu:

```
gaby:~> dd if=images-without-dir.tar.gz of=/dev/fd0H1440
98+1 records in
98+1 records out

gaby~> dd if=/dev/fd0H1440 of=/var/tmp/images.tar.gz
2880+0 records in
2880+0 records out

gaby:~> ls /var/tmp/images*
/var/tmp/images.tar.gz
```

Mutarea datelor este făcută pe un dispozitiv nemontat. Dischetele create în acest mod nu pot fi montate în sistemul de fișiere, dar este metoda prin care se pot face dischete de pornire sau de recuperare. Mai multe informații despre posibilitățile **dd** pot fi găsite în paginile man.

Această unealtă este parte a pachetului GNU *coreutils*.



Transferul discurilor

Comanda **dd** poate fi folosită pentru realizarea transferului unui întreg disc fix brut.

9.2.2. Realizarea unei copii cu ajutorul unui inscriptor de CD-uri

Pe unele sisteme, le este permis utilizatorilor să folosească CD-writer-ul. Datele trebuie formatare mai întâi. Folosiți comanda **mkisofs** în directorul care conține fișierele pe care dorîți să le copiați. Verificați cu ajutorul **df** dacă aveți destul spațiu disponibil, deoarece va fi creat un fișier nou de o dimensiune aproximativ egală cu cea a directorului curent:

```
[rose@blob recordables] df -h .
Filesystem           Size   Used Avail Use% Mounted on
/dev/hde5            19G   15G   3.2G  82% /home

[rose@blob recordables] du -h -s .
325M   .

[rose@blob recordables] mkisofs -J -r -o cd.iso .
<--snap-->
making a lot of conversions
<--/snap-->
98.95% done, estimate finish Fri Apr  5 13:54:25 2002
Total translation table size: 0
Total rockridge attributes bytes: 35971
Total directory bytes: 94208
Path table size(bytes): 452
Max brk space used 37e84
166768 extents written (325 Mb)
```

Opțiunile **-J** și **-r** sunt folosite pentru a face posibilă montarea CD-ROM-ului în diferite sisteme de fișiere, după cum puteți afla din paginile man. După aceasta, CD-ul poate fi creat cu unealta **cdrecord**, folosind opțiunile potrivite:

```
[rose@blob recordables] cdrecord -dev 0,0,0 -speed=8 cd.iso
Cdrecord 1.10 (i686-pc-linux-gnu) (C) 1995-2001 Joerg Schilling
scsdevice: '0,0,0'
scsibus: 0 target: 0 lun: 0
```

```

Linux sg driver version: 3.1.20
Using libscg version 'schily-0.5'
Device type      : Removable CD-ROM
Version         : 0
Response Format: 1
Vendor_info     : 'HP          '
Identification  : 'CD-Writer+ 8100 '
Revision        : '1.0g'
Device seems to be: Generic mmc CD-RW.
Using generic SCSI-3/mmc CD-R driver (mmc_cdr).
Driver flags    : SWAUDIO
Starting to write CD/DVD at speed 4 in write mode for single session.
Last chance to quit, starting real write in 0 seconds.
Operation starts.

```

În funcție de inscriptorul dumneavoastră, aveți timp să fumați o țigară sau, mai bine și mai sănătos, să mâncăți o fructă și/sau să serviți o ceașcă de cafea. La terminarea sarcinii, veți primi un mesaj de confirmare:

```

Track 01: Total bytes read/written: 341540864/341540864
           (166768 sectors).

```

Există și unelte grafice care vă fac viața mai ușoară. Mai cunoscute sunt **xcdroast**, care este disponibil pe saitul [X-CD-Roast](#), fiind inclus în mai multe sisteme și în directorul GNU. Atât KDE cât și Gnome vă pun la dispoziție programe pentru crearea CD-urilor dumneavoastră.

9.2.3. Copii de siguranță pe/de pe unități jazz, dispozitive USB și alte medii externe

Aceste dispozitive sunt, de regulă, montate în sistemul de fișiere. După procedura de montare, ele sunt accesate ca directoare normale, aşa că puteți folosi comenziile obișnuite de manipulare a fișierelor.

În exemplul de mai jos, sunt copiate imagini de pe o cameră USB pe discul fix:

```

robin:~> mount /mnt/camera
robin:~> mount | grep camera
/dev/sdal on /mnt/camera type vfat (rw,nosuid,nodev)

```

Dacă aparatul foto este singurul dispozitiv de stocare USB pe care-l conectați la sistem, este în regulă. Dar rețineți că dispozitivele USB au intrări în directorul `/dev` odată ce sunt conectate la sistem. Astfel dacă introduceți în sistem un stick USB, acesta va figura ca intrare în `/dev/sda`, iar dacă atașați aparatul foto după acel stick, va figura ca intrare în `/dev/sdb` - în situația în care nu aveți discuri SCSI, care se găsesc tot în `/dev/sd*`. Pe sistemele mai noi, de la kernelul 2.6, un sistem de gestionare a conectărilor, numit HAL (Hardware Abstraction Layer) se asigură că utilizatorii nu se mai lovesc de aceste probleme. Dacă doriți să verificați unde se află dispozitivul dumneavoastră, consultați comanda **dmesg** după conectarea acestuia.

Puteți apoi copia fișierele:

```

robin:~> cp -R /mnt/camera/* images/
robin:~> umount /mnt/camera

```

În același mod, un dispozitiv jazz poate fi montat în `/mount/jazz`.

Trebuie să adăugați liniile necesare în `/etc/modules.conf` și `/etc/fstab` pentru ca sistemul să funcționeze bine. Căutați în documentația specifică dispozitivelor pentru mai multe informații. Pe sistemele în care versiunile kernelului sunt de la 2.6.x în sus, verificați și paginile de manual ale **modprobe** și fișierul `modprobe.conf`.

9.2.4. Realizarea copiilor de siguranță pe unități de bandă

Se face prin comanda **tar** (consultați secțiunile de mai sus). Unealta **mt** este folosită pentru controlul dispozitivelor cu bandă magnetică, cum ar fi `/dev/st0`. Au fost scrise cărți întregi despre realizarea copiilor de siguranță pe bandă magnetică, aşa că vă invităm să citiți lista din [Appendix B](#) pentru mai multe informații. Bazele de date pot necesita alte proceduri pentru realizarea copiilor de siguranță, din cauza arhitecturii acestora.

Comenzile corecte pentru crearea copiilor de siguranță sunt puse, de regulă, într-un director `cron`, pentru a fi executate în mod regulat. În medii mai mari, suitea [Amanda](#) sau alte soluții comerciale sunt folosite la realizarea copiilor de siguranță pe mai multe calculatoare deodată. Lucrul cu benzile însă, ca îndatorire a administratorului de sistem, este dincolo de scopul acestui ghid.

9.2.5. Unelte ale distribuției dumneavoastră

Cele mai multe distribuții pun la dispoziție propriile unelte pentru a vă ușura munca. Iată o scurtă trecere în revistă:

- SuSE: YaST include acum module pentru realizarea de copii de siguranță și restaurarea acestora.
 - RedHat: File Roller asigură administrarea vizuală a arhivelor (comprimate). Se pare că se preferă unealta X-CD-Roast pentru mutarea copiilor de siguranță pe dispozitive externe.
 - Mandrake: X-CD-Roast.
 - Cele mai multe distribuții includ utilitarele **dump** și **restore** pentru realizarea copiilor de siguranță pentru sistemele de fișiere `ext2` și `ext3`. Aceste programe poartă scris pe o gamă largă de dispozitive externe și, literal, aruncă fișierul, fișierele sau sistemul de fișiere bit cu bit pe dispozitivul specificat. Precum **dd**, vă permite realizarea copiilor de siguranță pentru fișiere speciale, cum ar fi cele din `/dev`.
-

9.3. Utilizarea rsync

9.3.1. Introducere

Programul **rsync** este o unealtă rapidă și flexibilă care vă permite realizarea copiilor de siguranță la distanță. Este comună sistemelor UNIX și celor asemănătoare UNIX-ului, ușor de configurat și de folosit în scripturi. Doar pentru că „r” din **rsync** vine de la „remote” nu înseamnă neapărat că nu puteți folosi programul decât în rețea. Dispozitivul dumneavoastră „la distanță” poate fi un mediu de stocare extern pe USB sau o altă partitură pe discul fix, aşa că nu trebuie să aveți două calculatoare pentru a vă folosi de acest program.

9.3.2. Un exemplu: rsync pe un dispozitiv de stocare USB

După cum am discutat în [Secțiunea 3.1.2.3](#), trebuie mai întâi să montați dispozitivul. Este posibil să fie nevoie de contul de administrator pentru această operație:

```
root@theserver# mkdir /mnt/usbstore  
root@theserver# mount -t vfat /dev/sdal /mnt/usbstore
```



Ușor de folosit

Din ce în ce mai multe distribuții permit accesul la dispozitivele externe pentru utilizatorii obișnuiți și realizează automat montarea dispozitivelor USB, CD-ROM și a altora de acest gen.

Acest ghid presupune existența asistenței USB în sistemul dumneavoastră. Consultați [Ghidul USB](#) pentru ajutor în cazul în care nu vă funcționează ceva din cele prezentate. Verificați cu `dmesg` dacă `/dev/sdal` este, într-adevăr, dispozitivul de montat.

După aceea, puteți începe realizarea copiei de siguranță propriu-zise, de exemplu copia directorului `/home/karl`:

```
karl@theserver:~> rsync -avz /home/karl/ /mnt/usbstore
```

Ca de obicei, vă recomandăm paginile man pentru mai multe informații.

9.4. Criptarea

9.4.1. Generalități

9.4.1.1. De ce ar trebui să vă criptați datele?

Criptarea este sinonimă cu secretizarea. Legat de copiile de siguranță, criptarea este folositoare în cazurile în care trebuie să vă păstrați datele în locuri în care nu puteți controla accesul, cum ar fi serverele furnizorului de internet.

În afară de asta, criptarea poate fi aplicată și E-mail-urilor: în mod normal, mesajele de poștă electronică nu sunt criptate și ele sunt trimise în clar pe rețeaua Internetului. Dacă mesajele pe care le trimiteți conțin informații sensibile, este mai bine să le criptați.

9.4.1.2. GNU Privacy Guard

Pe sistemele Linux veți găsi GnuPG, GNU Privacy Guard, care este o colecție de programe care sunt compatibile cu programul PGP (Pretty Good Privacy), disponibil comercial.

În acest ghid vom prezenta modul de utilizare de bază al uneltelelor de criptare și vom arăta lucrurile de care aveți nevoie pentru generarea unei chei de criptare și cum să o folosiți pentru criptarea datelor dumneavoastră, pe care le puteți păstra apoi în locuri publice. Moduri de folosire mai avansate pot fi găsite în paginile manualelor mai multor comenzi.

9.4.2. Generarea unei chei

Înainte de a începe criptarea datelor, trebuie să creați o pereche de chei. Perechea constă într-o cheie publică și una privată. Puteți trimite cheia publică corespondenților, care o vor folosi pentru criptarea datelor pentru dumneavoastră, pe care le decriptați cu cheia privată. Întotdeauna trebuie să păstrați cheia privată, să nu o puneti nimănui la dispoziție, deoarece în acest mod, datele destinate numai dumneavoastră pot fi decriptate de terțe părți. Pentru a vă asigura că nu apare nici un accident, cheia privată este protejată de o parolă. Perechea de chei este generată prin comanda:

```
willy@ubuntu:~$ gpg --key-gen
gpg (GnuPG) 1.4.2.2; Copyright (C) 2005 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

gpg: directory `/home/willy.gnupg' created
gpg: new configuration file `/home/willy/.gnupg/gpg.conf' created
gpg: WARNING: options in `/home/willy/.gnupg/gpg.conf' are not yet
active during this run
gpg: keyring `/home/willy/.gnupg/secring.gpg' created
gpg: keyring `/home/willy/.gnupg/pubring.gpg' created
Please select what kind of key you want:
 (1) DSA and Elgamal (default)
 (2) DSA (sign only)
 (5) RSA (sign only)
Your selection? 1
DSA keypair will have 1024 bits.
ELG-E keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048) 4096
Requested keysize is 4096 bits
```

```

Please specify how long the key should be valid.
  0 = key does not expire
  <n>  = key expires in n days
  <n>w = key expires in n weeks
  <n>m = key expires in n month
  <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y

You need a user ID to identify your key; the software constructs the
user ID from the Real Name, Comment and Email Address in this form:
  "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Willy De Wandel
Email address: wdw@mvg.vl
Comment: Willem
You selected this USER-ID:
  "Willy De Wandel (Willem) <wdw@mvg.vl>"

Change (N)ame, (C)omment, (E)mail or (O)key/(Q)uit? o
You need a Passphrase to protect your secret key.

Passphrase:

```

În acest moment introduceți parola. Poate fi o frază, cu cât mai lungă, cu atât mai bine, singura condiție fiind aceea de a nu o uita niciodată. Pentru verificare, trebuie să introduceți din nou parola aleasă.

Perechea de chei este generată de un program care amestecă numere la întâmplare, în funcție, printre altele, de activitatea sistemului. Dacă porniți unele programe, mutați cursorul sau introduceți la întâmplare câteva caractere într-o fereastră terminal, șansele de a fi generat un număr care conține multe cifre diferite cresc, iar cheia va fi mult mai greu de spart.

9.4.3. Despre cheia dumneavoastră

După ce cheia a fost creată, veți primi un mesaj despre *amprentă*. Aceasta este o secvență de 40 de numere hexazecimale, atât de lungă pentru a preîntâmpina apariția aceleiași chei de două ori, pe orice calculator. Puteți fi siguri de unicitatea acestei secvențe. Forma scurtă a cheii este compusă din numele dumneavoastră, urmat de ultimele 8 numere hexazecimale.

Puteți obține informații despre cheia dumneavoastră astfel:

```
willy@ubuntu:~$ gpg --list-keys
/home/willy/.gnupg/pubring.gpg
-----
pub      1024D/BF5C3DBB 2006-08-08
          Willy De Wandel (Willem) <wdw@mvg.vl>
uid
sub      4096g/A3449CF7 2006-08-08
```

ID-ul cheii generate este „BF5C3DBB”. Puteți trimite acest ID și numele dumneavoastră unui *server de chei*, astfel încât să fie disponibile corespondenților dumneavoastră, pentru a o folosi la criptarea datelor care vă sunt destinate. Mai puteți trimite cheia publică direct persoanelor care au nevoie de ea. Partea publică a cheii dumneavoastră este seria lungă de numere pe care o vedeti prin opțiunea *--export* a comenzi **gpg**:

```
gpg --export -a
```

În această prezentare am presupus că aveți nevoie de cheia de criptare pentru a cripta și decripta date numai pentru folosința proprie. Citiți paginile man ale programului **gpg** dacă doriti să știți mai multe lucruri pe acest subiect.

9.4.4. Criptarea datelor

Acum puteți cripta o arhivă .tar sau una comprimată, înainte de a o salva drept copie de siguranță pe un mediu de stocare sau pe un server pentru copii de siguranță. Folosiți comanda **gpg** precum în exemplul de mai jos:

```
gpg -e -r (partea a) uid arhivă
```

Opțiunea **-e** spune programului **gpg** să cripteze, opțiunea **-r** indică pentru cine să cripteze. Rețineți că doar utilizatorul care este arătat după opțiunea **-r** poate decripta datele. Un exemplu:

```
willy@ubuntu:~$ gpg -e -r Willy /var/tmp/home-willy-20060808.tar
```

9.4.5. Decriptarea fișierelor

Prin opțiunea **-d** a comenzi **gpg** puteți decripta datele care au fost criptate pentru dumneavoastră. Datele se vor derula pe ecran, dar o copie criptată va rămâne pe disc. Pentru alte formate decât textul simplu trebuie să salvați datele decriptate, pentru ca să le puteți deschide cu programul potrivit. Acest lucru se face prin opțiunea **-o** a comenzi **gpg**:

```
willy@ubuntu:~$ gpg -d -o /var/tmp/home-willy-decrypt.tar /var/tmp/home-willy-20060808.tar.gpg  
You need a passphrase to unlock the secret key for  
user: "Willy De Wandel (Willem) <wdw@mvg.vl>"  
4096 ELG-E key, ID A3449CF7, created 2006-08-08 (main key ID BF5C3DBB)  
gpg: encrypted with 4096-bit ELG-E key, ID A3449CF7, created 2006-08-08  
      "Willy De Wandel (Willem) <wdw@mvg.vl>"
```



Lipsă parolă = date pierdute

Dacă nu vă amintiți parola, datele sunt pierdute. Nici măcar administratorul de sistem nu va putea decripta datele dumneavoastră. Iată de ce o copii ale cheilor importante și parole sunt păstrate în seifuri ale băncilor.

9.5. Sumar

V-am pregătit o listă care conține comenziile folosite în realizarea copiilor de siguranță:

Tabelul 9-1. Comenzi noi în Capitolul 9: Copiile de siguranță

Comandă	Înțelesul comenzi
bzip2	Un compresor de fișiere sortator de blocuri.
cdrecord	Înregistrare de date sau audio pe CD.
dd	Convertește și copiază un fișier.
fdformat	Formatarea comună a unei dischete.
gpg	Criptează și decriptează datele.
gzip	Comprimă și despachetează fișiere.
mcopy	Copiază fișiere MSDOS pe/de pe UNIX.
mdir	Afișează un director MSDOS.
mformat	Adaugă un fișier MSDOS la o dischetă formatată comun.
mkbootdisk	Creează o dischetă de pornire a sistemului.
mount	Montează un sistem de fișiere (îl integrează în sistemul curent de fișiere prin atașarea la un punct de montare).

rsync	Sincronizează directoarele.
tar	Utilitar pentru arhivare pe unități de bandă, folosit și pentru realizarea copiilor de siguranță scrise pe discuri.
umount	Demontează sistemele de fișiere.

9.6. Exerciții

- Faceți o copie a directorului dumneavoastră home în `/var/tmp` folosind comanda **tar**. Comprimați apoi fișierul folosind **gzip** sau **bzip2**. Creați o arhivă curată, una care să nu se împrăștie peste tot la despachetare.
- Formatați o dischetă și puneti pe ea câteva fișiere din director dumneavoastră personal. Schimbați discheta cu una care aparține altuia și copiați fișierele conținute de acea dischetă în directorul personal.
- Formatați o dischetă în DOS. Folosiți *mtools* pentru a copia și a șterge fișiere pe aceasta.
- Ce se întâmplă cu o dischetă neformatată atunci când doriți să o montați în sistemul de fișiere?
- Dacă dețineți un dispozitiv de stocare usb, încercați și copiați niște fișiere pe acesta.
- Folosindu-vă de **rsync**, faceți o copie a directorului dumneavoastră personal pe un sistem de fișiere local sau din rețea.
- Atunci când doriți să păstrați fișiere pe servere din rețea, este bine ca aceste fișiere să fie criptate. Realizați o arhivă tar a directorului personal și criptați-o.

Capitolul 10. Rețele

Când vine vorba de rețea, Linux-ul este sistemul de operare care oferă cele mai multe avantaje, nu doar pentru că rețeaua este integrată intim în sistemul de operare propriu-zis și sunt puse la dispozitie multe unele și aplicații în mod gratuit, ci pentru robustețea dovedită sub trafic intens, obținută după ani de testare și rezolvare a problemelor raportate în cadrul proiectului Open Source (Sursă deschisă).

Rafturi întregi de cărți pline de informații au fost scrise despre Linux și rețelistică, noi încercând doar să oferim o privire de ansamblu pe marginea acestui subiect. După terminarea acestui capitol, veți cunoaște mai multe despre:

- ◆ Protocole de rețea admise
- ◆ Fișierele de configurare a rețelei
- ◆ Comenzile pentru configurarea și probarea rețelei
- ◆ Serviciile și programele client care permit aplicații diverse de rețea
- ◆ Partajarea de fișiere și tipărire
- ◆ Executarea de la distanță a unor comenzi și aplicații
- ◆ Notiuni de bază despre conexiunile de rețea
- ◆ Executarea securizată a aplicațiilor la distanță
- ◆ Firewall-uri și detectarea intrușilor

10.1. Generalități despre rețele

10.1.1. Modelul OSI

Un protocol este, la bază, un set de reguli pentru comunicare.

Pentru a transmite date în rețea, de exemplu un mesaj de poștă electronică de la calculatorul dumneavoastră către un alt calculator aflat la capătul lumii, mai multe dispozitive și programe trebuie să lucreze împreună.

Toate aceste dispozitive și programe folosesc limbaje diferite. Să luăm în considerare programul dumneavoastră de E-mail: este capabil să converseze cu sistemul de operare al calculatorului, printr-un protocol specific, dar nu poate comunica cu dispozitivele în sine. Aveți nevoie de un program special în sistemul de operare, care să facă acest lucru. Pe de altă parte, calculatorul trebuie să comunice cu linia telefonică sau alt sistem de conectare la Internet. În afară de ceea ce se vede, conexiunile de rețea au nevoie de comunicare pentru a transmite mesajul dumneavoastră de poștă electronică pe tot traseul către calculatorul de destinație.

Toate aceste protocole diferite de comunicare sunt clasificate în 7 nivele, care sunt cunoscute ca *Open Systems Interconnection Reference Model*, pe scurt *OSI Model*. Pentru a fi înțeles mai ușor, acest model este redus la un sistem de protocole în 4 nivele, descris în tabelul de mai jos:

Numele nivelului	Protocolele nivelului
Nivelul aplicațiilor	HTTP, DNS, SMTP, POP,...
Nivelul de transport	TCP, UDP
Nivelul de rețea	IP, IPv6
Nivelul de acces la rețea	PPP, PPPoE, Ethernet

Fiecare nivel poate folosi funcționalitățile nivelului aflat dedesubtul său; fiecare nivel își poate exporta funcționalitățile nivelului aflat deasupra sa. Cu alte cuvinte: un nivel poate comunica doar cu nivelurile adiacente. Să luăm din nou exemplul mesajului dumneavoastră de poștă electronică: îl introduceți prin nivelul aplicațiilor. În calculatorul dumneavoastră, el călătorește prin nivelurile de transport și rețea. Calculatorul îl transmite în rețea prin nivelul de acces la rețea. Tot acesta este și nivelul care-l poartă în jurul lumii. La destinație, calculatorul de destinație îl preia prin nivelul propriu de rețea și îl va afișa cu ajutorul nivelurilor de transport și aplicații.



