

ATENŢIONARE!

Conținutul acestei platforme de instruire a fost elaborat în cadrul proiectului "Dezvoltarea resurselor umane în educație pentru administrarea rețelelor de calculatoare din școlile românești prin dezvoltarea și susținerea de programe care să sprijine noi profesii în educație, în contextul procesului de reconversie a profesorilor și atingerea masei critice de stabilizare a acestora în școli, precum și orientarea lor către domenii cerute pe piața muncii". Conținutul platformei este destinat în exclusivitate pentru activități de instruire a membrilor grupului țintă eligibil în proiect.

Utilizarea conținutului în scopuri comerciale sau de către persoane neautorizate nu este permisă.

Copierea, totală sau parțială, a conținutului de instruire al acestei platforme de către utilizatori autorizați este permisă numai cu indicarea sursei de preluare (platforma de instruire eadmin.cpi.ro).

Pentru orice probleme, nelămuriri, sugestii, informații legate de aspectele de mai sus vă rugăm să utilizați adresa de email: proiect.eadmin@cpi.ro

Acest material a fost elaborat de Cristian Oftez, în cadrul S.C. Centrul de Pregătire în Informatică S.A., partener de implementare a proiectului POSDRU /3/1.3/S/5.

Versiunea materialui de instruire: V2.0

8. Utilizatori și accesul la resurse

Utilizatori şi Grupuri de utilizatori

Linux este un sistem de operare multi-user şi multi-tasking, drept urmare trebuie să avem mecanisme care să ofere control deplin asupra accesului în sistem. Ca şi în alte sisteme de operare, în Linux beneficiem de conturi de utilizatori, prin care ne putem desfăşura activitățile. Fiecare utilizator are o parolă şi face parte din cel puțin un grup de utilizatori – un grup de utilizatori este un set de utilizatori, care ar putea avea ceva în comun, şi sunt grupați astfel pentru o mai bună organizare: putem grupa de exemplu toți studenții unei facultăți în grupul **studenți** şi toți profesorii în grupul **profesori**.

Motivul grupării utilizatorilor este posibilitatea de a stabili accesul la resurse ale sistemului la un nivel mai general decât cel de utilizator dar și acest lucru este posibil, dat fiind un număr redus de utilizatori). Sistemul de permisiuni stabilește clar cine și cum poate accesa fișierele sistemului. Ca mențiune, putem afla numele utilizatorului curent cu ajutorul comenzii "**whoami**" și grupurile din care face parte prin comanda "**groups**".

Fiecare utilizator din sistem va avea de asemenea un director privatat în directorul /home, unde de regulă va păstra fişierele personale. De asemenea, acest director va avea permisiunile setate în așa fel încât să fie accesibil doar proprietarului. Excepție la această regulă este superuser-ul "root", care practic poate face orice într-un sistem Linux.

Adăugarea de noi uşeri şi grupuri se poate face doar de către "root", neexistând însă o limită de utilizatori sau grupuri.

Pentru administrarea conturilor și grupurilor de utilizatori, avem o serie de opțiuni, de la folosirea utilitarelor grafice – cea mai simplă metodă de altfel – până la folosirea unei game mari de comenzi ce vor fi descrise în continuare.

Crearea conturilor de utilizator

Pentru crearea unui nou utilizator se va folosi comanda "**useradd**", cu sintaxa următoare: "useradd [opțiuni] <nume_utilizator>". Opțiunile cele mai întâlnite sunt:

- c comentariu
- d directorul "home" al utilizatorului. Implicit este /home/<nume_utilizator>
- e data de expirare a contului, în format AAAA-LL-ZZ (an, lună, zi)
- g numele sau numărul grupului primar al uşerului
- G Lista cu grupuri adiționale în care user-ul nou creat va fi inclus
- D folosit ca atare, va afişa setările implicite, iar folosit în conjuncție cu alte opțiuni, schimbă setările implicite pentru opțiunile respective. De exemplu, putem schimba "shell"-ul folosit implicit din "bash" în "sh" prin comanda "useradd –D –s /bin/sh"
- m această opțiune va avea ca efect crearea directorului home al utilizatorului dacă acesta nu există.

EXEMPLU: Comanda "**useradd** –**m user2**" va crea un cont de utilizator nou, numit "user2".

Crearea unei noi parole pentru un cont de utilizator

Pentru ca orice cont de utilizator să fie funcțional, acesta trebuie neapărat să aibă asociată o parolă. Comanda pentru crearea sau schimbarea unui cont este "**passwd**". Aceasta poate fi folosită fără nici un parametru, când va încerca schimbarea parolei contului logat în momentul respectiv, sau poate avea ca parametru numele contului pentru care se încearcă atribuirea unei noi parole. Este important de amintit, ca un cont nu este activ și deci nu poate fi folosit dacă nu are atribuită o parolă.

EXEMPLU: Comanda "**passwd user2**" va seta parola pentru contul creat la pasul anterior.

Grupuri de utilizatori

După cum am amintit mai devreme, conturile fac parte din cel puțin un grup de utilizatori; pentru control asupra acestora, putem folosi comenzile:

- groupadd crează un nou grup (EXEMPLU: "groupadd test")
- groupmod modifică un grup. (EXEMPLU: "groupmod –n <nume