În realitate lucrurile sunt mult mai complicate

Secțiunea de mai sus și cea următoare sunt incluse deoarece mai devreme sau mai târziu vă veți lovi de anumite termeni proprii rețelisticii; aceste secțiuni vă vor ajuta să vă consolidați un punct de plecare, dacă trebuie să aflați mai multe detalii pe marginea acestui subiect.

10.1.2. Câteva protocoale de rețea mai cunoscute

Linux-ul recunoaște mai multe protocoale de rețea. Vom prezenta pe cele mai importante:

10.1.2.1. TCP/IP

The Transport Control Protocol și Internet Protocol sunt două din cele mai populare căi de comunicare pe Internet. Multe aplicații, precum navigatorul dumneavoastră și programele de poștă electronică sunt realizate pe baza acestei suite de protocoale.

La bază, IP asigură soluții pentru trimitera pachetelor de informații de la un calculator la altul, în timp ce TCP aranjează pachetele în fluxuri, astfel încât pachetele diferitelor aplicații să nu se amestecă și să fie primite la destinație în ordinea corectă.

Un punct de plecare în studierea TCP și IP este reprezentat de următoarele lucrări:

- **man 7 ip**: descrie implementarea protocolului IPv4 în Linux (versiunea 4 fiind cea mai des întâlnită ediție a acestui protocol).
- **man 7 tcp**: implementarea protocolului TCP.
- RFC793, RFC1122, RFC2001 pentru TCP, iar pentru IP, RFC791, RFC1122 și RFC1112.

Documentele [Request For Comments](#) conțin descrierea standardelor de rețea, a protocoalelor, aplicațiilor și implementările acestora. Documentele sunt administrate de Internet Engineering Task Force, o comunitate internațională care răspunde de operaționalitatea Internetului precum și de evoluția și dezvoltarea arhitecturii Internetului.

Furnizorul dumneavoastră de Internet are o arhivă disponibilă, sau puteți consulta aceste RFC-uri via <http://www.ietf.org/rfc.html>.

10.1.2.2. TCP/IPv6

Nimeni nu se aștepta ca Internetul să crească în ritmul pe care-l cunoaștem azi. S-a dovedit că IP are anumite dezavantaje atunci când un număr foarte mare de calculatoare sunt în rețea, cel mai mare fiind acela legat de alocarea unei adrese unice fiecărui calculator existent în rețea. De aceea, IP versiunea 6 a fost destinată să răspundă nevoilor actuale ale Internetului.

Din nefericire, nu toate aplicațiile și serviciile pot lucra încă cu IPv6. Migrarea este în desfășurare în cazul multor aplicații și servicii care pot beneficia de pe urma implementării protocolului amintit. Câteva aplicații folosesc vechiul protocol, iar pentru cele care au fost reproiectate noua versiune este activă. În momentul în care verificăți configurația rețelei, lucrurile nu sunt aşa de clare, întrucât au fost luate tot felul de măsuri pentru a împiedica amestecarea conexiunilor, prin ascunderea unui protocol de celălalt.

Mai multe informații pot fi găsite în următoarele documente:

- **man 7 ipv6**: implementarea protocolului Linux IPv6.
 - RFC1883 care descrie protocolul IPv6.
-

10.1.2.3. PPP, SLIP, PLIP, PPOE

Kernelul Linux integrează protocoalele PPP (Point-to-Point-Protocol), SLIP (Serial Line IP), PLIP (Parallel Line IP) și PPP Over ETHERNET. PPP este cea mai folosită cale prin care utilizatorii individuali accesează furnizorul de Internet (ISP - Internet Service Provider), deși în zonele des populate acesta este

înlocuit de PPPOE, protocol folosit în cazul conexiunilor ADSL (Asymmetric Digital Subscriber Line).

Cele mai multe distribuții de Linux au încorporat unelte ușor de folosit pentru configurarea conexiunii la Internet. Ceea ce trebuie să cunoașteți este numele de utilizator și parola furnizate de ISP-ul dumneavoastră, iar în cazul PPP un număr de telefon. Aceste date sunt introduse într-o aplicație care rulează în modul grafic, care vă permite și pornirea sau oprirea conexiunii la Internet.

10.1.2.4. ISDN

Kernelul Linux-ului include și tratarea protocolului ISDN. Isdn4linux controlează plăcile ISDN PC și poate emula un modem cu setul de comenzi Hayes (comenzile „AT”). Posibilitățile acoperă simpla utilizare a unui program terminal cât și o conexiune totală la Internet.

Verificați în documentația sistemului dumneavoastră.

10.1.2.5. AppleTalk

Appletalk este numele sistemului de conectivitate al calculatoarelor Apple. Permite o rețea peer-to-peer cu funcționalități de bază cum ar fi partajarea de fișiere și imprimante. Fiecare calculator se poate comporta în același timp ca server sau client, iar programele și dispozitivele necesare sunt incluse în orice calculator Apple.

Linux-ul asigură implementarea totală a AppleTalk. Netatalk este implementarea la nivelul kernelului a protocolului AppleTalk Protocol Suite, fiind la bază destinat sistemelor BSD și derivate. Integrează rutarea AppleTalk, care deservește sistemele de fișiere UNIX și AFS care utilizează AppleShare și imprimantele UNIX precum și pe cele AppleTalk.

10.1.2.6. SMB/NMB

Pentru compatibilitatea cu mediul MS Windows, suita Samba, care include protocolele NMB și SMB, poate fi instalată pe orice sistem compatibil UNIX. Protocolul Server Message Block (cunoscut și drept Session Message Block, NetBios sau LanManager Protocol), este folosit pe sistemele MS Windows 3.11, NT, 95/98, 2K și XP pentru partajarea discurilor și imprimantelor.

Funcționalitățile de bază ale suitei Samba sunt: partajarea discurilor Linux cu cele ale Windows-ului, accesarea SMB de pe calculatoarele Linux, accesarea imprimantelor Linux de către calculatoarele Windows și accesarea imprimantelor Windows de către sisteme Linux.

Cele mai multe distribuții de Linux includ un pachet *samba*, care configuraază cea mai mare parte a serverului și pornește **smbd**, serviciul Samba, și **nmbd**, serviciul netbios, implicit, la pornirea calculatorului. Samba poate fi configurată în modul grafic, cu ajutorul unei interfețe web sau în modul liniei de comandă, cu ajutorul fișierelor text de configurare. Serviciul face ca un calculator Linux să apară drept o gazdă MS Windows în fereastra My Network Places/Network Neighbourhood proprii mediilor MS Windows; partajarea unui sistem Linux nu poate fi distinsă de oricare altă gazdă din mediul MS Windows.

Obțineți mai multe informații din următoarele surse:

- **man smb.conf**: descrie alcătuirea principalului fișier de configurare pentru serviciul Samba.
 - [The Samba Project Documentation](#) (sau căutați saitul-oglindă local al samba.org) conține un ghid ușor de înțeles pentru instalare și testare, care explică și modul în care puteți configura serviciul Samba ca Primary Domain Controller (Controlor Primar de Domenii). Pe acest sait sunt disponibile și toate paginile de manuale.
-

10.1.2.7. Alte protocole

Linux-ul se poate descurca și cu Amateur Radio, WAN internetworking (X25, Frame Relay, ATM), InfraRed și alte conexiuni fără fir, dar, întrucât aceste protocole necesită echipamente dedicate, nu le vom discuta în această lucrare.

10.2. Configurarea rețelei și informații despre rețea

10.2.1. Configurarea interfețelor de rețea

Toate distribuțiile principale de Linux, prietenoase cu utilizatorul, vin cu diferite unelte grafice care permit configurarea calculatorului pentru o rețea locală, conectarea la un furnizor de servicii Internet sau accesarea unei rețele wireless (fără fir). Aceste unelte pot fi pornite din linia de comandă sau din meniu:

- Configurarea Ubuntu este făcută prin selectarea System -> Administration -> Networking (Sistem -> Administrare -> Utilitare de rețea).
- RedHat Linux include **redhat-config-network**, care dispune atât de interfață în modul grafic, cât și de cea text.
- YAST sau YAST2 din Suse sunt o unealtă de configurare all-in-one.
- Mandrake/Mandriva vine cu un utilitar numit Network and Internet Configuration Wizard, pe care vă sfătuim să-l porniți din Mandrake's Control Center (Centrul de Control al Mandrake)
- Pe sistemele Gnome: **gnome-network-preferences**.
- Pe sistemele KDE: **knetworkconf**.

Documentația sistemului pe care lucrați conține multe sfaturi și informații despre disponibilitatea și modul de folosire a uneltelor specifice.

Informații pe care trebuie să le furnizați:

- Pentru conectarea la rețea locală, de exemplu la calculatoarele de acasă sau de la serviciu: numele calculatorului, numele domeniului și adresa de IP. Dacă doriți să vă configurați o rețea locală, este bine să studiați mai mult pentru început. La serviciu, informațiile pe care le-am menționat sunt furnizate calculatorului, de cele mai multe ori, la pornirea acestuia. Dacă nu sunteți sigur, mai bine nu introduceți nici o informație.
- Pentru conectarea la Internet: numele de utilizator și parola furnizate de ISP-ul dumneavoastră, numărul de telefon dacă folosiți un modem pentru linia telefonică. Furnizorul dumneavoastră de Internet vă alocă, de regulă automat, o adresă de IP și toate cele necesare pentru ca aplicațiile dumneavoastră legate de Internet să lucreze cum trebuie.

10.2.2. Fișierele de configurare ale rețelei

Așistentul grafic modifică un set specific de fișiere de configurare ale rețelei, folosind câteva comenzi de bază. Numele exact al fișierelor de configurare și localizarea lor în sistemul de fișiere depind în mare măsură de distribuția și versiunea de Linux pe care o folosiți. Totuși, există fișiere de configurare ale rețelei care sunt întâlnite pe toate sistemele asemănătoare UNIX:

10.2.2.1. /etc/hosts

Fișierul `/etc/hosts` conține întotdeauna adresa locală IP a calculatorului, 127.0.0.1, folosită de comunicațiile interprocese. Nu ștergeți această linie niciodată! Câteodată conține adrese ale altor calculatoare, care pot fi contactate fără a fi nevoie de folosirea unui serviciu extern de nume, precum DNS (Domain Name Server).

O moștră de fișier `hosts` al unei mici rețele este următoarea:

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1      localhost.localdomain    localhost
192.168.52.10  tux.mylan.com          tux
192.168.52.11  winxp.mylan.com        winxp
```

Citiți mai multe în **man hosts**.

10.2.2.2. /etc/resolv.conf

Fisierul `/etc/resolv.conf` controlează accesul la un server DNS, consultați [Secțiunea 10.3.7](#). Acest fișier conține numele dumneavoastră de domeniu și numele serverului (serverelor) ce trebuie contactate:

```
search mylan.com  
nameserver 193.134.20.4
```

Citiți mai multe în paginile man ale `resolv.conf`.

10.2.2.3. /etc/nsswitch.conf

Fisierul `/etc/nsswitch.conf` definește ordinea în care sunt contactate numele diferite de servicii. Pentru utilizarea Internetului, este important ca `dns` să apară în linia „hosts”:

```
[bob@tux ~] grep hosts /etc/nsswitch.conf  
hosts: files dns
```

Prin aceasta se dau instrucțiuni calculatorului să caute numele gazdelor și adresele de IP începând cu fișierul `/etc/hosts` și să contacteze serverul DNS dacă o anume gazdă nu apare în acest fișier. Alte nume posibile de servicii care pot fi listate sunt: LDAP, NIS și NIS+.

Ca de obicei, mai multe despre acest subiect găsiți în [man nsswitch.conf](#).

10.2.3. Comenzile de configurare ale rețelei

10.2.3.1. Comanda ip

Scripturile specifice distribuțiilor și uneltele grafice sunt doar interfețe ale comenzi `ip` (sau `ifconfig` și `route` pe sistemele mai vechi) care afișează și modifică configurația părții de rețea integrată în kernel.

Comanda `ip` este folosită pentru a aloca adrese IP interfețelor de rețea, pentru configurarea rutărilor pentru Internet și alte rețele, pentru afișarea configurărilor TCP/IP, și altele.

Următoarele comenzi afișează adresa de IP și informații legate de rutare:

```
benny@home benny> ip addr show  
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 brd 127.255.255.255 scope host lo  
        inet6 ::1/128 scope host  
            link-local brd ff00:00:00:00:00:00  
            valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100  
    link/ether 00:50:bf:7e:54:9a brd ff:ff:ff:ff:ff:ff  
    inet 192.168.42.15/24 brd 192.168.42.255 scope global eth0  
        inet6 fe80::250:bfff:fe7e:549a/10 scope link  
            link-local brd ff00:00:00:00:00:00  
            valid_lft forever preferred_lft forever  
  
benny@home benny> ip route show  
192.168.42.0/24 dev eth0  scope link  
127.0.0.0/8 dev lo  scope link  
default via 192.168.42.1 dev eth0
```

Lucruri de ținut minte:

- apar două interfețe de rețea, chiar pe calculatoarele care au doar o placă fizică de rețea: „lo” este bucla locală (local loop), folosită de comunicațiile interne; „eth0” este un nume comun pentru interfață reală. Nu schimbați niciodată configurația buclei locale, pentru că sistemul va funcționa incorrect. Interfețele fără fir sunt definite de regulă ca „wlan0”; modemurile ca „ppp0”, dar pot avea și alte nume.
- adresele de IP, marcate cu „inet”: bucla locală întotdeauna are adresa 127.0.0.1, în timp ce interfețele fizice au tot felul de combinații.

- adresele fizice ale echipamentelor, care pot fi solicitate ca parte a procedurii de autentificare în rețea, sunt marcate ca „ether”. Bucla locală are 6 perechi de zero, iar placa de rețea are 6 perechi de caractere hexazecimale, din care primele 3 perechi sunt specifice fabricantului.

10.2.3.2. Comanda ifconfig

Cu toate că **ip** este cea mai nouă cale de configurare a unui sistem Linux, **ifconfig** este încă folosit destul de des. Folosiți-l fără opțiuni pentru a afișa informații despre interfețele de rețea:

```
els@asus:~$ /sbin/ifconfig
eth0      Link encap:Ethernet HWaddr 00:50:70:31:2C:14
          inet addr:60.138.67.31 Bcast:66.255.255.255 Mask:255.255.255.192
          inet6 addr: fe80::250:70ff:fe31:2c14/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:31977764 errors:0 dropped:0 overruns:0 frame:0
            TX packets:51896866 errors:0 dropped:0 overruns:0 carrier:0
            collisions:802207 txqueuelen:1000
            RX bytes:2806974916 (2.6 GiB) TX bytes:2874632613 (2.6 GiB)
            Interrupt:11 Base address:0xec00

          inet addr:127.0.0.1 Mask:255.0.0.0
            inet6 addr: ::1/128 Scope:Host
              UP LOOPBACK RUNNING MTU:16436 Metric:1
              RX packets:765762 errors:0 dropped:0 overruns:0 frame:0
              TX packets:765762 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:0
              RX bytes:624214573 (595.2 MiB) TX bytes:624214573 (595.2 MiB)
```

Puteți observa și aici cele mai importante aspecte ale configurației interfeței:

- Adresa de IP este marcată cu „inet addr”.
- Adresa dispozitivului se află după eticheta „HWaddr”.

Ambele comenzi, **ifconfig** și **ip**, afișează mai în detaliu informații și date statistice despre fiecare interfață și, ceea ce este foarte important, dacă acestea sunt „UP” (pornite) și „RUNNING” (lucrează).

10.2.3.3. Comenzile PCMCIA

Pe laptopul dumneavoastră folosiți, de regulă, pentru conexiunea la rețeaua companiei, placa Ethernet, dar dacă sunteți puși în situația de a configura o conexiune acasă sau într-un hotel, este posibil să fie nevoie de activarea cardului (plăcii) PCMCIA. Puteți face acest lucru prin utilitarul de control **cardctl** sau **pccardctl** pe distribuțiile mai noi.

Iată un exemplu de utilizare:

cardctl insert

În acest moment puteți configura cardul, fie în modul grafic, fie prin interfața liniei de comandă. Înainte de a scoate cardul, folosiți comanda:

cardctl eject

O distribuție serioasă ar trebui să integreze în uneltele de configurare ale rețelei utilitare pentru administrarea cardurilor PCMCIA, nefiind nevoie de executarea manuală a acestor comenzi.

10.2.3.4. Mai multe informații

Discutarea mai multor chestiuni legate de configurația rețelei este dincolo de scopul acestui document. Sursa dumneavoastră primară pentru mai multe informații este reprezentată de paginile manualelor pentru serviciile pe care doriți să le configurați. Lecturi adiționale:

- [The Modem-HOWTO](#): Vă ajută la alegerea, conectarea, configurarea, rezolvarea problemelor și înțelegerea modemurilor în sistem analog pentru PC-uri.
- [LDP HOWTO Index, Secțiunea 4.4](#): administrează liste de HOWTO despre rețelistică în general, protocole, dial-up, DNS, VPN, bridging (punți), rutare, securitate și altele.
- Cele mai multe sisteme au o versiune a fișierului `ip-cref` (căutați-l prin comanda `locate`); formatul PS al acestui fișier poate fi vizualizat de exemplu cu `gv`.

10.2.4. Numele interfețelor de rețea

Pe un calculator Linux, numele dispozitivului `lo` sau al *buclei locale* (*local loop*) este legat de adresa internă 127.0.0.1. Calculatorul va fi pus în dificultate în momentul lansării aplicațiilor dacă acest dispozitiv nu este prezent; este mereu disponibil, chiar pe calculatoarele care nu sunt legate la rețea.

Primul dispozitiv ethernet, `eth0` în cazul unei interfețe fizice (placă) standard de rețea, țineste către adresa dumneavoastră locală LAN IP. Calculatoarele care sunt clienti obișnuiți au doar o placă de rețea. Ruterele (routers), care fac legătura între rețele, dispun de câte un dispozitiv de rețea pentru fiecare rețea deservită.

Dacă folosiți un modem pentru a vă conecta la Internet, probabil că acesta va fi denumit `ppp0`.

Există mult mai multe nume, de exemplu pentru interfețele virtuale private de rețea (VPN), iar mai multe interfețe pot fi active simultan, de aceea ieșirea comenziilor `ifconfig` sau `ip` poate deveni stufoasă dacă nu este folosită nici o opțiune. Pot fi active mai multe interfețe de același tip. În acest caz, ele vor fi denumite secvențial: prima va primi numărul 0, a doua sufixul 1, cea de-a treia va primi 2, și a.m.d. Cazul despre care am vorbit este întâlnit pe servere, pe sisteme care au o configurație redundantă, pe rutere, firewall-uri și multe altele.

10.2.5. Verificarea configurației calculatorului cu netstat

În afară de comanda `ip`, care afișează configurația rețelei, există comanda `netstat`, care dispune de multe opțiuni și este foarte folositoare pe sistemele UNIX.

Informațiile legate de rutare sunt afișate prin opțiunile `-nr` ale acestei comenzi:

```
bob:~> netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask        Flags MSS Window irtt Iface
192.168.42.0   0.0.0.0       255.255.255.0 U        40 0          0 eth0
127.0.0.0      0.0.0.0       255.0.0.0    U        40 0          0 lo
0.0.0.0         192.168.42.1  0.0.0.0    UG       40 0          0 eth0
```

În imaginea de mai sus este vorba despre un calculator, tipic client, într-o rețea IP. Are doar o placă de rețea, `eth0`. Interfața `lo` este bucla locală.



Metoda modernă

Metoda modernă de a obține aceeași informație este folosirea comenzi `ip`:

`ip route show`

Când acest calculator încearcă să contacteze o gazdă aflată în altă rețea decât cea proprie, indicată prin linia care începe cu 0.0.0.0., va trimite cererea de conectare calculatorului (ruterului) cu adresa de IP 192.168.42.1, și, pentru asta va folosi interfața primară, `eth0`.

Calculatoarele care sunt în aceeași rețea, linia care începe cu 192.168.42.0, vor fi contactate tot prin intermediul interfeței primare, dar nu mai este necesară rutarea, datele sunt puse direct pe rețea.

Calculatoarele pot avea tabele de rutare mult mai complicate decât acesta, cu diferite perechi „Destinations-Gateway” pentru conectarea cu diverse rețele. Dacă aveți ocazia de a vă conecta la un server de aplicații, de exemplu la locul de muncă, puteți verifica, în scop educațional, informațiile legate de rutare.

10.2.6. Alte gazde

O colecție impresionantă de unelte este destinată administrării de la distanță a calculatoarelor Linux. Saitul oglindă local vă oferă destul de multe astfel de unelte. Apelați la documentația specifică programelor care vă interesează, deoarece discutarea acestor aspecte nu face obiectul acestui ghid.

Vom discuta în această secțiune doar comenziile uneltelelor text foarte utilizate pe sistemele UNIX/Linux.

10.2.6.1. Comanda host

Pentru a afișa informații despre gazde sau domenii, folosiți comanda **host**:

```
[emmy@pc10 emmy]$ host www.eunet.be
www.eunet.be. has address 193.74.208.177

[emmy@pc10 emmy]$ host -t any eunet.be
eunet.be. SOA dns.eunet.be. hostmaster.Belgium.EU.net.
2002021300 28800 7200 604800 86400
eunet.be. mail is handled by 50 pophost.eunet.be.
eunet.be. name server ns.EU.net.
eunet.be. name server dns.eunet.be.
```

Informații similare pot fi afișate prin folosirea comenzi **dig**, care oferă informații adiționale despre cum sunt stocate înregistrările în serverul de nume.

10.2.6.2. Comanda ping

Pentru a verifica dacă un calculator este disponibil în rețea, folosiți **ping**. Dacă sistemul este configurat pentru a transmite mai mult de un pachet, întrerupeți **ping** prin combinația de taste **Ctrl+C**:

```
[emmy@pc10 emmy]$ ping a.host.be
PING a.host.be (1.2.8.3) from 80.20.84.26: 56(84) bytes of data.
64 bytes from a.host.be(1.2.8.3): icmp_seq=0 ttl=244 time=99.977ms
--- a.host.be ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/mdev = 99.977/99.977/99.977/0.000 ms
```

10.2.6.3. Comanda traceroute

Pentru a verifica ruta pe care o urmează un pachet prin rețea, folosiți comanda **traceroute**:

```
[emmy@pc10 emmy]$ /usr/sbin/traceroute www.eunet.be
traceroute to www.eunet.be(193.74.208.177), 30 hops max, 38b packets
1 blob (10.0.0.1)
    0.297ms  0.257ms  0.174ms
2 adsl-65.myprovider.be (217.136.111.1)
    12.120ms 13.058ms 13.009ms
3 194.78.255.177 (194.78.255.177)
    13.845ms 14.308ms 12.756ms
4 gigabitethernet2-2.int2.gam.brussels.skynet.be (195.238.2.226)
    13.123ms 13.164ms 12.527ms
5 pecbru2.car.belbone.be (194.78.255.118)
    16.336ms 13.889ms 13.028ms
6 ser-2-1-110-ias-be-vil-ar01.kpnbelgium.be (194.119.224.9)
    14.602ms 15.546ms 15.959ms
7 unknown-195-207-939.eunet.be (195.207.93.49)
    16.514ms 17.661ms 18.889ms
8 S0-1-0.Leuven.Belgium.EU.net (195.207.129.1)
    22.714ms 19.193ms 18.432ms
9 dukat.Belgium.EU.net (193.74.208.178) 22.758ms * 25.263ms
```

Pe unele sisteme, **traceroute** a fost redenumită **tracepath**.

10.2.6.4. Comanda whois

Informații specifice numelui de domeniu pot fi apelate prin comanda **whois**, după cum se explică de numeroasele servere **whois**, precum cel din exemplul de mai jos:

```
[emmy@pc10 emmy]$ whois cnn.com
[whois.crsnic.net]

Whois Server Version 1.3

$<--snap server message-->

Domain Name: CNN.COM
Registrar: NETWORK SOLUTIONS, INC.
Whois Server: whois.networksolutions.com
Referral URL: http://www.networksolutions.com
Name Server: TWDNS-01.NS.AOL.COM
Name Server: TWDNS-02.NS.AOL.COM
Name Server: TWDNS-03.NS.AOL.COM
Name Server: TWDNS-04.NS.AOL.COM
Updated Date: 12-mar-2002
>>> Last update of whois database: Fri, 5 Apr 2002 05:04:55 EST <<<
The Registry database contains ONLY .COM, .NET, .ORG, .EDU domains
and Registrars.

[whois.networksolutions.com]

$<--snap server message-->

Registrant:
Turner Broadcasting (CNN-DOM)
1 CNN Center
Atlanta, GA 30303

Domain Name: CNN.COM

Administrative Contact:
$<--snap contactinfo-->
Technical Contact:
$<--snap contactinfo-->
Billing Contact:
$<--snap contactinfo-->
Record last updated on 12-Mar-2002.
Record expires on 23-Sep-2009.
Record created on 22-Sep-1993.
Database last updated on 4-Apr-2002 20:10:00 EST.

Domain servers in listed order:

TWDNS-01.NS.AOL.COM      149.174.213.151
TWDNS-02.NS.AOL.COM      152.163.239.216
TWDNS-03.NS.AOL.COM      205.188.146.88
TWDNS-04.NS.AOL.COM      64.12.147.120
```

Pentru alte nume de domenii precum .com, .net, .org și .edu, s-ar putea să fie nevoie să specificați serverul whois, precum acesta pentru .be domains:

whois domain.be@whois.dns.be

10.3. Aplicații Internet/Intranet

Sistemele Linux sunt o platformă excelentă pentru serviciile de rețea. În această secțiune vom încerca să vă prezentăm cele mai cunoscute servicii și aplicații de rețea.

10.3.1. Tipuri de servicii

10.3.1.1. Servicii independente

Oferirea unui serviciu poate fi abordată în două moduri. Un serviciu (sau daemon) poate rula în modul independent sau poate depinde de activarea unui alt serviciu.

Serviciile de rețea care sunt utilizate intens și/sau continuu, rulează de regulă independent: sunt programe de sine stătătoare care rulează non-stop. Sunt inițiate adesea la pornirea calculatorului și așteaptă solicitări în puncte specifice de conexiune sau porturi pentru care sunt configurate să asculte. Când apare o cerere, este procesată și așteptarea continuă până la următoarea cerere. Un server web este un exemplu tipic: vă așteptați să fie disponibil 24 de ore pe zi, iar dacă este prea ocupat ar trebui create mai multe instanțe pentru a servi simultan mai mulți utilizatori. Alte exemple sunt depozitele mari de programe precum [Sourceforge](#) sau [your Tucows mirror](#), care trebuie să se descurce cu mii de cereri FTP pe zi.

Un exemplu de serviciu de sine stătător pe calculatorul dumneavoastră de acasă este **named** (serviciul de nume), un serviciu pentru schimbarea de nume. Serviciile independente au propriile procese care rulează, evidențiate de comanda **ps**:

```
bob:~> ps auxw | grep named
named    908  0.0  1.0 14876 5108 ?      S  Mar14  0:07 named -u named
```

Există totuși și servicii care sunt utilizate pe calculatorul dumneavoastră, chiar dacă nu există nici un proces ce rulează pentru acel serviciu. Exemple ar fi serviciul FTP, serviciul secure copy sau finger. Acestea beneficiază de serviciul Internet Daemon (**inetd**), care ascultă pentru ele.

10.3.1.2. (x)inetd

Pe calculatoarele de acasă, lucrurile sunt mai degrabă calme. Este posibil să aveți o mică rețea și să transferați fișiere dintr-un calculator pe altul, din când în când, folosind FTP sau Samba (pentru conectivitatea cu sistemele MS Windows). În aceste cazuri, pornirea tuturor serviciilor de care aveți nevoie doar ocazional și rularea acestora continuu este un lucru devorator de resurse. Așa că în configurațiile de bază veți găsi că serviciile de care aveți nevoie depind de un program central care ascultă toate porturile serviciilor de care este responsabil. Acest superserviciu, Internet service daemon, este pornit la inițializarea sistemului. Sunt două implementări de bază: **inetd** și **xinetd** (the extended Internet service daemon). Unul din ele rulează în mod normal pe orice sistem Linux:

```
bob:~> ps -ef | grep inet
root   926  1 0 Mar14 ?  00:00:00 xinetd-ipv6 -stayalive -reuse \
-pidfile /var/run/xinetd.pid
```

Serviciile pentru care este responsabil Internet daemon sunt listate în fișierul său de configurare, /etc/inetd.conf, pentru **inetd**, și în /etc/xinetd.d în cazul **xinetd**. Printre serviciile administrate se regăsesc serviciile de partajare de fișiere și imprimante, SSH, FTP, telnet, serviciul de configurare pentru Samba, serviciile talk și time.

De îndată ce o cerere de conexiune este primită, serviciul central va porni o instanță a serviciului solicitat. Astfel, în exemplul de mai jos, când utilizatorul *bob* pornește o sesiune FTP pe gazda locală, un serviciu FTP va rula atât timp cât sesiunea este activă:

```
bob:~> ps auxw | grep ftp
bob    793  0.1  0.2 3960 1076 pts/6      S     16:44  0:00 ncftp localhost
ftp    794  0.7  0.5 5588 2608 ?          SN     16:44  0:00 ftpd:
localhost.localdomain: anonymous/bob@his.server.com: IDLE
```

Desigur, același lucru se întâmplă când deschideți conexiuni la distanță: ori un anume serviciu răspunde direct, ori un serviciu la distanță (**x)inetd**) pornește serviciul de care aveți nevoie și îl oprește în momentul în care terminați treaba.

10.3.2. Poșta electronică

10.3.2.1. Serviciile

Sendmail este programul standard pentru poștă electronică sau Mail Transport Agent pentru platformele UNIX. Este solid, scalabil, și, dacă este configurat corect în combinație cu echipamentele potrivite, se descurcă cu mii de utilizatori fără nici un efort. Mai multe informații despre cum se configurează Sendmail sunt incluse în pachetele sendmail și sendmail-cf. Vă sfătuim să citiți și fișierele README și README.cf din /usr/share/doc/sendmail. De asemenea, trebuie citite paginile **man sendmail** și **man aliases**.

Qmail este alt serviciu de mail, care câștigă popularitate deoarece se prezintă a fi mai securizat decât Sendmail. În timp ce Sendmail este un program monolit, Qmail este compus din programe mai mici care interacționează, fiind mai ușor de securizat. Postfix este un alt serviciu de mail care câștigă teren printre utilizatori.

Aceste servicii administrează liste de mail, filtre, scanări de viruși și multe alte lucruri. Programe antivirus gratuite sau comerciale sunt disponibile pentru Linux. Exemple de programe pentru liste de poștă electronică sunt Mailman, Listserv, Majordomo și EZmlm. Consultați paginile web ale programului antivirus favorit pentru informații despre Linux și serviciile asigurate. Amavis și Spamassassin sunt implementări gratuite ale unor scanere antivirus și antispam.

10.3.2.2. Servicii de mail la distanță

Cele mai cunoscute protocoale pentru accesarea la distanță a serviciilor de poștă electronică sunt *POP3* și *IMAP4*. IMAP și POP permit operațiuni offline (în modul deconectat), accesul la distanță la mesajele noi și ambele se bazează pe serviciul SMTP pentru trimiterea de mesaje.

În timp ce POP este un protocol simplu, ușor de implementat și acceptat de aproape orice client de mail, este preferat IMAP deoarece:

- poate administra etichete persistente ale statutului mesajelor.
- poate stoca și aduce mesajele de poștă electronică.
- poate accesa și administra mai multe căsuțe de poștă electronică.
- administrează actualizări și căsuțe de poștă electronică comune.
- poate fi folosit la accesarea mesajelor Usenet și a altor documente.
- IMAP lucrează atât on-line, cât și deconectat.
- este optimizat pentru obținerea unor performanțe maxime on-line, în special pentru rețele de mică viteză.

10.3.2.3. Clienți de poștă electronică

Există multe programe pentru poștă electronică, atât pentru mediul grafic, cât și pentru cel text; noi vom arăta pe cele mai cunoscute. Alegeti-vă favoritul.

Comanda **mail** din UNIX este prezentă de ani buni, chiar înainte de existența rețelelor. Ea este o interfață simplă pentru trimiterea de mesaje și fișiere de mărime mică către alți utilizatori, care pot salva aceste mesaje, le pot redirecționa, răspunde, etc.

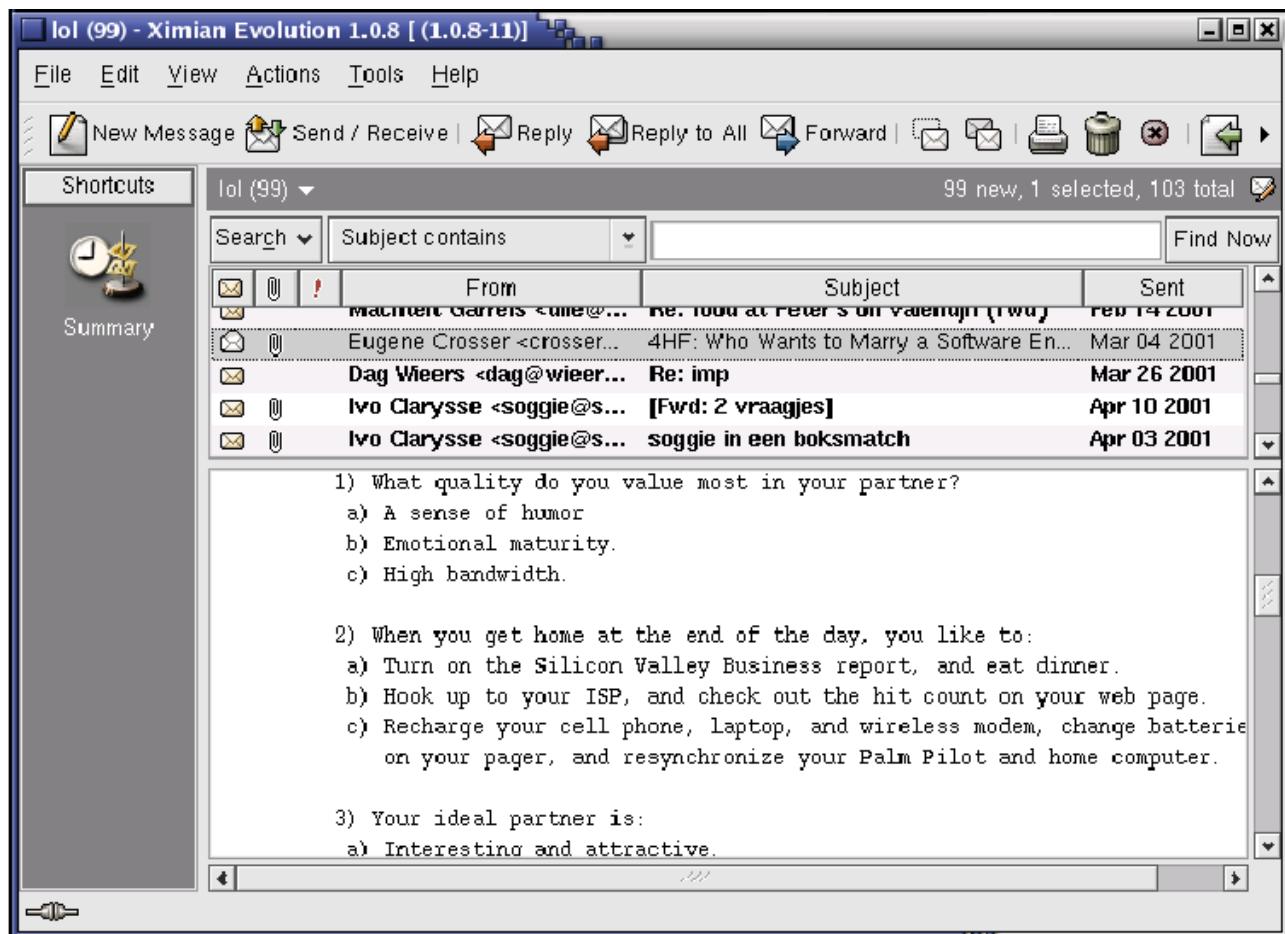
Cu toate că acest program nu mai este folosit drept client de mail în zilele noastre, el este totuși folositor, de exemplu pentru trimiterea ieșirii unei comenzi:

```
mail <future.employer@whereIwant2work.com> < cv.txt
```

Cititorul de mesaje de poștă electronică **elm** este o îmbunătățire de care **mail** avea nevoie, o astfel de îmbunătățire fiind și **pine** (Pine Is Not ELM). Programul **mutt** este chiar mai recent și oferă facilități noi.

Pentru acei utilizatori care preferă interfața grafică pentru poștă electronică există sute de opțiuni. Cel mai apreciat program de utilizatorii noi este Mozilla Mail/Thunderbird, care dispune de o opțiune anti spam ușor de configurat, și Evolution, clona programului MS Outlook. Kmail este apreciat de utilizatorii KDE.

Figura 10-1. Programul de poștă electronică și citire a știrilor Evolution



Mai menționăm existența aplicațiilor web pentru poștă electronică, cum ar fi Squirrelmail, Yahoo!, gmail de la Google și Hotmail.

O trecere în revistă găsiți la [Linux Mail User HOWTO](#).

Cele mai multe distribuții de Linux includ **fetchmail**, un program de primire a mesajelor și de înaintare a acestora. El aduce mesaje de pe serverele de poștă electronică (POP, IMAP și altele) și le predă sistemului local de livrare. Puteți apoi gestiona aceste mesaje prin intermediul clientilor normali de poștă electronică. Poate fi rulat în sistemul daemon pentru a sonda în mod repetat unul sau mai multe sisteme la un interval specificat. Informații și exemple de utilizare găsiți în paginile Info; Directorul /usr/share/doc/fetchmail [-<version>] conține lista completă a facilităților și FAQ pentru începători.

Filtrul **procmail** poate fi folosit la filtrarea mesajelor primite, la crearea listelor de mail, la preprocesarea mesajelor, la înaintarea selectivă a mesajelor și la alte lucruri. Programul care îl acompaniază, **formail**, printre altele, permite generarea automată a răspunsurilor și divizarea căsuțelor de poștă electronică. Procmail este folosit de multă vreme pe sistemele UNIX și Linux, fiind un program robust, proiectat să lucreze și în cele mai rele circumstanțe. Găsiți mai multe informații în directorul `/usr/share/doc/procmail [-<version>]` și în paginile manualelor.



O notă privind eticheta (regulile de comportament) în cazul E-mail-ului

Anumiți oameni cred azi că mesajele de poștă electronică nu ar trebui să fie prea formale. Desigur, acest lucru depinde de mai mulți factori. Dacă scrieți cuiva pe care nu-l cunoașteți, este bine să păstrați distanță, la fel cum ați procedat în cazul unei scrisori tradiționale. Și nu uitați: oamenii pe care nu-i cunoașteți pot fi atât bărbați cât și femei...

10.3.3. Web

10.3.3.1. Serverul web Apache

Apache este, de departe, cel mai cunoscut server web, folosit pe mai mult de jumătate din serverele de pe Internet. Cele mai multe distribuții de Linux includ Apache. Printre avantajele Apache se numără alcătuirea modulară a acestuia, capabilitățile SSL, stabilitatea și viteza. Cu echipamentele corespunzătoare și configurarea corectă, poate suporta cele mai grele condiții de trafic.

Pe sistemele Linux, configurațiile serverului sunt făcute în directorul /etc/httpd. Cel mai important fișier de configurație este httpd.conf; este destul de explicit. Dacă aveți nevoie de ajutor, puteți căuta în magazinile man ale httpd sau vizitați saitul [Apache](#).

10.3.3.2. Navigatoare web

Există numeroase navigatoare web, atât libere și gratuite, cât și comerciale, pentru platformele Linux. Netscape Navigator, ca singura opțiune de luat în seamă, este de domeniul trecutului, de când Mozilla/Firefox oferă o alternativă competitivă pentru mai multe sisteme de operare, cum ar fi MS Windows și MacOS X.

Amaya este un navigator W3C. Opera este un navigator comercial, compact și rapid. Multii administratori de ecran pun la dispoziție facilități de navigare pe web, ca **nautilus**.

Printre navigatoarele cunoscute din mediul text enumerăm **lynx** și **links**. Trebuie să vă definiți configurația serverelor proxy în consolă prin configurația variabilelor adesea. Navigatoarele text sunt rapide și la îndemână atunci când nu este disponibil un mediu grafic, ca atunci când se folosesc în scripturi.

10.3.3.3. Servere proxy

10.3.3.3.1. Ce este un server proxy?

Companiile și organizațiile doresc ca utilizatorii lor să folosească un server proxy. În special în mediile cu mulți utilizatori, un server proxy poate îmbunătăți încărcarea paginilor web. Serverele proxy stochează paginile web. În momentul în care un utilizator solicită o pagină web care a mai fost vizitată, serverul va oferi aceea pagină imediat utilizatorului, fără a mai fi nevoie de obținerea ei de pe Internet, ceea ce ia mai mult timp. Desigur că se iau măsuri ca serverul proxy să verifice și să furnizeze cea mai recentă versiune a paginii solicitate. În anumite medii, folosirea serverelor proxy este obligatorie, în altele poate fi la alegerea dumneavoastră.

10.3.3.3.2. Configurația proxy

Dacă știți numele și portul serverului proxy, normal că trebuie să furnizați această informație navigatorului dumneavoastră. Totuși, multe aplicații (în linie de comandă) depind de variabilele http_proxy și ftp_proxy pentru ca ele să funcționeze corect. Pentru un comportament normal, trebuie să adăugați o linie asemănătoare cu următoarea în fișierul ~/.bashrc:

```
export http_proxy=http://nume_utilizator:parolă@nume_proxy_server:număr_port
```

De exemplu:

```
export http_proxy=http://willy:Appelsi3ntj3@proxy:80
```

Dacă nu este necesar să furnizați un nume de utilizator și parolă, nu introduceți nimic înainte de semnul „@”, semnul fiind necesar.

10.3.4. File Transfer Protocol (Protocolul pentru transferul de fișiere)

10.3.4.1. Servere FTP

Pe un sistem Linux, serviciul FTP este rulat din **xinetd**, folosind serviciul WU-ftp, deși serviciul

FTP poate fi configurat ca un serviciu independent pe sistemele cu trafic FTP intens. Consultați exercițiile.

Serviciul FTP include, pe lângă altele, vsftpd, Ncftpd și Proftpd.

Cele mai multe distribuții de Linux conțin pachetul anonftp, care configurează un server FTP anonim și fișierele de configurare necesare.

10.3.4.2. Clienți FTP

Cele mai multe distribuții de Linux includ **ncftp**, o versiune îmbunătățită a comenzi UNIX **ftp**, pe care este posibil să o cunoașteți din linia de comandă a Windows-ului. Programul **ncftp** oferă facilități suplimentare cum ar fi interfețe mai prietenoase și mai stilizate, autocompletarea numelor fișierelor, adăugarea și reluarea anumitor funcții, semne de carte, administrarea sesiunilor și altele:

```
thomas:~> ncftp blob
NcFTP 3.0.3 (April 15, 2001) by Mike Gleason (ncftp@ncftp.com) .
Connecting to blob...
blob.some.net FTP server (Version wu-2.6.1-20) ready.

Logging in...
Guest login ok, access restrictions apply.
Logged in to blob.
ncftp / > help
Commands may be abbreviated. 'help showall' shows hidden and
unsupported commands.
'help <command>' gives a brief description of <command>.

ascii      cat      help      lpage      open      quote      site
bgget      cd       jobs      lpwd       page      rename     type
bput       chmod    lcd       lrename   pdir      rhelp      umask
bgstart    close    lchmod   lrm        pls       rm        version
binary     debug    lls       lrmdir    put       rmdir
bookmark   dir      lmkdir   ls        pwd       set
bookmarks  get      lookup   mkdir    quit      show
ncftp / >
```

Paginile man sunt excelente, ele conținând și multe exemple. Din nou, vă spunem că sunt disponibile și aplicații grafice.



FTP nu este securizat!

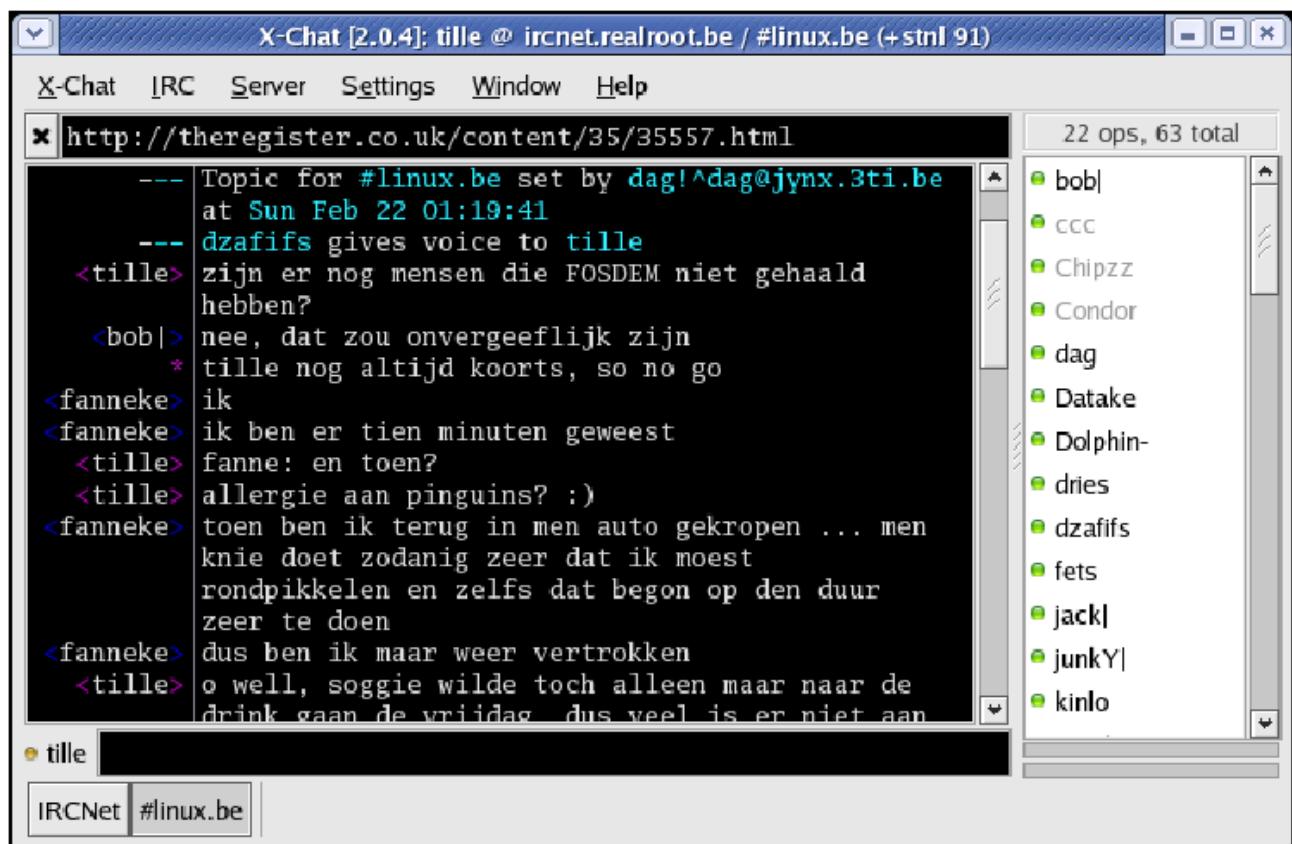
Nu folosiți Protocolul pentru transfer de fișiere (FTP) pentru conectări non-anonime, decât dacă știți ce faceți. Numele dumneavoastră de utilizator și parola pot fi interceptate de alți utilizatori ai rețelei! Folosiți în schimb FTP securizat; programul **sftp** este inclus în suita Secure SHell, consultați [Secțiunea 10.4.4.4](#).

10.3.5. Chat și conferință

Sunt disponibile mai multe sisteme și programe pentru fiecare distribuție, care înlocuiesc chat-ul bazat pe stilul vechi al IRC. Iată o listă scurtă și incompletă a celor mai folosite programe:

- **gaim**: mesagerie instant multiprotocol pentru Linux, Windows și Mac, compatibil cu MSN Messenger, ICQ, IRC și multe altele; Consultați paginile Info sau saitul Gaim pentru mai multe informații.
- **xchat**: client IRC pentru sistemul de ferestre X (vezi Figura 19-2 de pe pagina care urmează). Pagina programului se află pe [SourceForge](#).
- **aMSN**: o clonă a MSN.
- **Konversation, kopete, KVirc** și multe altele pentru suita KDE.
- **gnomemeeting**: program de videoconferință pentru UNIX (acum cunoscut drept Ekiga).
- **jabber**: platformă de mesagerie instant Open Source, compatibilă cu ICQ, AIM, Yahoo, MSN, IRC, SMTP și altele.

Figura 19-2. X-Chat



- **psi**: client jabber, consultați [the PSI Jabber Client Homepage](#).
- **skype**: program pentru con vorbiri asemănătoare celor telefonice, folosind Internetul, cu alți utilizatori Skype. Consultați <http://www.skype.com> pentru mai multe informații. Skype este gratuit, dar nu liber (deschis).
- **Gizmo**: un program gratuit (dar nu liber) de telefonie pentru calculatorul dumneavoastră, consultați <http://www.gizmoproject.com>.

10.3.6. Servicii de știri

Punerea pe picioare a unui serviciu Usenet implică multă experiență și atenție la detaliile configurării, așa că vă îndreptăm către pagina [INN](#) pentru mai multe informații.

Există câteva grupuri de știri interesante în ierarhia *comp**, care pot fi accesate folosind mai multe programe pentru mediul text sau grafic. Mulți clienți de mail se pot descurca și cu administrarea grupurilor de știri, verificați aşadar programul propriu sau căutați în saitul oglindă local programe Open Source ca **tin**, **slrn** și **mutt** sau descărcați Mozilla ori alt client grafic.

[Deja.com](#) păstrează o arhivă o tuturor grupurilor de știri, susținută de Google. Este un instrument excelent atunci când aveți nevoie de ajutor: sunt foarte multe șanse ca cineva să se fi confruntat cu aceeași problemă, să fi găsit și soluția, iar răspunsul să fie postat într-unul din grupurile de știri.

10.3.7. The Domain Name System (Sistemul numelui de domenii)

Toate aceste aplicații au nevoie de serviciile DNS pentru a potrivi adresele IP cu numele gazdelor și invers. Un server DNS nu cunoaște toate adresele de IP din lume, dar lucrează în rețea cu alte servere DNS pe care le poate chestiona în legătură cu adrese necunoscute. Cele mai multe sisteme UNIX pot rula **named**, care este o parte a pachetului BIND (Berkeley Internet Name Domain) distribuit de Internet Software Consortium. Poate lucra ca un serviciu independent de server de nume, pentru a îmbunătăți accesul la rețea pe sistemele Linux.

Fișierul principal de configurare este în /etc/resolv.conf, acesta determinând ordinea în care sunt contactate DNS-urile:

```
search somewhere.org
nameserver 192.168.42.1
nameserver 193.74.208.137
```

Mai multe informații pot fi găsite în paginile Info ale **named**, în fișierul /usr/share/doc/bind[<version>] și la pagina [Proiectului Bind. DNS HOWTO](#) acoperă utilizarea BIND ca server DNS.

10.3.8. DHCP

DHCP este the Dynamic Host Configuration Protocol (Protocolul de configurare dinamică a gazdei), care înlocuiește gradual bătrânelul **bootp** în mediile mari. Este folosit pentru controlarea parametrilor vitali de rețea, precum adrese de IP și serverele de nume ale gazdei. DHCP este compatibil retroactiv cu **bootp**. Pentru configurația serviciului, trebuie să citiți HOWTO.

Calculatoarele client care folosesc DHCP sunt configurate de regulă prin interfața grafică care configurațiază **dhpcd**, serviciul client DHCP. Verificați documentația sistemului dacă trebuie să configurați calculatorul drept client DHCP.

10.3.9. Servicii de autentificare

10.3.9.1. Tradiționale

Tradițional, utilizatorii sunt autentificați local, folosind informațiile stocate în /etc/passwd și /etc/shadow, fișiere existente pe orice sistem. Chiar folosind un serviciu de rețea pentru autentificare, fișierele locale vor fi întotdeauna prezente pentru a configura conturile sistemului pentru utilizarea administrativă, precum conturile de root, serviciile care rulează pe anumite conturi și conturile create pentru anumite programe sau scopuri.

Aceste fișiere sunt primele care sunt examineate de cineva care dorește să preia controlul asupra calculatorului dumneavoastră, așa că asigurați-vă că politica permisiunilor și a deținătorilor este strictă:

```
bob:~> ls -l /etc/passwd /etc/shadow
-rw-r--r--    1 root      root          1803 Mar 10 13:08 /etc/passwd
-r-----    1 root      root          1116 Mar 10 13:08 /etc/shadow
```

10.3.9.2. PAM

Linux-ul poate folosi PAM, The Pluggable Authentication Module, o metodă flexibilă de autentificare pe sistemele UNIX.

Avantajele PAM:

- Poate fi folosită o schemă comună de autentificare pentru o gamă largă de aplicații.
- PAM poate fi implementată pe multe aplicații fără a fi necesară recompilarea aplicațiilor pentru a rula cu PAM.
- Administratorii și dezvoltatorii de programe se bucură de avantaje ca flexibilitatea și controlul autentificărilor.
- Dezvoltatorii nu trebuie să proiecteze un program pentru o anumită schemă de autentificare. Ei se pot concentra pe detaliile programului în sine.

Directorul /etc/pam.d conține fișierele de configurație ale PAM (înainte era /etc/pam.conf). Fiecare aplicație sau serviciu are propriul fișier. Oricare linie din fișier are patru elemente:

- *Module (Module):*
 - ◆ auth: asigură autentificarea propriu-zisă (poate prin solicitarea și verificarea unei parole) și stabilește permisiunile, precum apartenența la un grup sau bilete Kerberos.

- ◆ account: verifică pentru a se asigura că accesul este permis pentru utilizator (contul nu a expirat, utilizatorului îi este permis să se autentifice la acel moment al zilei, etc.).
- ◆ passwords: folosit pentru configurarea parolelor.
- ◆ session: folosit după ce un utilizator se autentifică. Acest modul execută sarcini adiționale de care este nevoie pentru permiterea accesului (de exemplu, montarea directorului personal al utilizatorului sau punerea la dispoziție a căsuței poștale).

Ordinea în care sunt alăturate modulele, în aşa fel încât să fie folosite mai multe, este foarte importantă.

- *Control Flags (Etichete de control)*: spune PAM ce acțiuni să inițieze în caz de eșec sau succes. Valorile pot fi required, requisite, sufficient sau optional.
- *Module Path (Calea către modul)*: traseul către modulul care va fi folosit, de regulă în /lib/security.
- *Arguments (Argumente)*: informații pentru module.

Fișierele /etc/shadow și /etc/passwd sunt detectate automat de PAM.

Mai multe informații sunt disponibile în paginile man ale **pam** și pe pagina [Proiectul Linux-PAM](#).

10.3.9.3. LDAP

Lightweight Directory Acces Protocol este un sistem client-server pentru accesarea unor directoare de servicii globale sau locale prin intermediul rețelei. Pe sistemele Linux, este folosită implementarea OpenLDAP. Include **slapd**, un serviciu independent; **slurpd**, un server replicator independent; librăriile care implementează protocolul LDAP și o serie de utilități, unelte și clienți.

Principalul beneficiu al utilizării LDAP este securizarea anumitor tipuri de informații în interiorul organizației dumneavoastră. De exemplu, toate listele de utilizatori din organizația dumneavoastră pot fi integrate într-un director LAPD. Acest director poate fi apelat de toate aplicațiile pe care LAPD le permite și care au nevoie de aceste informații. Pot fi accesate și de utilizatorii care au nevoie de informațiile din director.

Alte beneficii ale LAPD sau X.500 Lite includ ușurința în implementare (comparată cu X.500) și Application Programming Interface (API), ceea ce se traduce prin faptul că numărul aplicațiilor LAPD și porțile LAPD vor crește ca număr în viitor.

Ca minusuri, dacă dorîți să folosiți LAPD, veți avea nevoie de aplicații care să funcționeze cu acest serviciu sau de abilitatea de a folosi porțile LAPD. În timp ce folosirea LAPD se extinde, nu putem spune că disponem de multe aplicații pentru Linux capabile să lucreze cu acest serviciu. De asemenea, cu toate că LDAP dispune de un anume control al accesului, nu posedă atât de multe facilități legate de securitate precum X.500.

Întrucât LDAP este un protocol deschis și configurabil, poate fi folosit la stocarea oricărui tip de informație legată de structura particulară a unei organizații. Exemple comune sunt bazele de date cu adrese de poștă electronică, autentificarea în combinație cu PAM, bazele de date care conțin numere de telefon și bazele de date care conțin fișierele de configurație ale sistemelor.

Consultați informațiile specifice sistemului dumneavoastră și paginile man ale comenzi care au legătură cu acest serviciu, cum ar fi **Idapmodify** și **Idapsearch** pentru mai multe detalii. Mai multe informații puteți găsi în [LDAP Linux HOWTO](#), care prezintă instalarea, configurarea, rularea și întreținerea unui server LDAP pe sistemele Linux. Autoarea acestei introduceri în Linux a scris și [LDAP Operation HOWTO](#), în care sunt descrise lucrurile de bază pe care trebuie să le cunoască oricine care are de-a face cu administrarea, operarea și integrarea serviciilor LDAP.

10.4. Executarea la distanță a aplicațiilor

10.4.1. Introducere

Există mai multe moduri în care pot fi executate comenzi sau lansate programe pe calculatoare aflate la distanță și obținerea ieșirilor acestora, în mod text sau grafic, pe stația dumneavoastră de lucru. Conectarea poate fi securizată sau nu. Cu toate că vă sfătuim să folosiți conexiuni securizate în loc să vă

trimiteți parola necriptată pe rețea, vom discuta și unele aplicații practice ale mecanismului mai vechi (nesecurizat), întrucât acestea sunt încă folosite în mediile rețelelor moderne, cum ar fi rularea sau repararea unor aplicații exotice.

10.4.2. Rsh, rlogin și telnet

Comenzile **rlogin** și **rsh** pentru autentificarea și executarea la distanță a comenziilor sunt moștenite din UNIX. Ele sunt folosite rareori pentru că sunt extrem de nesigure dar sunt integrate în orice sistem Linux pentru asigurarea compatibilității retroactive cu programele UNIX.

Telnet, pe de altă parte, este încă folosit, adesea de către administratorii de sisteme sau rețele. Telnet este una din cele mai puternice unelte pentru accesarea la distanță a fișierelor și pentru administrarea la distanță, permitând conexiuni de oriunde există Internet. Combinat cu un server X, aplicații grafice de pe un alt calculator pot fi afișate local. Nu este nici o diferență între a lucra local pe un calculator și folosirea unui calculator aflat la distanță.

Deoarece conexiunea nu este criptată, permiterea unei legături **telnet** implică riscuri majore de securitate. Pentru executarea normală a programelor la distanță, vă sfătuim să folosiți Secure SHell sau **ssh**. Vom prezenta metoda sigură pe parcursul acestei secțiuni.

Totuși, **telnet** este folosit în multe cazuri. Mai jos sunt câteva exemple în care un server de mail și unul de web sunt testate pentru răspunsuri:

Verificarea stării unui server de mail:

```
[jimmy@blob ~] telnet mailserver 25
Trying 192.168.42.1...
Connected to mailserver.
Escape character is '^>'.
220 ml.some.net ESMTP Sendmail 8.11.6/8.11.6; 200302281626
ehlo some.net
250-ml.some.net Hello blob.some.net [10.0.0.1], pleased to meet you
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-SIZE
250-DSN

250-ONEX
250-ETRN
250-XUSR
250 HELP
mail from: jimmy@some.net
250 2.1.0 jimmy@some.net... Sender ok
rcpt to: davy@some.net
250 2.1.5 davy@some.net... Recipient ok
data
354 Enter mail, end with "." on a line by itself
test
.
250 2.0.0 g2MA1R619237 Message accepted for delivery
quit
221 2.0.0 ml.some.net closing connection
Connection closed by foreign host.
```

Verificarea unui server web din punct de vedere al răspunsului la cereri de bază:

```
[jimmy@blob ~] telnet www.some.net 80
Trying 64.39.151.23...
Connected to www.some.net.
Escape character is '^>'.
HEAD / ;HTTP/1.1

HTTP/1.1 200 OK
Date: Fri, 22 Mar 2002 10:05:14 GMT
Server: Apache/1.3.22 (UNIX) (Red-Hat/Linux)
```

```
mod_ssl/2.8.5 OpenSSL/0.9.6
DAV/1.0.2 PHP/4.0.6 mod_perl/1.24_01
Last-Modified: Fri, 04 Jan 2002 08:21:00 GMT
ETag: "70061-68-3c3565ec"
Accept-Ranges: bytes
Content-Length: 104
Connection: close
Content-Type: text/html

Connection closed by foreign host.

[jimmy@blob ~]
```

Această procedură este sigură, deoarece nu a trebuit să furnizați un nume de utilizator și/sau o parolă pentru a obține datele pe care le doriți, aşa că nimeni nu poate captura aceste informații importante din rețea.

10.4.3. Sistemul X Window

10.4.3.1. Caracteristicile X

După cum am explicat deja în Capitolul 7 (consultați [Sectiunea 7.3.3](#)), sistemul X Window include un server X care se ocupă de partea grafică pentru clienții care solicită afișajul.

Este important să faceți distincția dintre serverul X și aplicațiile client ale acestuia. Serverul X controlează direct afișajul și este responsabil pentru toate intrările și ieșirile via tastatură, mouse și ecran. Clienții X, pe de altă parte, nu accesează dispozitivele de intrare/ieșire în mod direct. Ei comunică cu serverul X care gestionează intrările și ieșirile. Clienții X fac munca propriu-zisă, precum calcularea valorilor, rularea aplicațiilor și așa mai departe. Serverul X doar deschide ferestrele prin care se transmit intrările și ieșirile pentru clientul specificat.

În operarea normală (mediul grafic), orice stație de lucru Linux este un server X pentru sine, chiar dacă rulează doar aplicații client. Toate aplicațiile pe care le rulați (de exemplu, Gimp, o fereastră terminal, navigatorul dumneavoastră, aplicațiile de birou, programul pentru redarea CD-urilor, etc.) sunt clienți ai serverului X. Serverul și clienții, în acest caz, rulează pe aceeași mașină.

Natura client/server a sistemului X îl transformă într-un mediu ideal pentru execuția la distanță a aplicațiilor și programelor. Deoarece procesele sunt executate de fapt pe calculatorul aflat la distanță, pe gazda locală sunt consumate puține resurse ale procesorului. Astfel de calculatoare, care lucrează doar ca server X, sunt numite terminale X și au fost foarte populare în trecut. Mai multe informații puteți găsi în [Remote application mini-HOWTO](#).

10.4.3.2. Telnet și X

Dacă doriți să folosiți **telnet** pentru a afișa aplicațiile grafice care rulează pe un calculator aflat la distanță, primul lucru pe care trebuie să-l faceți este să permiteți calculatorului aflat la distanță accesul la afișajul dumneavoastră (la serverul X propriu!) prin folosirea comenzii **xhost**, introducând o comandă similară celei de mai jos într-o fereastră terminală pe calculatorul dumneavoastră:

```
davy:~> xhost +remote.machine.com
```

După aceasta, conectați-vă la calculatorul aflat la distanță și spuneți-i să afișeze grafica pe calculatorul dumneavoastră prin configurarea variabilei DISPLAY:

```
[davy@remote ~] export DISPLAY="local.host.com:0.0"
```

După parcurserea acestei etape, orice aplicație pornită în această fereastră terminal va fi afișată pe calculatorul dumneavoastră, folosind resursele de calcul aflate la distanță, dar resursele grafice proprii (serverul X propriu) pentru afișarea aplicațiilor.

Această procedură pleacă de la ipoteza că dispuneți de o variantă a serverului X (XFree86, X.org, Exceed, Cygwin) instalat corespunzător pe calculatorul pe care doriți să afișați aplicațiile grafice. Arhitectura și sistemul de operare ale calculatorului client nu sunt importante, atât timp cât vă permit să rulați un server

X pe acestea.

Rețineți că afișarea unei ferestre terminal de pe un calculator aflat la distanță este tratată ca afișarea unei imagini.

10.4.4. Suite SSH

10.4.4.1. Introducere

Cele mai multe sisteme UNIX și Linux utilizează Secure SHell pentru a preveni riscurile de securitate generate de **telnet**. Pe sistemele Linux veți găsi probabil o versiune a OpenSSH, implementarea Open Source a protocolului SSH, care asigură comunicări securizate criptate între calculatoare care folosesc o rețea nesigură. În configurarea standard conexiunile X sunt înaintate automat, dar porturi arbitrare TCP/IP pot fi de asemenea înaintate, folosind un canal securizat.

Clientul **ssh** se conectează la gazda indicată. Utilizatorul trebuie să-și prezinte identitatea către calculatorul aflat la distanță și această identitate trebuie să corespundă cu cea din fișierul de configurare **sshd_config**, care poate fi găsit, în cele mai multe cazuri, în **/etc/ssh**. Fișierul de configurare este destul de explicit și permite în configurarea implicită cele mai folosite facilități. Dacă aveți nevoie de ajutor, îl puteți găsi în paginile man **sshd**.

După ce identitatea utilizatorului a fost verificată și acceptată de server, acesta execută comanda primită sau se conectează la calculator și oferă utilizatorului o consolă normală pe calculatorul aflat la distanță. Toate comunicările, comenzi pentru aplicațiile la distanță sau pentru consolă, vor fi criptate automat. Sesiunea este închisă atunci când comenziile sau consola de pe calculatorul aflat la distanță sunt terminate și toate conexiunile X11 și TCP/IP sunt închise.

Când vă conectați pentru prima dată la o gazdă, folosind oricare din programele care sunt incluse în colecția SSH, trebuie să stabiliți autenticitatea acelei gazde la care doriți să vă conectați:

```
lenny ~> ssh blob
The authenticity of host 'blob (10.0.0.1)' can't be established.
RSA fingerprint is 18:30:50:46:ac:98:3c:93:1a:56:35:09:8d:97:e3:1d.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'blob,192.168.30.2' (RSA) to the list of
known hosts.
Last login: Sat Dec 28 13:29:19 2002 from octarine
This space for rent.

lenny is in ~
```

Este important să tasteți „yes”, toate cele trei caractere, nu doar „y”. Prin aceasta se modifică fișierul **~/.ssh/known_hosts**, consultați [Secțiunea 10.4.4.3](#).

Dacă doriți doar să verificați ceva pe calculatorul aflat la distanță și apoi să vă reîntoarceți la calculatorul dumneavoastră, puteți introduce comanda pe care doriți să o executați la distanță ca argument pentru **ssh**:

```
lenny ~> ssh blob who
jenny@blob's password:
root      tty2          Jul 24 07:19
lena      tty3          Jul 23 22:24
lena      0:             Jul 25 22:03

lenny ~> uname -n
magrat.example.com
```

10.4.4.2. Înaintarea conexiunilor X11 și TCP

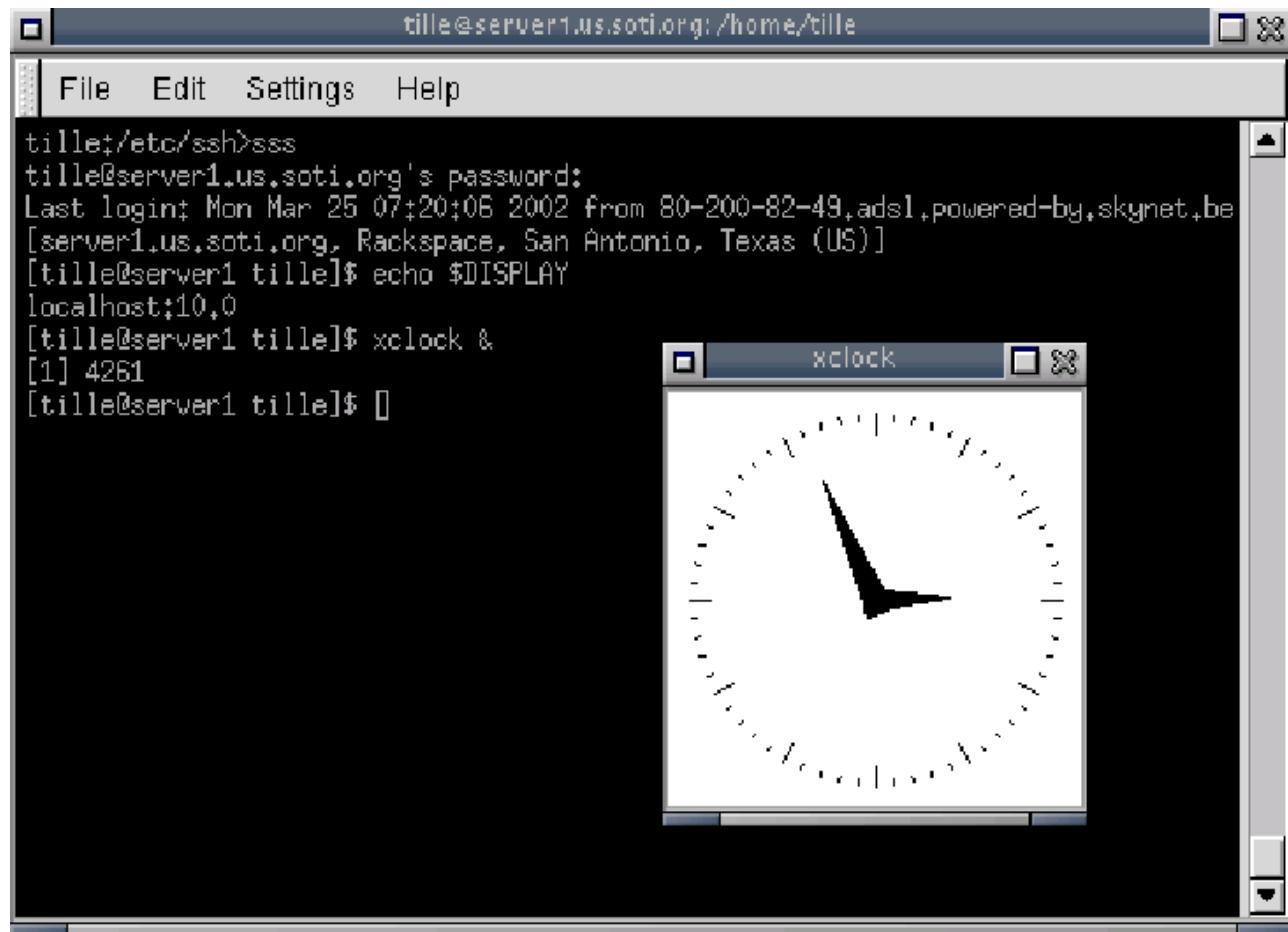
Dacă este configurată pe **yes** intrarea **X11Forwarding** pe calculatorul țintă și utilizatorul folosește aplicații X, variabila de mediu **DISPLAY** fiind configurață, conexiunea cu serverul X este automat înaintată către partea aflată la distanță astfel încât orice program X11 pornit din consolă va parcurge canalul criptat, iar conexiunea cu serverul X real va fi făcută de la calculatorul local. Utilizatorul nu trebuie să

configureze manual variabila DISPLAY. Înaintarea conexiunilor X11 poate fi configurată din linia de comandă sau prin fișierul de configurare **sshd**.

Valoarea pentru DISPLAY configurață prin **ssh** va ținti către calculatorul server, dar cu un număr de afișaje mai mare de 0. Acest lucru este normal și apare pentru că **ssh** crează un server X proxy pe calculatorul server (care rulează aplicațiile client X) pentru înaintarea conexiunilor pe canalul criptat.

Total se face automat, aşa încât atunci când tastăți numele unei aplicații grafice, este afișată pe calculatorul dumneavoastră și nu pe calculatorul aflat la distanță. Noi utilizăm **xclock** în exemplul care urmează, întrucât este un program mic, în general instalat pe toate sistemele Linux, ideal pentru astfel de teste.

Figura 10-3. SSH X11 forwarding



SSH va configura automat și datele Xauthority de pe calculatorul server. În acest scop, va genera la întâmplare un cookie de autorizare, păstrat în fișeul Xauthority de pe server și va verifica dacă fiecare conexiune înaintată poartă acest cookie pe care-l va înlocui cu cel real în momentul în care conexiunea este deschisă. Cookie-ul real de autentificare nu este trimis niciodată calculatorului server (după cum nici un fișier cookie nu este trimis în clar).



Serverul X

Procedura de mai sus pornește de la presupunerea că aveți configurat un server X pe calculatorul pe care dorîți să afișați o aplicație de pe calculatorul aflat la distanță. Clientul poate avea o arhitectură diferită și un alt sistem de operare atât timp cât poate rula un server X, cum ar fi Cygwin (care implementează un server X.org pentru clientii MS Windows și nu numai) sau Exceed, fiind astfel posibilă realizarea unei conexiuni la distanță cu orice calculator care rulează UNIX sau Linux.

10.4.4.3. Autentificarea la server

Sistemul client/server **ssh** menține și verifică o bază de date care conține identificarea tuturor gazdelor care au fost folosite. Cheile gazdelor sunt păstrate în `$HOME/.ssh/known_hosts` aflat în

directorul personal al utilizatorului. În plus, fișierul /etc/ssh/ssh_known_hosts este verificat automat în vederea comparării cu gazdele cunoscute. Orice gazdă nouă este adăugată automat în fișierul utilizatorului. Dacă identitatea unei gazde se schimbă la un moment dat, **ssh** emite un avertisment și dezactivează autentificarea prin parolă pentru a preveni ca vreun Troian să se folosească de acea parolă. Alt scop al acestui mecanism este prevenirea atacurilor man-in-the middle, care pot fi folosite la scurtcircuitarea criptării. În mediile care necesită standarde ridicate de securitate, **sshd** poate fi configurat să nu permită conectarea la calculatoarele ale căror chei au fost schimilate sau nu sunt cunoscute.

10.4.4.4. Copierea securizată la distanță.

Colecția SSH include comanda **scp** ca alternativă sigură a comenzi folosite extensiv în trecut, **rcp**, când exista doar **rsh**. **scp** folosește **ssh** pentru transferul de date, utilizează aceeași autentificare și asigură aceeași securitate ca **ssh**. Spre deosebire de **rcp**, **scp** va cere parolele sau propozițiile de control dacă ele sunt cerute pentru autentificare:

```
lenny /var/tmp> scp Schedule.sdc.gz blob:/var/tmp/
lenny@blob's password:
Schedule.sdc.gz 100% |*****| 100 KB 00:00

lenny /var/tmp>
```

Orice nume de fișier poate conține specificații despre gazdă și utilizatori pentru a indica faptul că fișierul este destinat copierii de pe/către acea gazdă. Este permisă copierea în cazul a două gazde aflate la distanță. Consultați paginile Info pentru mai multe informații.

Dacă folosiți mai des o interfață asemănătoare FTP, utilizați **sftp**:

```
lenny /var/tmp> sftp blob
Connecting to blob...
lenny@blob's password:

sftp> cd /var/tmp

sftp> get Sch*
Fetching /var/tmp/Schedule.sdc.gz to Schedule.sdc.gz

sftp> bye

lenny /var/tmp>
```



Copii securizate sau interfețe grafice FTP

Nu vă simțiți încă în largul dumneavoastră în mediul liniei de comandă? Încercați capabilitățile Konqueror legate de copierea securizată la distanță sau instalați Putty.

10.4.4.5. Cheile de autentificare

Comanda **ssh-keygen** generează, administrează și convertește cheile de autentificare pentru **ssh**. Poate genera chei RSA pentru folosirea cu protocolul SSH versiunea 1 și chei RSA sau DSA pentru protocolul SSH versiunea 2.

În mod normal, orice utilizator care dorește să folosească SSH cu autentificare RSA sau DSA rulează această comandă măcar odată pentru a genera chei de autentificare în \$HOME/.ssh/identity, **id_dsa** sau **id_rsa**. În plus, administratorul de sistem poate utiliza această comandă pentru generarea cheilor de gazdă pentru sistem.

Acest program generează cheile și solicită un fișier în care să stocheze cheile private. Cheia publică este păstrată într-un fișier cu același nume dar cu terminația **.pub**. Programul solicită și o propoziție de control. Câmpul pentru propoziția de control poate fi lăsat gol, ceea ce înseamnă că ea nu există (cheile gazdei trebuie să nu aibă propoziție de control) sau poate conține un sir de o lungime arbitrară.

Nu există nici o modalitate de a recupera o propoziție de control (parolă) pierdută. Dacă propoziția de control este pierdută sau uitată, trebuie să generați o cheie nouă și să o copiați în cheia publică

corespondentă.

Vom studia cheile SSH prin exerciții. Toate informațiile de care aveți nevoie pot fi găsite în paginile man sau Info.

10.4.5. VNC

VNC sau Virtual Network Computing este de fapt un sistem de afișare la distanță care permite vizualizarea unui spațiu de lucru (desktop environment) nu doar pe calculatorul local, ci oriunde pe Internet, pe o mare varietate de sisteme și arhitecturi, inclusiv MS Windows și câteva distribuții UNIX. Puteți, de exemplu, să rulați MS Word pe un sistem Windows NT și să afișați ieșirea pe un spațiu de lucru Linux. VNC asigură atât serverele cât și clientii, așa că funcționează și invers, adică permite afișarea programelor Linux pe clienti Windows. VNC este, probabil, cea mai usoară cale prin care puteți avea conexiuni X pe calculatorul dumneavoastră. Următoarele caracteristici fac ca VNC să fie diferit de un server X normal sau de implementările comerciale:

- Nu este stocat nimic pe calculatorul pe care se afișează: puteți părasi stația de lucru curentă și să puteți relua lucrul de la alt calculator, din punctul în care ați rămas. Când rulați un server X pe un calculator și acesta îngheată sau este repornit, toate aplicațiile la distanță pe care le rulați vor fi terminate. Cu VNC, acestea vor continua să ruleze.
- Este mic și simplu, nu necesită instalare, poate rula de pe o dischetă dacă este nevoie.
- Cu clientul Java este independent de platforma folosită, rulând pe orice acceptă X.
- Este divizibil: un spațiu de lucru poate fi afișat pe mai multe calculatoare.
- Este liber și gratuit.

Mai multe informații puteți găsi în paginile man ale VNC (**man vncviewer**) sau pe [saitul VNC](#).

10.4.6. Protocolul rdesktop

Pentru a permite administrarea calculatoarelor care rulează MS Windows, distribuțiile recente de Linux integrează Remote Desktop Protocol (RDP), care este implementat în clientul **rdesktop**. Protocolul este folosit de anumite produse Microsoft, inclusiv Windows NT Terminal Server, Windows 2000 Server, Windows XP și Windows 2003 Server.

Surprindeți-vă prietenii (sau șefii) cu modul „pe tot ecranul”, diversele tipuri ale aspectului tastaturii și modul single application, identice cu cele din implementarea originală. Comanda **man rdesktop** vă oferă mai multe informații. Pagina de Internet a proiectului o găsiți la <http://www.rdesktop.org/>.

10.4.7. Cygwin

[Cygwin](#) asigură multe din funcționalitățile UNIX pe sistemele MS Windows. În afară de a permite funcționarea uneltelelor liniei de comandă și a aplicațiilor grafice specifice UNIX-ului, poate fi folosit la afișarea unui spațiu de lucru Linux pe sistemele MS Windows, prin utilizarea la distanță a serverului X. Într-o consolă Cygwin Bash tastăți:

```
/usr/X11R6/bin/XWin.exe -query numele_calculatorului_dumneavoastră_sau_IP
```

Conexiunea este respinsă în configurarea implicită. Trebuie să modificați configurația X Display Manager (XDM) și, posibil, configurația X Font Server (XFS) pentru permiterea acestui tip de conexiune, ancare primiți un ecran de autentificare pe calculatorul aflat la distanță. În funcție de administratorul spațiului dumneavoastră de lucru (Gnome, KDE sau altul), pot fi necesare modificări și în alte fișiere de configurație.

Dacă nu aveți nevoie de afișarea spațiului de lucru în întregime, puteți utiliza SSH în Cygwin, exact în modul prezentat în [Secțiunea 10.4.4](#) fără să vă deranjați a mai edita fișiere de configurație.

10.5. Securitate

10.5.1. Introducere

Imediat ce un calculator este conectat la o rețea, devin posibile tot felul de agresiuni, fie că vorbim despre un sistem bazat pe UNIX sau oricare altul. Cunoaștem că pe această temă au fost scrise extraordinar de multe lucrări și nu ne facem un scop în sine din discutarea în detaliu a subiectului securității. Sunt însă câteva chestii logice, de bază, prin care un utilizator, novice chiar, poate obține un sistem mai sigur, deoarece majoritatea intruziunilor sunt posibile din cauza utilizatorilor neglijenți sau ignoranți.

Vă întrebați poate dacă aceste considerații se aplică și în cazul dumneavoastră, care folosiți un calculator acasă sau la serviciu, într-un mediu cu un grad mai mare de securizare. Întrebările la care trebuie să răspundeți sunt mai degrabă următoarele:

- Vreți să aveți controlul asupra sistemului dumneavoastră?
- Vreți să participați (fără voie) în activități incriminate penal?
- Vreți ca echipamentele dumneavoastră să fie folosite de altcineva?
- Vă asumați riscul pierderii conexiunii la Internet?
- Vreți să vă reinstalați sistemul de fiecare dată când sunteți victimă unui atac?
- Sunteți dispus să pierdeți date personale sau alt fel de date importante pentru dumneavoastră?

Noi presupunem că nu doriți ca aceste lucruri să devină realitate, așa că vă prezentăm o listă cu măsurile pe care le puteți lua pentru întărirea securității sistemului dumneavoastră. Informații pe larg puteți găsi în [Linux Security HOWTO](#).

10.5.2. Serviciile

Scopul nostru este rularea a cât mai puține servicii. Dacă numărul porturilor care sunt deschise pentru lumea exterioară este menținut la minim, puteți supraveghea mai ușor mediul sistemului. În cazurile în care serviciile nu pot fi opriate pentru rețeaua locală, încercați ca ele să nu fie disponibile pentru conexiunile exterioare.

O metodă empirică spune că dacă nu recunoașteți un serviciu anume, sunt multe șanse ca acest serviciu să nu fie necesar. De asemenea, anumite servicii nu trebuie neapărat să fie folosite pe Internet. Nu vă bazați pe ce ar trebui să ruleze, ci verificați care sunt serviciile care ascultă și pe ce porturi TCP, folosind comanda **netstat**:

```
[elly@mars ~] netstat -l | grep tcp
tcp        0      0 *:32769          *:*      LISTEN
tcp        0      0 *:32771          *:*      LISTEN
tcp        0      0 *:printer        *:*      LISTEN
tcp        0      0 *:kerberos_master *:*      LISTEN
tcp        0      0 *:sunrpc         *:*      LISTEN
tcp        0      0 *:6001           *:*      LISTEN
tcp        0      0 *:785            *:*      LISTEN
tcp        0      0 localhost.localdomain:smtp *:*      LISTEN
tcp        0      0 *:ftp             *:*      LISTEN
tcp        0      0 *:ssh             *:*      LISTEN
tcp        0      0 :1:x11-ssh-offset *:*      LISTEN
```

Lucruri pe care trebuie să le evitați:

- **exec**, **rlogin** și **rsh** și **telnet** pentru a nu pune la îndemâna atacatorilor diverse oportunități.
- X11 pe calculatoarele server.
- Nu folosiți lp dacă nu există nici o imprimantă legată la sistem.
- Dacă nu există în rețea calculatoare care rulează MS Windows, nu aveți nevoie de Samba.
- Nu permiteți FTP decât dacă este solicitat un serviciu FTP.
- Nu permiteți NFS și NIS pe Internet, dezactivați toate serviciile care au legătură cu acestea pe calculatoarele independente.
- Nu rulați MTA dacă nu sunteți un server de mail.

● ... Oprîți serviciile prin comanda **chkconfig**, scripturile de inițializare sau prin editarea fișierelor de configurare (**x**)**inetd**.

10.5.3. Actualizați regulat

Abilitatea de a se adapta rapid la un mediu în schimbare este ceea ce face ca Linux-ul să fie din ce în ce mai atractiv. Dar apare și posibilitatea ca actualizări de securitate să fie disponibile chiar în momentul în care instalați o versiune nou-nouă, aşa că primul lucru pe care trebuie să-l faceți (și este valabil pentru orice fel de sistem de operare) după instalare este căutarea actualizărilor. După aceasta, actualizați toate pachetele pe care le utilizați (regulat).

Anumite actualizări pot cere fișiere noi de configurare, în acest caz fișierele vechi fiind înlocuite. Verificați documentația pentru a vă asigura că totul merge bine și după actualizare.

10.5.4. Firewall-uri și politicile de acces

10.5.4.1. Ce este un firewall?

În secțiunea precedentă am adus vorba despre posibilitatea configurării ca firewall a unui sistem Linux. Cu toate că administrarea firewall-ului este o sarcină pentru administratorul de rețea, trebuie să cunoașteți câteva lucruri despre acesta.

Firewall este un termen vag, care poate desemna orice lucru care se comportă ca o barieră protectivă între noi și lumea exterioară, în general Internetul. Firewall poate fi atât un sistem dedicat cât și o aplicație care asigură această funcționalitate. Poate fi o combinație a acestora, inclusiv o sumă de dispozitive și programe. Firewall-urile sunt clădite pe „reguli” care sunt utilizate pentru definirea a ceea ce este permis a intra și/sau ieși dintr-un sistem dat sau rețea.

După întreruperea serviciilor care nu vă sunt necesare, trebuie să restricționați serviciile permise pentru a reduce la minim numărul conexiunilor necesare. Un bun exemplu este cazul în care lucrați acasă: doar conexiunea dintre biroul dumneavoastră și calculatorul de acasă trebuie permisă, noi recomandându-vă interzicerea conexiunile cu celelalte calculatoare de pe Internet.

10.5.4.2. Filtrele de pachete

Prima linie de apărare este constituită din *filtrele de pachete*, care pot căuta în interiorul pachetelor IP după care iau o decizie în funcție de conținutul acestora. Cel mai întâlnit este pachetul Netfilter, care conține comenzi pentru **iptables**, filtrul de pachete de nouă generație pentru Linux.

Una din cele mai notabile îmbunătățiri din kerenelele noi este capacitatea *stateful inspection* care nu doar că spune ce este în interiorul unui pachet, dar poate spune și dacă pachetul aparține sau este în legătură cu o conexiune nouă sau existentă. Shoreline Firewall sau, pe scurt, Shorewall, este interfața pentru firewall-ul Linux-ului (în ultima vreme se folosește Firestarter ca interfață grafică pentru **iptables** sau , chiar mai recent, ufw^{n.t.}).

Mai multe informații pot fi găsite la pagina proiectului [Netfilter/iptables](#).

10.5.4.3. TCP wrappers

TCP wrapping-ul asigură, în general, aceleași rezultate ca filtrarea pachetelor, dar lucrează în alt mod. Încercarea de realizare a unei conexiuni este acceptată, apoi sunt examineate fișierele de configurare, hotărându-se astfel acceptarea sau respingerea cererii de conectare. Controlează conexiunile la nivel de aplicații și nu la nivelul rețelei.

Se folosește de regulă cu **xinetd** pentru a furniza numele gazdei și controlul accesului pe baza adresei de IP. În plus, aceste unelte dețin posibilități de administrare a autentificărilor ușor de configurat și sunt accesibile în utilizare.

Două din avantaje: calculatorul care se conectează nu are de unde să știe că acest wrapper este utilizat și serviciul operează separat de aplicațiile pe care le protejează.

Accesul este controlat prin fișierele hosts.allow și hosts.deny. Fișierele următoare conțin documentația suplimentară la care doriți eventual să apelați: /usr/share/doc/tcp_wrappers [-<version>/] și /usr/share/doc/tcp. Paginile man care tratează fișierele de control al accesului bazat pe gazde oferă și exemple pe care le puteți studia.

10.5.4.4. Proxy

Serverele proxy pot îndeplini multe sarcini, nu toate având legătură cu securitatea. Dar faptul că acestea sunt, în esență, intermediari, face ca ele să fie un nod în care să se implementeze politici de control al accesului, de limitări ale conexiunilor directe prin intermediul unui firewall și să se configureze modul în care rețea din spatele serverului proxy acționează Internetul.

De regulă în combinație cu un filtru de pachete, și câteodată doar prin ele însăși, serverele proxy asigură un nivel în plus de control. Mai multe informații găsiți în [Firewall HOWTO](#) sau pe site-ul Squid.

10.5.4.5. Accesarea aplicațiilor individuale

Anumite servicii pot avea propriile metode de control al accesului. Exemple la îndemâna sunt Samba, X Window, Bind, Apache și CUPS. Verificați ce fișiere de configurare dictează comportamentul fiecărui serviciu în parte.

10.5.4.6. Fișierele jurnal

Mai mult decât orice altceva, stilul UNIX prin care se consumnează tot felul de activități în anumite tipuri de fișiere, ne confirmă că „se face ceva”. Fișierele jurnal ar trebui verificate regulat, manual sau automat. Firewall-urile și alte mijloace de control al accesului tind să creeze fișiere jurnal mari, aşa că vă sfătuim să încercați determinarea jurnalizării doar a evenimentelor anormale.

10.5.5. Detectarea intruziunilor

Intrusion Detection System (Sistemul pentru detectarea intruziunilor) este proiectat să semnaleze toate amenințările care au trecut prin firewall. Poate să surprindă o activitate de pătrundere în desfășurare sau să detecteze o pătrundere reușită după realizarea acesteia. În acest din urmă caz este prea târziu să vă gândiți la prevenirea pagubelor, dar, cel puțin, sunteți avertizați din vreme asupra problemei. Există două tipuri IDS de bază: cele care protejează rețele și cele care protejează calculatoare individuale.

În cazul IDS-urilor care pun accentul pe gazde, sunt folosite utilități care monitorizează sistemul de fișiere, în căutarea modificărilor apărute. Fișierele de sistem care au suferit schimbări care nu ar trebui să apară sunt un indiciu că ceva nu este în ordine. Oricine ar pătrunde în sistem și ar obține acces la contul de root va determina o modificare undeavă în sistem. De obicei este primul lucru pe care l-ar face un intrus, pentru a putea folosi apoi o poartă de intrare sau pentru a lansa un atac asupra altcuiva; în ambele cazuri se aduc modificări sau adăugiri fișierelor de sistem. Anumite sisteme includ pachetul de monitorizare a sistemului [tripwire](#), a cărui documentație o găsiți pe site-ul [Tripwire Open Source Project](#).

Detectarea intruziunilor în rețea este realizată printr-un sistem care analizează traficul care trece prin firewall (nu scannere de porturi, la care porturile reacționează). [Snort](#) este un exemplu de astfel de program Open Source. Whitehats.com furnizează o bază de date deschisă Intrusion detection, [arachNIDS](#).

10.5.6. Mai multe sfaturi

Iată câteva lucruri bine de ținut minte:

- Nu vă autentificați ca administrator dacă nu aveți treabă. Dezvoltatorii UNIX au născocit acum mai bine de douăzeci de ani **su** pentru întărirea securității.
- Accesul direct ca root este întotdeauna periculos și permite erori umane, aşa că folosiți comanda **sudo** sau autentificați-vă pe un cont separat de administrator când este nevoie. Mult mai bine din punct de vedere al securității este folosirea comenzi **sudo** pentru executarea acelor comenzi care necesită privilegii speciale, ceea ce vă oferă o perioadă limitată de acțiune ca administrator, după care se

revine automat la mediul normal de lucru.

- Luați în serios parolele. Folosiți parole criptate. Schimbați-vă parola regulat.
 - Încercați să folosiți întotdeauna SSH sau SSL. Evitați **telnet**, FTP și clienții de e-mail precum și alte programe client care trimit parole în clar pe rețea. Securitate nu înseamnă doar securizarea calculatorului, ci și securizarea parolelor dumneavoastră.
 - Limitați resursele sistemului prin folosirea **quota** sau/și **ulimit**.
 - Mesajele pentru root trebuie livrate, sau măcar citite de, către o persoană, nu către un sistem automat.
 - [Institutul SANS](#) vă oferă mai multe sfaturi, sortate în funcție de distribuție, printr-un serviciu de poștă electronică (liste de mail).
 - Verificați sursa programelor noi, luați-le dintr-un loc sau sait de încredere. Verificați pachetele noi înainte de instalare.
 - Dacă folosiți o conexiune la Internet temporară, închideți-o imediat ce nu mai aveți nevoie de aceasta.
 - Rulați serviciile de care aveți nevoie pe porturi aleatorii, în loc să folosiți porturile obișnuite, cunoscute de hackeri.
 - Cunoașteți-vă sistemul. După o vreme, puteți chiar simți că se întâmplă ceva aiurea.
-

10.5.7. Am fost victimă unui atac?

Cum puteți afla? Iată o listă cu evenimente care ar trebui să vă dea de gândit:

- Porturi misterioase deschise, procese ciudate.
 - Utilități de sistem (comenzi obișnuite) care se comportă straniu.
 - Probleme de autentificare.
 - Consumarea inexplicabilă a lățimii de bandă.
 - Fișiere raport alterate sau lipsă, comportament ciudat al serviciului de raportare a activității sistemului.
 - Interfețele sunt afișate în moduri neobișnuite.
 - Modificări neașteptate ale fișierelor de configurare.
 - Intrări stranii în fișierele care păstrează activitatea (istoricul) consolei.
 - Fișiere temporare neidentificate.
-

10.5.8. Refacerea de pe urma unei intruziuni

Pe scurt, fiți calmi. Apoi acționați în ordinea de mai jos:

- Deconectați calculatorul de la rețea.
 - Încercați să aflați cât de multe puteți despre modul în care v-a fost compromisă securitatea.
 - Faceți copii de siguranță ale fișierelor, datelor importante, ce nu țin de sistem. Dacă este posibil, verificați aceste date cu datele păstrate drept copii de siguranță, făcute înainte ca sistemul să fie compromis, pentru a vă asigura de integritatea acestora.
 - Reinstalați sistemul.
 - Folosiți parole noi.
 - Restaurați datele folosind copiile de siguranță.
 - Actualizați sistemul.
 - Reexaminați sistemul: blocați serviciile care nu vă sunt necesare, verificați regulile firewall-ului și politicile de acces.
 - Conectați-vă din nou la Internet.
-

10.6. Sumar

Linux-ul este conceput special pentru a lucra în rețea. Kernelul acestui sistem de operare integrează protocolele folosite în mod obișnuit dar și unele din cele mai specifice. Unele standard de rețea ale UNIX-ului sunt prezente în orice distribuție. Pe lângă acestea, cele mai multe distribuții oferă unele care să ajută la instalarea și administrarea rețelelor.

Linux-ul este renumit ca platformă stabilă pe care rulează numeroase servicii de Internet, numărul programelor disponibile pe Internet fiind uriaș. Precum UNIX-ul, puteți folosi și administra un sistem Linux de la distanță, folosind una din soluțiile care asigură executarea la distanță a programelor.

Subiectul securității a fost atins în treacăt. Linux-ul este alegerea ideală pentru utilizarea ca firewall, o soluție ieftină și la îndemână, dar poate fi folosit în alte câteva funcții de rețea, cum ar fi routere și servere proxy.

Întărirea securității unei rețele se face prin aplicarea frecventă a actualizărilor și prin folosirea bunului-simt.

Îată o trecere în revistă a unor comenzi legate de rețelistică:

Tabelul 10-2. Comenzi noi în Capitolul 10: Rețelistica

Comanda	Înțelesul comenzi
ftp	Transferă fișierele către alt calculator (nesecurizat).
host	Afișează informații despre calculatoare din rețea.
ifconfig	Afișează informații despre adresa de IP.
ip	Afișează informații despre adresa de IP.
netstat	Afișează informații despre rutare și statistici de rețea.
ping	Trimite cereri de răspuns către alte calculatoare.
rdesktop	Afișează un spațiu de lucru MS Windows pe sistemele Linux.
route	Arată informațiile legate de rutare.
scp	Copiere securizată a fișierelor de la și către alt calculator.
sftp	Transmitere securizată de fișiere către și de la un alt calculator.
ssh	Conexiune securizată cu alt calculator.
ssh-keygen	Generează chei de autentificare pentru Secure SHell.
telnet	Realizează o conexiune nesecurizată cu alt calculator.
tracepath/traceroute	Afișează ruta pe care o urmează pachetele către alt calculator.
whois	Obține informații despre un nume de domeniu.
xclock	Aplicație X Window, ideală pentru testarea unei conexiuni la distanță.
xhost	Unealtă de control al accesului pentru X Window.

10.7. Exerciții

10.7.1. Generalități despre rețelistică

- Afișați informațiile despre rețea în cazul stației dumneavoastră de lucru: adresa de IP, rutările, numele serverelor de domenii.
- Să presupunem că nu este disponibil nici un DNS. Ce puteți face pentru a accesa un calculator din rețea fără a introduce adresa de IP tot timpul?
- Cum puteți stoca permanent informațiile serverului proxy pentru un navigator în modul text, cum ar fi **links**?
- Ce nume de server operează domeniul redhat.com?

- Trimiteti un mesaj de poștă electronică contului dumneavoastră local. Încercați două metode de trimitere și de citire. Cum puteți verifica dacă l-ați primit?
 - Calculatorul dumneavoastră acceptă conexiuni FTP anonime? Cum folosiți programul **ncftp** pentru autentificarea cu numele dumneavoastră de utilizator și parolă?
 - Calculatorul dumneavoastră rulează un server web? Dacă nu, faceți-o. Verificați fișierele jurnal!
-

10.7.2. Conexiuni la distanță

- De la stația dumneavoastră de lucru, afișați o aplicație grafică, cum ar fi **xclock**, pe ecranul unui alt calculator. Conturile de care aveți nevoie trebuie configurate. Folosiți o conexiune securizată!
 - Configurați cheile SSH pentru a vă putea conecta la un alt calculator fără a avea nevoie de o parolă.
 - Faceți o copie de rezervă a directorului dumneavoastră personal în directorul /var/temp al celuilalt calculator, pe care-l folosiți ca „server pentru copii de siguranță”, utilizând **scp**. Arhivați și comprimați datele înainte de transfer! Conectați-vă la distanță folosind **ssh**, despachetați copia de siguranță și trimiteți un fișier înapoi la calculatorul dumneavoastră folosind **sftp**.
-

10.7.3. Securitate

- Faceți o listă cu toate porturile deschise (care ascultă) de pe calculatorul dumneavoastră.
 - Să presupunem că doriți să rulați un server web. Ce servicii ați dezactivat? De ce ați face acest lucru?
 - Instalați actualizările disponibile.
 - Cum puteți vedea cine s-a conectat la calculatorul dumneavoastră?
 - Creați o sarcină repetitivă care să vă reamintească să vă schimbați parola de utilizator în fiecare lună, și pe cea de root, de asemenea.
-

Capitolul 11. Sunet și video

Acest capitol tratează următoarele (pe scurt, întrucât domeniul audio și video este foarte larg):

- ◆ Configurarea plăcii de sunet
- ◆ Redarea CD-urilor, copierea CD-urilor
- ◆ Redarea fișierelor de muzică
- ◆ Controlul volumului
- ◆ Video și televiziune
- ◆ Înregistrare de sunet

11.1. Generalități despre domeniul audio

11.1.1. Instalare

De regulă, sistemul are instalate drivere (module de nucleu) audio iar configurarea a fost făcută la momentul instalării. Dacă doriți să vă schimbați echipamentul audio, cele mai multe sisteme asigură unelte care vă permit instalarea și configurarea fără probleme a acestui tip de dispozitive. Cele mai recente plăci de sunet cu capabilități plug-and-play ar trebui să fie recunoscute automat. Dacă auziți mostrele de sunet care sunt redate pe parcursul instalării, apăsați OK și totul va fi configurat automat.

Dacă placa de sunet nu este detectată automat, va fi prezentată o listă care conține tipurile de plăci de sunet și/sau proprietățile acestora, din care puteți alege. După aceea, trebuie să furnizați portul I/O corect, configurările IRQ și DMA. Informații despre aceste configurații găsiți în documentația plăcii de sunet. Dacă folosiți un sistem dual-boot care include și MS Windows, puteți găsi informațiile în Panoul de control al acestui sistem de operare.



Dacă detectarea automată a plăcii de sunet eșuează

Dacă placa de sunet nu este recunoscută implicit de sistem, trebuie să apelați la alte tehnici. Acestea sunt descrise în [Linux Sound HOWTO](#).

11.1.2. Drivere și arhitecturi

Există, în general, două tipuri de arhitectură de sunet: mai vechiul Open Sound System sau OSS, care merge cu orice sistem bazat pe UNIX, și mai nou Advanced Linux Sound Architecture sau ALSA, care lucrează mai bine cu Linux-ul, după cum sugerează și numele. ALSA include mai multe facilități și permite dezvoltarea mai rapidă a driverelor. Ne vom axa pe sistemul ALSA.

Aproape toate procesoarele audio mai des folosite lucrează cu ALSA. Doar anumite soluții profesionale din segmentul de vârf și anumite plăci de sunet dezvoltate de fabricanți care refuză să pună la dispoziție documentația procesoarelor pe care le utilizează nu sunt compatibile. O trecere în revistă a dispozitivelor pentru care există suport poate fi găsită pe saitul ALSA, la <http://www.alsa-project.org/alsa-doc/index.php?vendor>All#matrix>.

Configurarea sistemelor instalate cu ALSA se face prin unealta **alsactl**. În plus, anumite distribuții asigură unelte proprii pentru configurarea plăcii de sunet; acestea pot integra atât calea mai veche cât și pe cea nouă pentru administrarea dispozitivelor de sunet.

11.2. Redarea sunetului și a fișierelor video

11.2.1. Redarea și copierea CD-urilor

Pachetul **cde** este inclus în aproape orice distribuție și asigură **cdp** sau **cdplay**, un player CD în modul text. Administratorii de ecran includ o unealtă grafică, cum ar fi **gnome-cd** player în Gnome, care poate fi pornită din meniu grafic.

Trebuie să înțelegeți diferența dintre un CD audio și unul care conține date. Nu trebuie să montați în sistemul de fișiere un CD audio pentru a-l asculta. Asta deoarece datele de pe astfel de CD-uri nu sunt

fișiere de sistem proprii Linux-ului; ele sunt accesate și trimise pe canalul de ieșire audio direct, folosind un program de redare pentru CD-uri audio. Dacă vorbim despre un CD care conține fișiere .mp3, trebuie ca să montați acest CD în sistemul de fișiere, apoi să utilizați unul din programele pe care le vom discuta mai jos pentru a reda muzica înregistrată pe acest disc. Am discutat în [Secțiunea 7.5.5](#) despre montarea CD-urilor în sistemul de fișiere.

Unealta **cdparanoia** din pachetul cu același nume citește formatul audio direct de pe CD ca date, fără conversie analog, și scrie datele într-un fișier sau le livrează într-un alt format, dintre care mai folosit este .waw. Cele mai multe distribuții includ diferite unele de conversie a formatelor, de exemplu în .mp3, sau pun la dispoziție pachete de programe, pe Internet, pentru aceste sarcini. Proiectul GNU furnizează câteva playere programe pentru redarea CD-urilor, extragere de fișiere audio și unele de conversie a formatelor, precum și administratori de baze de date pentru fișierele audio; consultați [Free Software, Audio Section](#) pentru informații detaliate.

Crearea de CD-uri audio se face mai ușor, printre multe altele, cu unealta **kaudiocreator** din suita KDE. Informații clare sunt conținute de KDE Help Center (Centrul de ajutor KDE).

Scrierea CD-urilor a fost tratată în [Secțiunea 9.2.2](#).

11.2.2. Redarea fișierelor audio

11.2.2.1. Fișierele .mp3

Formatul .mp3 este citit de sistemele Linux. Cele mai multe distribuții includ numeroase programe care pot reda aceste fișiere. Printre multe altele, XMMS, prezentat în imaginea de mai jos, este mai apreciat, în parte datorită aspectului asemănător cu programul Winamp din Windows.

Figura 11-1. Programul de redare a fișierelor muzicale XMMS



Foarte utilizate sunt și aplicația KDE Amarok, care câștigă constant popularitate, și MPlayer, care, în plus, poate reda fișiere video.



Restricții

Anumite distribuții nu permit redarea fișierelor MP3 fără modificarea configurației, din cauza restricțiilor de licențiere a formatului MP3. Va fi nevoie de instalarea unor programe în plus pentru a reuși redarea acestui format.

■ Este și cazul distribuției Ubuntu, care, în instalarea standard, nu redă acest format. Trebuie să activați arhivele de programe universe și multiverse pentru a instala pachetele adiționale ubuntu-restricted-extras care se ocupă, printre altele, și de redarea formatului .mp3. Veți avea nevoie și de pachete adiționale pentru redarea DVD-urilor. De menționat că în varianta KiwiLinux, un Ubuntu conceput pentru România și Ungaria, aceste formate sunt suportate în instalarea standard^{n.t.}. ■

În modul text, folosiți comanda **mplayer**:

```
[tille@octarine ~]$ mplayer /opt/mp3/oriental/*.mp3
MPlayer 1.0pre7-RPM-3.4.2 (C) 2000-2005 MPlayer Team
CPU: Advanced Micro Devices Duron Spitfire (Family: 6, Stepping: 1)
Detected cache-line size is 64 bytes
CPUFlags: MMX: 1 MMX2: 1 3DNow: 1 3DNow2: 1 SSE: 0 SSE2: 0
```

```

Playing /opt/oldopt/mp3/oriental/Mazika_Diana-Krozon_Super-Star_Ensani-Ma-
Bansak.mp3.
Cache fill: 1.17% (98304 bytes)      Audio file detected.
Clip info:
Title: Ensani-Ma-Bansak.mp3
Artist: Diana-Krozon
Album: Super-Star
Year:
Comment:
Genre: Unknown
=====
Opening audio decoder: [mp3lib] MPEG layer-2, layer-3
mpg123: Can't rewind stream by 450 bits!
AUDIO: 44100 Hz, 2 ch, s16le, 160.0 kbit/11.34% (ratio: 20000->176400)
Selected audio codec: [mp3] afm:mp3lib (mp3lib MPEG layer-2, layer-3)
=====
Checking audio filter chain for 44100Hz/2ch/s16le -> 44100Hz/2ch/s16le...
AF_pre: 44100Hz/2ch/s16le
AO: [oss] 44100Hz 2ch s16le (2 bps)
Building audio filter chain for 44100Hz/2ch/s16le -> 44100Hz/2ch/s16le...
Video: no video
Starting playback...
A: 227.8 (03:23:.1) 1.8% 12%

```

11.2.2.2. Alte formate

Discutarea tuturor formatelor audio și a modului de redare pentru acestea ne plasează dincolo de scopul acestui ghid. Vă prezentăm doar o trecere (incompletă) în revistă a celor mai folosite programe de redare și administrare a fișierelor audio:

- Ogg Vorbis: format audio liber: consultați [directorul audio GNU](#) pentru unelte - acestea pot fi deja incluse în distribuția dumneavoastră. Formatul a fost dezvoltat deoarece formatul MP3 a devenit proprietar. ■ De menționat că Ogg este superior atât din punct de vedere al comprimării, cât și al naturaleții sunetului redat; din păcate, puțin folosit^{11.2.2.2.1.} ■
- Real audio și video: [realplay](#) de la [RealNetworks](#).
- SoX sau Sound eXchange: de fapt un convertor de formate, care este inclus în programul [play](#). Redă .waw, .ogg și alte formate, inclusiv cele binare brute.
- Playmidi: un player midi, consultați directorul GNU.
- AlsaPlayer: inclus în proiectul Advanced Linux Sound Arhitecture, consultați saitul [AlsaPlayer](#).
- [mplayer](#): redă practic orice, inclusiv fișiere mp3. Mai multe informații găsiți pe saitul [MPlayerHQ](#).
- [hxplayer](#): integrează suport pentru fișierele RealAudio și RealVideo, mp3, mp4audio, Flash, waw și altele, consultați [HelixDNA](#) (nu toate componentele de mai sus ale acestui program sunt libere).
- [rhythmbox](#): construit pe cadrul GStreamer, poate reda orice format compatibil cu acesta din urmă, care pretinde că redă tot, consultați [Rhythmbox](#) și [GStreamer](#).

Verificați documentația sistemului și paginile man ale programelor care vă interesează pentru explicații detaliate despre modul acestora de folosință.



Nu am aceste aplicații instalate pe sistemul meu!

Multe din aplicațiile și uneltele discutate în secțiunea de mai sus sunt programe opționale. Există aşadar posibilitatea ca astfel de aplicații să nu fie prezente în sistemul dumneavoastră după instalarea implicită, dar sunt disponibile pentru distribuția aleasă ca pachete adiționale. Se poate întâmpla însă și ca unul din programele menționate mai sus să nu fie disponibil pentru distribuția dumneavoastră. În acest caz, descărcați programul de pe Internet.

11.2.2.3. Controlul volumului

aumix și **alsamixer** sunt două unelte des folosite pentru controlul volumului. Folosiți tastele săgeți pentru navigarea prin câmpurile disponibile. **alsamixer** dispune de o interfață grafică pentru meniul Gnome sau prin comanda **gnome-alsamixer**. Unealta **kmix** face același lucru în KDE.

Indiferent de maniera în care alegeți să ascultați muzică sau alte fișiere audio, amintiți-vă că ceilalți oameni nu sunt interesați de ceea ce se redă pe calculatorul dumneavoastră. Încercați să vă purtați civilizați, în special acolo unde se lucrează. Folosiți căști de calitate în locul celor care sunt introduse în pavilionul urechii, pentru a nu distrage pe cei cu care împărtășiți același mediu.

11.2.3. Înregistrarea

Aveți la dispoziție destule unelte pentru înregistrarea de muzică sau voce. Pentru înregistrare de voce puteți folosi **arecord** în linia de comandă:

```
alexey@russia:~> arecord /var/tmp/myvoice.wav
Recording WAVE '/var/tmp/myvoice.wav' : Unsigned 8 bit, Rate 8000 Hz, Mono
Aborted by signal Interrups...
```

„Interrupt” înseamnă că aplicația a primit semnalul **Ctrl+C**. Redați înregistrarea prin comanda **play**.

Ceea ce v-am prezentat mai sus este un test pe care-l puteți face înainte de testarea aplicațiilor care au nevoie de voce, cum ar fi Voice over IP (VoIP). Nu uitați să verificați dacă microfonul este activat. Dacă nu vă auziți vocea, verificați-vă configurațiile de sunet. Adesea se întâmplă ca microfonul să fie închis sau volumul acestuia să fie foarte jos. Puteți ușor reface configurațiile prin **alsamixer** sau interfața grafică pentru volum specifică distribuției dumneavoastră.

Din KDE puteți încerca **krec**, iar din Gnome - **gnome-sound-recorder**.

11.3. Redarea video, redare flux și televiziune

Sunt disponibile mai multe instrumente:

- **xine**: player video liber.
- **ogle**: player DVD.
- **okle**: versiunea KDE a **ogle**.
- **mplayer**: player pentru filme al Linux-ului.
- **totem**: redă fișiere video și audio, CD-uri audio, VCD și DVD.
- **realplay**: de la RealNetworks.
- **hxplay**: o alternativă pentru Real, consultați [HelixDNA](#).
- **kaffeine**: player media pentru KDE.

Veți găsi cu siguranță unul din aceste instrumente în meniul grafic.

Rețineți că nu sunt disponibile în instalarea implicită toate codecurile necesare pentru redarea diferitelor formate video. Va trebui să descărcați codecurile w32 și/sau libdvdcss.

[LPD](#) a pus la dispoziție un document care completează această secțiune. Se numește [DVD Playback HOWTO](#) și descrie diversele instrumente disponibile pentru redarea filmelor pe sistemele care au unitate DVD. Este o adăugire binevenită la [DVD HOWTO](#), care explică modul de instalare pentru unitățile DVD.

Pentru vizionarea programelor TV puteți alege din programele următoare, existând, bineînțeles, și altele:

- **tvtimer**: un program foarte bun, cu administrator al stațiilor, interactiv cu teletextul, mod film și [multe altele](#).
 - **zapping**: program pentru televiziune specific Gnome.
 - **xawtv**: program pentru TV specific X11.
-

11.4. Telefonie Internet

11.4.1. Ce este?

Telefonia Internet sau, altfel spus, Voice over IP (VoIP) ori telefonia digitală, permite părților să schimbe fluxuri de date (voce) pe rețea. Mareea diferență este aceea că datele sunt purtate pe o rețea cu scopuri generale, Internetul, contrar telefoniei clasice, care utilizează o rețea dedicată pentru transmisia de voce. Cele două rețele pot fi conectate totuși în circumstanțe speciale, dar, pentru moment, acest lucru nu este un standard. Cu alte cuvinte: este de așteptat să nu puteți suna pe cineva care dispune de o linie telefonică clasică. Dacă este posibil, va trebui să plătiți un abonament pentru acest lucru.

În timp ce există numeroase aplicații pentru tot felul de scopuri, atât libere, cât și proprietare, în domeniul telefoniei Internet nu s-au înregistrat progrese spectaculoase. Sistemul nu este de încredere, poate fi încet sau să aibă un nivel de zgomot de fond destul de ridicat, aşa că nu poate înlocui sub nici o formă telefonia conventională - gândiți-vă cel puțin la apelurile de urgență. Cu toate că furnizorii își iau precauțiile necesare, nu există nici o garanție că veți reuși efectuarea unei astfel de convorbiri.

Cele mai multe aplicații nu folosesc criptarea, aşa că fiți precauți, deoarece este destul de ușor ca cineva să tragă cu urechea. Dacă vă gândiți serios la securitate, citiți documentația care însoțește clientul dumneavoastră VoIP. În plus, dacă folosiți un firewall, trebuie configurat să permită conexiuni de oriunde, astfel că folosirea unui asemenea program ridică numeroase semne de întrebare legate de nivelul de securitate al sistemului dumneavostră.

11.4.2. Ce vă trebuie?

11.4.2.1. Partea de server

În primul rând, aveți nevoie de un furnizor care să vă ofere acest serviciu. Serviciul oferit poate integra și telefonia tradițională, poate fi gratuit sau nu. Menționăm câțiva astfel de furnizori: [SIPphone](#), [Vonage](#), [Lingo](#), [AOL TotalTalk](#) și mulți alți furnizori locali care oferă așa numitul „serviciu telefonic integral”. Servicii de telefonie numai pe Internet sunt oferite de [Skype](#), [SIP Broker](#), [Google](#).

Dacă doriți să vă faceți un server propriu, aplecați-vă asupra [Asterisk](#).

11.4.2.2. Partea de client

Pe partea de client, aplicațiile pe care le folosiți depind de configurația rețelei. Dacă dispuneți de o conexiune directă la Internet, nu veți avea probleme, presupunând că știți la ce server să vă conectați și că aveți alocat un nume de utilizator și o parolă pentru autentificarea la serviciu.

Dacă sunteți în spatele unui firewall care folosește Network Address Translation (NAT), există posibilitatea ca anumite servicii să nu funcționeze, întrucât este vizibilă doar adresa IP a firewall-ului, nu și pe cea a calculatorului, ceea ce duce la imposibilitatea rutării pe Internet, de exemplu atunci când vă aflați în rețeaua companiei și adresa IP începe cu 10., 192.168. sau un alt prefix de subnet care nu poate fi rutat. Acest lucru depinde, totuși, de protocolul utilizat de aplicație.

De asemenea, lățimea de bandă poate fi un factor de blocaj: anumite aplicații sunt optimizate pentru consumarea unei lățimi de bandă mai mici, în timp ce altele pot solicita conexiuni de bandă largă. Aceste nevoi sunt impuse de codurile folosite de aplicații.

Printre cele mai cunoscute aplicații putem enumera Skype, cu o interfață care amintește de cea a mesageriilor instantanee, și [X-Lite](#), versiunea gratuită a XTen softphone, care arată ca un telefon mobil. Cu toate că aceste programe sunt disponibile gratuit și sunt folosite pe scară largă, ele nu sunt libere: acestea folosesc protocoale proprietare și/sau sunt disponibile doar în pachete binare, nu în format cod sursă.

Programe VoIP gratuite și libere sunt: [Gizmo](#), [Linphone](#), [GnomeMeeting](#) și [KPhone](#).



Echipamente

Chiar dacă dispuneți de un microfon incorporat în calculatorul dumneavoastră, în special dacă acesta este un laptop, veți obține rezultate mai bune dacă folosiți un set dedicat microfon-căști. Dacă sunteți în măsură să decideți, achiziționați un set care lucrează pe USB, întrucât funcționează independent de dispozitivele audio existente pe calculator. Folosiți [alsamixer](#) pentru configurarea intrărilor și ieșirilor pentru

nivelurile de sunet după preferințele dumneavoastră.

Aplicațiile VoIP se bucură de o piață în creștere extraordinar de rapidă. Voluntari din toată lumea încearcă să documenteze stadiul actual la <http://www.voip-info.org/>.

11.5. Sumar

Platforma GNU/Linux integrează total capacitați multimedia. O mare varietate de dispozitive precum plăci de sunet, tv-tunere, microfoane, playere CD și DVD sunt acoperite de Linux. Lista aplicațiilor este practic fără de sfârșit, așa că vă prezentăm o listă trunchiată cu noi comenzi, limitându-ne la comenzile audio generale:

Tabelul 11-1. Comenzi noi în Capitolul 11: Audio

Comanda	Înțelesul comenzi
alsactl	Configurează sistemul de sunet ALSA.
alsamixer	Ajustează nivelurile de ieșire a sunetului pentru driverul ALSA.
arecord	Înregistrează o moștă de sunet.
aumix	Mixer audio.
cdp	Redă un CD audio.
cdparanoia	Extrage pistele audio de pe un CD audio.
cdplay	Redă un CD audio.
gnome-alsamixer	Interfață grafică din Gnome pentru ALSA.
gnome-cd	Interfață Gnome pentru redarea CD-urilor audio.
gnome-sound-recorder	Interfață Gnome pentru înregistrare de sunet.
kaudiocreator	Interfață KDE pentru crearea CD-urilor audio.
kmix	Interfață KDE pentru ajustările de sunet.
krec	Interfață KDE pentru înregistrare de sunet.
mplayer	Player multimedia
play	Unealtă a liniei de comandă pentru redarea mostrelor de sunet.

11.6. Exerciții

1. Din meniul Gnome sau KDE, deschideți panoul de configurare a sunetului. Asigurați-vă că boxele sau setul de căști sunt conectate la sistem și stabiliți un nivel de sunet care vă place. Dacă aveți un sistem compatibil ALSA, alegeți panoul de configurare potrivit.
 2. Dacă aveți un microfon, încercați să vă înregistrați propria voce. Verificați ca nivelul de intrare a sunetului să nu fie prea mare, deoarece va aduce tonuri înalte, supărătoare, în conversațiile cu ceilalți sau va transfera zgomotul de fundal. Din linia de comandă, puteți încerca să folosiți comenzile **arecord** și **aplay** pentru înregistrare și redare de sunet.
 3. Localizați fișierele de sunet din sistemul dumneavoastră și redați-le.
 4. Introduceți un CD audio și redați-l.
 5. Căutați un partener de conversație și configurați un program VoIP (probabil va trebui să instalați mai întâi unul).
 6. Puteți asculta un post de radio de pe Internet?
 7. Dacă aveți o unitate DVD și un film pe un astfel de disc, încercați să-l redați.
-

Appendix A. De aici, încotro?

Vă oferim o trecere în revistă a unor lucrări și saituri folositoare.

A.1. Cărți folositoare

A.1.1. Linux în general

- "Linux in a Nutshell" de Ellen Siever, Jessica P. Hackman, Stephen Spainhour, Stephen Figgins, O'Reilly UK, ISBN 0596000251
 - "Running Linux" de Matt Welsh, Matthias Kalle Dalheimer, Lar Kaufman, O'Reilly UK, ISBN 156592469X
 - "Linux Unleashed" de Tim Parker, Bill Ball, David Pitts, Sams, ISBN 0672316889
 - "When You Can't Find Your System Administrator" de Linda Mui, O'Reilly UK, ISBN 1565921046
 - Atunci când cumpărați o distribuție, aceasta va conține și un manual destul de bine pus la punct.
-

A.1.2. Editoare

- "Learning the Vi Editor" de Linda Lamb și Arnold Robbins, O'Reilly UK, ISBN 1565924266
 - "GNU Emacs Manual" de Richard M. Stallman, iUniverse.Com Inc., ISBN 0595100333
 - "Learning GNU Emacs" de Debra Cameron, Bill Rosenblatt și Eric Raymond, O'Reilly UK, ISBN 1565921526
 - "Perl Cookbook" de Tom Christiansen și Nathan Torkington, O'Reilly UK, ISBN 1565922433
-

A.1.3. Console

- "Unix Shell Programming" de Stephen G.Kochan and Patrick H.Wood, Sams Publishing, ISBN 067248448X
 - "Learning the Bash Shell" de Cameron Newham și Bill Rosenblatt, O'Reilly UK, ISBN 1565923472
 - "The Complete Linux Shell Programming Training Course" by Ellie Quigley și Scott Hawkins, Prentice Hall PTR, ISBN 0130406767
 - "Linux and Unix Shell Programming" de David Tansley, Addison Wesley Publishing Company, ISBN 0201674726
 - "Unix C Shell Field Guide" de Gail și Paul Anderson, Prentice Hall, ISBN 013937468X
-

A.1.4. X Window

- "Gnome User's Guide" de the Gnome Community, iUniverse.Com Inc., ISBN 0595132251
 - "KDE Bible" de Dave Nash, Hungry Minds Inc., ISBN 0764546929
 - "The Concise Guide to XFree86 for Linux" de Aron HSiao, Que, ISBN 0789721821
 - "The New XFree86" de Bill Ball, Prima Publishing, ISBN 0761531521
 - "Beginning GTK+ and Gnome" de Peter Wright, Wrox Press, ISBN 1861003811
 - "KDE 2.0 Development" de David Sweet and Matthias Ettrich, Sams Publishing, ISBN 0672318911
 - "GTK+/Gnome Application Development" de Havoc Pennington, New Riders Publishing, ISBN 0735700788
-

A.1.5. Rețele

- "TCP/IP Illustrated, Volume I: The Protocols" de W. Richard Stevens, Addison-Wesley Professional Computing Series, ISBN 0-201-63346-9
- "DNS and BIND" de Paul Albitz, Cricket Liu, Mike Loukides și Deborah Russell, O'Reilly & Associates, ISBN 0596001584
- "The Concise Guide to DNS and BIND" de Nicolai Langfeldt, Que, ISBN 0789722739
- "Implementing LDAP" de Mark Wilcox, Wrox Press, ISBN 1861002211
- "Understanding and deploying LDAP directory services" de Tim Howes și co., Sams, ISBN 0672323168

- "Sendmail" de Brian Costales și Eric Allman, O'Reilly UK, ISBN 1565922220
- "Removing the Spam : Email Processing and Filtering" de Geoff Mulligan, Addison Wesley Publishing Company, ISBN 0201379570
- "Managing IMAP" de Dianna & Kevin Mullet, O'Reilly UK, ISBN 059600012X

A.2. Saituri utile

A.2.1. Informații generale

- [The Linux documentation project](#): toate documentațiile, pagini man, HOWTO, FAQ.
- [LinuxQuestions.org](#): forumuri, documentație, programe pentru descărcat, multe altele.
- [Google for Linux](#): un motor de căutare specializat.
- [Google Groups](#): arhivă a postărilor grupurilor de știri, care include comp.os.linux.
- [Slashdot](#): știri zilnice.
- <http://www.oreilly.com>: cărți despre administrarea sistemelor Linux și a rețelelor, Perl, Java...
- [POSIX](#): standardul.
- [Linux HQ](#): menține o bază de date completă a surselor, actualizărilor și documentațiilor pentru diferitele versiuni ale kernelului Linux.

A.2.2. Referințe specifice diverselor arhitecturi

- [AlphaLinux](#): Linux pentru arhitectura Alpha (de exemplu Digital Workstation).
- [Linux-MIPS](#): Linux pe MIPS (de exemplu SGI Indy).
- [Linux on the Road](#): ghiduri pentru instalarea și rularea Linux-ului pe laptopuri, PDA-uri, telefoane mobile, etc. Fișiere de configurare pentru diverse modele.
- [MkLinux](#): Linux pentru Apple .

A.2.3. Distribuții

- [The Fedora Project](#): sistem de operare dezvoltat de o comunitate sprijinită de RedHat.
- [Mandriva](#)
- [Ubuntu](#) : distribuție care rulează de pe CD, nu trebuie să instalați nimic.
- [Debian](#)
- [TurboLinux](#)
- [Slackware](#)
- [SuSE](#)
- [LinuxISO.org](#): CD-uri cu imagini pentru toate distribuțiile.
- [Knoppix](#): distribuție care rulează de pe CD, nu trebuie să instalați nimic.
- [DistroWatch.com](#): find a Linux that goes with your style.
- ...

A.2.4. Programe

- [Freshmeat](#): programe noi, arhive de programe.
- [OpenSSH](#): saitul Secure SHell.
- [OpenOffice](#): suită de birou compatibilă cu MS Office.
- [KDE](#): saitul K Desktop.
- [GNU](#): GNU și programe GNU.
- [Gnome](#): saitul oficial Gnome.
- [RPM Find](#): toate pachetele RPM.
- [Samba](#): serviciu de partajare fișiere și imprimante cu sistemele MS Windows.
- [Home of the OpenLDAP Project](#): OpenLDAP server/client/utilități, FAQ și documentație.
- [Sendmail Homepage](#): prezentarea facilităților Sendmail, care include exemple de configurare.
- [Netfilter](#): conține informații diverse despre iptables: HOWTO, FAQ, ghiduri ...
- [Saitul oficial GIMP](#) : toate informațiile despre programul GNU de manipulare a imaginilor.
- [SourceForge.net](#): saitul pentru dezvoltarea programelor Open Source.
- [vIm homepage](#)

Appendix B. Comenzi DOS versus comenzi Linux

Vom prezenta comenziile DOS și echivalentul acestora pentru Linux.

Ca un mijloc suplimentar de orientare pentru noii veniți în Linux, dar care au experiență în Windows, tabelul de mai jos listează comenzi MS-DOS și echivalentul acestora în Linux. Comenziile proprii Linux-ului au, de regulă, opțiuni numeroase. Citiți paginile man sau Info ale comenziilor care vă interesează pentru mai multe informații.

Tabelul B-1. Prezentarea comenziilor DOS/Linux

Comenzi DOS	Comenzi Linux
<comandă> /?	man <comandă> sau comandă --help
cd	cd
chdir	pwd
cls	clear
copy	cp
date	date
del	rm
dir	ls
echo	echo
edit	vim (sau alt editor)
exit	exit
fc	diff
find	grep
format	mke2fs sau mformat
mem	free
mkdir	mkdir
more	more sau chiar less
move	mv
ren	mv
time	date

Appendix C. Facilitățile consolei

Vă prezentăm o trecere în revistă a facilităților comune tuturor consolelor, cât și a celor specifice, în funcție de tip.

C.1. Facilități comune

Următoarele facilități există în orice consolă. Comenzile stop, suspend, jobs, bg și fg sunt disponibile numai pentru sistemele care integrează controlul sarcinilor.

Tabelul C-1. Facilități comune tuturor consolelor

Comandă	Înțelesul comenzi
>	Redirecționează ieșirea
>>	Adaugă la fișier
<	Redirecționează intrarea
<<	Redirecționează intrarea
	Conexează ieșirea
&	Rulează procese în fundal
;	Comenzi separate în aceeași linie
*	Fiecare caracter din numele fișierului contează
?	Contează un singur caracter din numele fișierului
[]	Fiecare caracter cuprins între paranteze drepte contează
()	Execută în subconsolă
``	Substituie ieșirea comenzi cuprinse între caractere
'' ''	Citare parțială (permite variabile și expansiuni ale comenzi)
''	Citare în întregime (fără expansiune)
\	Citează caracterul următor
\$var	Folosiți valoare pentru variabilă
\$\$	ID-ul procesului
\$0	Numele comenzi
\$n	Argumentul nr. n (cu n de la 0 la 9)
\$*	Toate argumentele ca simple cuvinte
#	Începe comentariul
bg	Execuție în fundal
break	Pauză
cd	Schimbă directorul
continue	Reia execuția unui program
echo	Afișează ieșirea
eval	Evaluează argumentele
exec	Execută o consolă nouă
fg	Execută în prim plan
jobs	Arată sarcinile active
kill	Stopază sarcinile active
newgrp	Mutare într-un grup nou
shift	Schimbă parametrii pozitionali

stop	Suspendă un serviciu care rulează în fundal
suspend	Suspendă un serviciu care rulează în prim plan
time	Cronometrează o comandă
umask	Configurează sau listează permisiunile asociate fișierelor
unset	Șterge variabile sau definiții de funcții
wait	Așteaptă ca un serviciu din fundal să se termine.

C.2. Facilități particulare

Tabelul de mai jos prezintă diferențele majore dintre consolele standard (**sh**), Bourne Again SHell (**bash**), Korn (**ksh**) și C (**csh**).



Compatibilitățile între console

Întrucât consola Bourne Again SHell este o îmbunătățire a consolei **sh**, toate comenziile **sh** vor funcționa în **bash** - dar nu și invers. **bash** dispune de multe facilități particulare și, după cum arată și tabelul de mai jos, multe altele importante din alte console.

Întrucât consola Turbo C este o îmbunătățire a **csh**, toate comenziile **csh** vor funcționa și în **tcsh**, dar nu și invers.

Tabelul C-2. Facilități particulare tipurilor de console

sh	bash	ksh	csh	Înțeles/Acțiune
\$	\$	\$	%	Prompterul implicit al utilizatorului
	>	>	>!	Forțează redirecționarea
> fișier 2>&1	&> fișier sau fișier 2>&1	> fișier 2>&1	>& fișier	Redirecționează stdout și stderr către fișier
	{ }		{ }	Expandează elementele listate
`comandă`	`comandă` sau \$(comandă)	\$ (comandă)	`comandă`	Substituie ieșirea comenzi
\$HOME	\$HOME	\$HOME	\$home	Directorul personal
	~	~	~	Simbolul directorului personal
	~+, ~-, dirs	~+, ~-	=-, =N	Accesează directoarele
var=value	VAR=value	var=value	set var=value	Numirea variabilelor
export var	export VAR=value	export var=val	setenv var val	Configurează variabilele de mediu
	\$(nnnn)	\$ { nn }		Pot fi desemnate mai mult de 9 argumente
"\$@"	"\$@"	"\$@"		Toate argumentele luate separat drept cuvinte
\$#	\$#	\$#	\$#argv	Numărul argumentelor
\$?	\$?	\$?	\$status	Statutul de ieșire al celei mai recente comenzi
\$!	\$!	\$!		PID-ul celui mai recent proces trimis în fundal
\$-	\$-	\$-		Opțiunile curente
.file	source file sau .file	.file	source file	Citește comenziile din

				file
	alias x='y'	alias x=y	alias x y	Numele x se referă la comanda y
case	case	case	switch sau case	Alege alternativele
done	done	done	end	Termină o declarație în bucătă
esac	esac	esac	endsw	Termină case sau switch
exit n	exit n	exit n	exit (expr)	Ieșire cu afișarea statutului
for/do	for/do	for/do	foreach	Trecere prin variabile
	set -f, set -o nullglob dotglob nocaseglob noglob		noglob	Ignoră caracterele substituite pentru generarea de nume
hash	hash	alias -t	hashstat	Afișează comenzi dispersate (urmărește aliasurile)
hash cmd\$	hash cmd\$	alias -t cmd\$	rehash	Amintește localizarea comenziilor
hash -r	hash -r		unhash	Uită localizarea comenziilor
	history	history	history	Listează comenzi anterioare
	Tasta Săgeată Sus+Enter sau !!	r	!!	Reia comanda anterioară
	!str	r str	!str	Reia ultima comandă care începe cu „str”
	!cmd:s/x/y/	r x=y cmd	!cmd:s/x/y/	Înlocuiește „x”cu „y” în cea mai recentă comandă care începe cu „cmd”, apoi execută
if [\$i -eq 5]	if [\$i -eq 5]	if ((i==5))	if (\$i==5)	Test
fi	fi	fi	endif	Sfârșitul declarației if
ulimit	ulimit	ulimit	limit	Limitează resursele
pwd	pwd	pwd	dirs	Afișează directorul curent
read	read	read	\$<	Citește din terminal
trap 2	trap 2	trap 2	onintr	Ignoră întreruperile
	unalias	unalias	unalias	Șterge aliasurile
until	until	until		Începe bucla until
while/do	while/do	while/do	while	Începe bucla while

Bourne Again SHell dispune de mult mai multe funcții, care nu sunt listate aici. Tabelul arată doar cum această consolă a încorporat idei bune din alte console: nu există spații goale în coloana corespunzătoare consolei **bash**. Mai multe informații despre comenzi particulare Bash găsiți în paginile Info, secțiunea „Bash Features”.

Mai multe informații:

Trebuie să citiți cel puțin un manual, manualul consolei pe care o folosiți. Alegerea pe care vă sfătuim să o faceți este **info bash**, **bash** fiind consola GNU și cea mai prietenoasă cu începătorii. Scoateți-l la imprimantă și luați-l cu dumneavoastră, parcurgeti-l oricând aveți la dispoziție măcar 5 minute.

Consultați [Appendix B](#) dacă aveți dificultăți în asimilarea comenziilor consolei.

Glosar

Secțiunea curentă vă prezintă, în ordine alfabetică, o trecere în revistă a comenziilor discutate în acest document.

A

a2ps

Formatează fișierele pentru tipărire pe o imprimantă PostScript, consultați [Secțiunea 8.1.2](#).

acroread

Vizualizator PDF, consultați [Secțiunea 8.1.2.2](#).

adduser

Creează un utilizator nou sau actualizează informațiile implicate ale unui utilizator.

alias

Creează în consolă un alias pentru o comandă.

alsaconf

Configurează placa de sunet folosind driverul ALSA, consultați [Secțiunea 11.1.2](#).

alsamixer

Regleză sunetul pe ieșirile dispozitivului de sunet ALSA, consultați [Secțiunea 11.2.2.3](#).

anacron

Execută comenzi periodic, nu presupune rularea continuă a calculatorului.

apropos

Caută în baza de date whatis după sirul introdus, consultați [Secțiunea 2.3.3.2](#).

apt-get

Administratorul de pachete APT, consultați [Secțiunea 7.5.3.2](#).

arecord

Înregistrează o mostră de sunet, consultați [Secțiunea 11.2.3](#).

aspell

Verifică ortografia.

at, atq, atrm

Trimite în aşteptare, examinează sau șterge servicii cu executare întârziată, consultați [Secțiunea 4.1.2.2](#) și [Secțiunea 4.4.3](#).

aumix

Ajustează mixerul audio, consultați [Secțiunea 11.2.2.3](#).

(g)awk

Limbaj de scanare și procesare a tiparelor.

B

bash

Bourne Again SHell, consultați [Secțiunea 3.2.3.2](#) și [Secțiunea 7.2.5](#).

batch

Trimite în aşteptare, examinează și șterge sarcinile cu executare întârziată.

bg

Rulează o sarcină în fundal, consultați [Secțiunea 4.1.2.1](#).

bitmap

Editor hartă de pixeli și utilități de conversie pentru sistemul X Window.

bzip2

Un compresor de fișiere, consultați [Secțiunea 9.1.1.3](#).

C

cardctl

Administrează cardurile PCMCIA, consultați [Secțiunea 10.2.3.3](#).

cat

Conexează fișierele și afișează rezultatul la ieșirea standard, consultați [Secțiunea 2.2](#) și [Secțiunea 3.2.4](#).

cd

Schimbă directorul, consultați [Secțiunea 2.2](#).

cdp/cdplay

Un program interactiv în modul text pentru controlarea și redarea CD-urilor audio în Linux, consultați [Secțiunea 11.2.1](#).

cdparanoia

Un utilitar audio pentru citirea CD-urilor, care include facilități de verificare a datelor, consultați [Secțiunea 11.2.1](#).

cdrecord

Înregistrează un CD-R, consultați [Secțiunea 9.2.2.](#)

chattr

Modifică atributele fișierelor.

chgrp

Modifică grupurile.

chkconfig

Actualizează sau cere informații despre nivelurile de rulare pentru serviciile de sistem, consultați [Secțiunea 4.2.5.1.](#)

chmod

Modifică permisiunile asociate fișierelor, consultați [Secțiunea 3.4.1](#), [Secțiunea 3.4.2.1](#) și [Secțiunea 3.4.2.4.](#)

chown

Modifică deținătorii fișierelor și grupurile, consultați [Secțiunea 3.4.2.3.](#)

compress

Comprimă fișierele.

cp

Copiază fișiere și directoare, consultați [Secțiunea 3.3.2.](#)

crontab

Administrează fișierele crontab, consultați [Secțiunea 4.4.4.](#)

csh

Deschide o consolă C, consultați [Secțiunea 3.2.3.2.](#)

cut

Șterge secțiuni din fiecare linie a unui fișier (ale unor fișiere), consultați [Secțiunea 7.2.5.2.](#)

D**date**

Afișează sau modifică data și ora sistemului.

dd

Convertește și copiază un fișier (transfer pe disc - disk dump), consultați [Secțiunea 9.2.1.2.](#)

df

Afișează utilizarea sistemului de fișiere de pe disc, consultați [Secțiunea 3.1.2.3.](#)

dhcpcd

Client DHCP, consultați [Secțiunea 10.3.8.](#)

diff

Caută diferențele dintre două fișiere.

dig

Trimite pachete de interrogare către serverele de nume, consultați [Secțiunea 10.2.6.1.](#)

dmesg

Afișează sau controlează memoria tampon a kernelului.

du

Estimează utilizarea spațiului de către fișiere.

dump

Creează copii de siguranță pentru sistemul de fișiere, consultați [Secțiunea 9.2.5.](#)

E**echo**

Afișează o linie de text, consultați [Secțiunea 3.2.1.](#)

ediff

Diff pentru translatorul în limba engleză.

egrep

grep extins.

eject

Demontează și ejectează mediile detasabile.

emacs

Pornește editorul Emacs, consultați [Secțiunea 6.1.2.1.](#)

exec

Invocă subprocesele, consultați [Secțiunea 4.1.5.1.](#)

exit

Închide consola curentă, consultați [Secțiunea 2.2.](#)

export

Adaugă funcții mediului consolei, consultați [Secțiunea 3.2.1](#), [Secțiunea 7.2.1.2](#) și [Secțiunea 7.2.4.2.](#)

F

fax2ps

Convertește un facsimil TIFF în PostScript, consultați [Secțiunea 8.1.2.](#)

fdformat

Formatează dischete, consultați [Secțiunea 9.2.1.1.](#)

fdisk

Administrator de partitii pentru Linux, consultați [Secțiunea 3.1.2.2.](#)

fetchmail

Administrează mesajele de poștă electronică de pe servere POP, IMAP, ETRN sau ODMR, consultați [Secțiunea 10.3.2.3.](#)

fg

Aduce o sarcină din fundal în prim plan, consultați [Secțiunea 4.1.2.1.](#)

file

Determină tipul unui fișier, consultați [Secțiunea 3.3.1.2.](#)

find

Caută fișiere, consultați [Secțiunea 3.3.3.3.](#)

firefox

Navigator de Internet, consultați [Secțiunea 10.3.3.2.](#)

fork

Creează un proces nou, consultați [Secțiunea 4.1.5.1.](#)

formail

(Re)formatează mesajele de poștă electronică, consultați [Secțiunea 10.3.2.3.](#)

fortune

Afișează un mesaj, la întâmplare, de regulă interesant.

ftp

G

galeon

Navigator grafic de Internet

gdm

Administratorul de spațiu de lucru Gnome, consultați [Secțiunea 4.2.4.](#)

gedit

Editor pentru interfață grafică, consultați [Secțiunea 6.3.3.3.](#)

(min/a)getty

Controlează dispozitivele din consolă.

gimp

Program de manipulare a imaginilor.

gpg

Criptează, verifică și decriptează fișiere, consultați [Secțiunea 9.4.1.2.](#)

grep

Afișează liniile care se potrivesc unui tipar (sortare liniară), consultați [Secțiunea 3.3.3.4](#) și [Secțiunea 5.3.1.](#)

groff

Emulează comanda nroff cu groff, consultați [Secțiunea 8.1.2.](#)

grub

Consola grub, consultați [Secțiunea 4.2.3](#) și [Secțiunea 7.5.4.](#)

gv

Vizualizator PostScript și PDF, consultați [Secțiunea 8.1.2.2.](#)

gvim

Versiunea grafică a editorului vim, consultați [Secțiunea 6.3.3.3.](#)

gzip

Comprimă sau despachetează fișiere, consultați [Secțiunea 9.1.1.3.](#)

H

halt

Oprește sistemul, consultați [Secțiunea 4.2.6.](#)

head

Afișează prima parte a conținutului fișierelor, consultați [Secțiunea 3.3.4.3.](#)

help

Afișează ajutorul pentru comenziile proprii consolei.

host

Utilitar care cauță DNS-urile, consultați [Secțiunea 10.2.6.1.](#)

httpd

Protocol Apache de transfer pentru hipertext, consultați [Secțiunea 10.2.3.1](#).

I**id**

Configurează interfața de rețea sau afișează configurația, consultați [Secțiunea 10.1.2.3](#).

info

Citește documentele Info, consultați [Secțiunea 2.3.3.1](#).

init

Controlează inițializarea proceselor, consultați [Secțiunea 4.1.5.1](#), [Secțiunea 4.2.4](#) și [Secțiunea 4.2.5](#).

insserv

Administrează scripturile init, consultați [Secțiunea 4.2.5.1](#).

iostat

Afișează statisticile I/O, consultați [Secțiunea 4.3.5.4](#).

ipchains

Administrează firewall-ul IP, consultați [Secțiunea 10.4.4.2](#).

iptables

Administrează pachetele IP, consultați [Secțiunea 10.4.4.2](#).

J**jar**

Unealtă de arhivare Java, consultați [Secțiunea 9.1.1.4](#).

jobs

Listează sarcinile care se desfășoară în fundal.

K**kdm**

Administrator pentru spațiul de lucru al KDE, consultați [Secțiunea 4.2.4](#).

kedit

Editor grafic KDE, consultați [Secțiunea 6.3.3.3](#).

kill(all)

Termină procesele, consultați [Secțiunea 4.1.2.1](#).

konqueror

Navigator de fișiere, pentru pagini de ajutor și pentru Internet, consultați [Secțiunea 3.3.2.1](#).

ksh

Deschide o consolă Korn, consultați [Secțiunea 3.2.3.2](#).

kwrite

Editor grafic KDE, consultați [Secțiunea 6.3.3.3](#).

L**less**

more cu mai multe opțiuni.

lilo

Încărător de sisteme pentru Linux (acum se folosește mai mult **grub**), consultați [Secțiunea 4.2](#).

links

Navigator WWW pentru modul text, consultați [Secțiunea 10.2.3.2](#).

In

Creează legături între fișiere, consultați [Secțiunea 3.3.5](#).

loadkeys

Încarcă tabela de modificare a aspectului tastaturii, consultați [Secțiunea 7.4.1](#).

locate

Caută fișiere, consultați [Secțiunea 3.3.3.3](#) și [Secțiunea 4.4.4](#).

logout

Închide consola curentă, consultați [Secțiunea 2.1.3](#).

lp

Trimite cereri către serviciul de tipărire LP, consultați [Secțiunea 8.1](#).

lpq

Program de control pentru tipărire, consultați [Secțiunea 8.1](#).

lpq

Program pentru examinarea lucrărilor trimise la coada pentru tipărire, consultați [Secțiunea 8.1](#).

lpr	Tipărire offline, consultați Secțiunea 8.1 .
lprm	Șterge sarcinile trimise la imprimantă, consultați Secțiunea 8.1 .
ls	Afișează conținutul directoarelor, consultați Secțiunea 2.2 , Secțiunea 3.1.1.2 și Secțiunea 3.3.1.1 .
lynx	Navigator WWW pentru modul text, consultați Secțiunea 10.2.3.2 .

M

mail	Trimit și primește mesaje de poștă electronică, consultați Secțiunea 10.3.2.3 .
man	Citește paginile de manuale, consultați Secțiunea 2.3.2 .
mc	Midnight CCommander, navigator de fișiere, consultați Secțiunea 3.3.2.1 .
mcopy	Copiază fișiere MSDOS în/din UNIX.
mdir	Afișează un director MSDOS.
memusage	Afișează gradul de folosire al memoriei, consultați Secțiunea 4.3.5.3 .
memusagestat	Afișează statistici referitoare la utilizarea memoriei, consultați Secțiunea 4.3.5.3 .
mesg	Controlează accesul la scriere în terminalul dumneavoastră, consultați Secțiunea 4.1.6 .
mformat	Formatează o dischetă în MSDOS, consultați Secțiunea 9.2.1.1 .
mkbootdisk	Creează o dischetă pentru pornirea sistemului.
mksofs	Creează un sistem de fișiere hibrid ISO9660, consultați Secțiunea 9.2.2 .
mplayer	Player pentru filme al Linux-ului, consultați Secțiunea 11.2.2 și Secțiunea 11.3 .
more	Filtru pentru afișarea unei secvențe de text, câte un ecran pe afișare, consultați Secțiunea 3.3.4.2 .
mount	Montează un sistem de fișiere sau afișează informații despre un sistem de fișiere deja montat, consultați Secțiunea 7.5.5.1 .
mozilla	Navigator web, consultați Secțiunea 10.2.3.2 .
mt	Controlează operațiile cu benzile magnetice.
mtr	Unealtă de diagnosticare pentru rețele.
mv	Redenumește fișiere, consultați Secțiunea 3.3.2 .

N

named	Server pentru nume de domenii, consultați Secțiunea 10.3.7 .
nautilus	Administrator de fișiere, consultați Secțiunea 3.3.2.1 .
nctfp	Program de navigare pentru serviciile ftp (nesecurizat!), consultați Secțiunea 10.3.4.2 .
netstat	Afișează conexiunile la rețea, tabela de rutare, statistici referitoare la interfața de rețea, conexiuni deghizate și difuzările multiple, consultați Secțiunea 10.1.2.5 și Secțiunea 10.4.2 .
newgrp	Autentificare într-un alt grup, consultați Secțiunea 3.4.2.2 .
nfsstat	Afișează statistici despre sistemele de fișiere în rețea.
nice	Rulează un program cu modificarea priorității programate, consultați Secțiunea 4.3.5.1 .

nmap

Unealtă pentru explorarea rețelei și scanner de securitate.

ntp

Serviciul Network Time Protocol, consultați [Secțiunea 7.4.3.](#)

ntpdate

Configurează ora și data cu ajutorul unui server NTP, consultați [Secțiunea 7.4.3.](#)

ntsysv

Interfață simplă pentru configurarea nivelurilor de rulare, consultați [Secțiunea 4.2.5.1.](#)

O**ogle**

Player DVD care citește meniurile DVD, consultați [Secțiunea 11.3.](#)

P**passwd**

Schimbă parola, consultați [Secțiunea 2.2](#) și [Secțiunea 4.1.6.](#)

pccardctl

Administrează cardurile PCMCIA, consultați [Secțiunea 10.2.3.3.](#)

pdf2ps

Transformă fișierele PDF în PostScript, consultați [Secțiunea 8.1.2.](#)

perl

Practical Extraction and Report Language.

pg

Paginează ieșirea text, consultați [Secțiunea 3.3.4.2.](#)

pgerp

Caută procese pe criterii bazate pe nume și alte atribute, consultați [Secțiunea 4.1.4.](#)

ping

Trimite o cerere de răspuns unui alt calculator, consultați [Secțiunea 10.2.6.2.](#)

play

Redă o moștră de sunet, consultați [Secțiunea 11.2.3.](#)

pr

Convertește fișierele text în vedeera tipăririi.

printenv

Afișează, în parte sau în totalitate, variabilele de mediu, consultați [Secțiunea 7.2.1.](#)

procmail

Procesor independent de mesaje de poștă electronică, consultați [Secțiunea 10.3.2.3.](#)

ps

Afișează statutul proceselor, consultați [Secțiunea 4.1.4](#) și [Secțiunea 4.3.5.4.](#)

pstree

Afișează procesele în formă arborescentă, consultați [Secțiunea 4.1.4.](#)

pwd

Afișează directorul de lucru curent, consultați [Secțiunea 2.2.](#)

Q**quota**

Afișează gradul de folosire a discului și limitările, consultați [Secțiunea 3.2.3.3.](#)

R**rcp**

Copiere la distanță (nesecurizată!).

rdesktop

Client Remote Desktop Protocol, consultați [Secțiunea 10.4.6.](#)

reboot

Oprește sistemul, consultați [Secțiunea 4.2.6.](#)

recode

Convertește fișierele într-un alt set de caractere, consultați [Secțiunea 7.4.4.](#)

renice

Modifică prioritatea unui proces, consultați [Secțiunea 4.3.5.1.](#)

restore

Restaurează copiile de siguranță create cu **dump**, consultați [Secțiunea 9.2.5.](#)

rlogin	Autentificare la distanță (telnet, nesecurizată!), consultați Secțiunea 10.4.2 și Secțiunea 10.5.2 .
rm	Șterge un fișier, consultați Secțiunea 3.3.2 .
rmdir	Șterge un director, consultați Secțiunea 3.3.2.2 .
roff	Sistem de formatare, consultați Secțiunea 8.1.2 .
rpm	Administrator de pachete RPM, consultați Secțiunea 7.5.2.1 .
rsh	Consolă la distanță (nesecurizată!), consultați Secțiunea 10.4.2 .
rsync	Sincronizează două directoare, consultați Secțiunea 9.3 .

S

scp	Copiere la distanță securizată, consultați Secțiunea 10.4.4.1 .
screen	Administrator de ecran cu emulator pentru VT100, consultați Secțiunea 4.1.2.1 .
set	Afișează, introduce sau modifică o variabilă.
setterm	Configurează atributele terminalului.
sftp	ftp securizat (criptat), consultați Secțiunea 10.4.4.1 .
sh	Deschide o consolă standard, consultați Secțiunea 3.2.3.2 .
shutdown	Oprește sistemul, consultați Secțiunea 4.2.6 .
sleep	Așteaptă o perioadă stabilită, consultați Secțiunea 4.4.1 .
slocate	Versiune îmbunătățită din punct de vedere al securității pentru GNU Locate, consultați Secțiunea 3.3.3.3 .
slrn	Client în modul text pentru Usenet, consultați Secțiunea 10.2.6 .
snort	Unealtă pentru detectarea intruziunilor în rețea.
sort	Sortează linii din fișierele text, consultați Secțiunea 5.3.2 .
spell	Verifică ortografia, consultați Secțiunea 5.1.2.3 .
ssh	Consolă securizată, consultați Secțiunea 10.4.4.1 .
ssh-keygen	Generează chei de autentificare și le administreză, consultați Secțiunea 10.4.4.5 .
stty	Modifică și afișează configurațiile terminalului.
su	Schimbă utilizatorul, consultați Secțiunea 3.2.1 , Secțiunea 7.5.3.2 și Secțiunea 10.4.6 .

T

tac	Conexează și afișează fișierele în ordine inversă, consultați cat .
tail	Afișează ultima parte a conținutului unui fișier, consultați Secțiunea 3.3.4.3 .
talk	Conversează cu un utilizator.
tar	Utilitar pentru arhivare, consultați Secțiunea 9.1.1.1 .
tcsh	Deschide o consolă Turbo C, consultați Secțiunea 3.2.3.2 .

telinit	Controlează inițializarea proceselor, consultați Sectiunea 4.2.5 .
telnet	Interfață cu utilizatorul pentru protocolul TELNET (nesecurizat!), consultați Sectiunea 10.4.2 .
tex	Formatează fișiere text, consultați Sectiunea 8.1.2 .
time	Cronometrează o comandă sau oferă informații despre resursele folosite, consultați Sectiunea 4.3.2 .
tin	Program pentru citirea știrilor, consultați Sectiunea 10.2.6 .
top	Afișează procesele care solicită cel mai mult procesorul, consultați Sectiunea 4.1.4 , Sectiunea 4.3.5.3 și Sectiunea 4.3.5.4 .
touch	Modifică eticheta de timp a fișierelor, consultați Sectiunea 7.1.2 .
traceroute	Afișează ruta urmată de pachete în rețea, consultați Sectiunea 10.2.6.3 .
tripwire	Verifică integritatea fișierelor unui sistem UNIX, consultați Sectiunea 10.4.5 .
troff	Formatează documente, consultați Sectiunea 8.1.2 .
tvttime	Aplicație de bună calitate pentru televiziune.
twm	Tab Window Manager pentru sistemul X Window.

U

ulimit	Controlează resursele, consultați Sectiunea 7.1.2.5 .
umask	Configurează masca fișierului în funcție de utilizator, consultați Sectiunea 3.4.2.2 .
umount	Demontează un sistem de fișiere.
uncompress	Despacetează fișierele comprimate.
uniq	Șterge liniile duplicate dintr-un fișier sortat, consultați Sectiunea 5.3.2 .
up2date	Actualizează pachete RPM.
update	Serviciu al kernei, care trimită înapoi pe disc memoria tampon nefolositoare.
update-rc.d	Configurează scripturile init, consultați Sectiunea 4.2.5.1 .
uptime	Afișează timpul de lucru al sistemului și solicitarea medie a sistemului, consultați Sectiunea 4.1.4 și Sectiunea 4.3.5.2 .
urpmi	Actualizează pachetele RPM, consultați Sectiunea 7.5.3.3 .
userdel	Șterge contul unui utilizator și fișierele conexe.

V

vi(m)	Pornește editorul vi (îmbunătățit), consultați Sectiunea 6.1.2.2 .
vimtutor	Tutorialul Vim.
vmstat	Afișează statistică legată de memoria virtuală, consultați Sectiunea 4.3.5.4 .

W

w

Arată cine este autentificat și ce face.

wall

Trimit un mesaj către terminalele fiecăruia, consultați [Secțiunea 4.2.6](#).

wc

Afișează numărul bițiilor, al cuvintelor și liniilor dintr-un fișier, consultați [Secțiunea 3.2.1](#).

which

Arată traseul complet al comenzi (consolei), consultați [Secțiunea 3.2.1](#) și [Secțiunea 3.3.3.2](#).

who

Arată cine este autentificat, consultați [Secțiunea 4.1.6](#).

who am i

Afișează ID-ul utilizatorului curent.

whois

Interoghează o bază de date whois, consultați [Secțiunea 10.2.6.1](#).

write

Trimit un mesaj altui utilizator, consultați [Secțiunea 4.1.6](#).

X

xargs

Introduce și execută comenzi de la dispozitivul standard de intrare, consultați [Secțiunea 3.3.3.3](#).

xauth

Utilitar X authority.

xawtv

Un program X11 pentru televiziune.

xcdroast

Interfață grafică pentru cdrecord, consultați [Secțiunea 9.2.2](#).

xclock

Ceas digital/analog pentru X.

xconsole

Monitorizează mesajele de sistem în X.

xdm

Administrator de ecran cu suport pentru XDMCP, host chooser, consultați [Secțiunea 4.2.4](#) și [Secțiunea 7.3.2](#).

xdvi

Vizualizator DVI, consultați [Secțiunea 8.1.2.2](#).

xedit

Editor grafic pentru X Window, consultați [Secțiunea 6.3.3.3](#).

xfs

Serverul de fonturi X.

xhost

Controlează accesul programelor în X, consultați [Secțiunea 10.4.3.2](#).

xine

Player video liber, consultați [Secțiunea 11.3](#).

xinetd

Serviciu extins pentru Internet, consultați [Secțiunea 10.3.1.2](#).

xload

Afișează solicitarea medie a sistemului în X, consultați [Secțiunea 4.3.5.6](#).

xlsfonts

Listează fonturile în X.

xmms

Player audio pentru X, consultați [Secțiunea 11.2.2.1](#).

xpdf

Vizualizator PDF, consultați [Secțiunea 8.1.2.2](#).

xterm

Emulator de terminal pentru X.

Y

yast

Unealtă pentru administrarea sistemului în Novell SuSE Linux.

yum

Actualizează pachete RPM, consultați [Secțiunea 7.5.3](#).

Z

zapping

Program pentru televiziune în mediul Gnome.

zcat

Comprimă sau despachetează fișiere.

zgrep

Caută în fișierele posibil comprimate o expresie regulată.

zmore

Filtru pentru afișarea textului comprimat.
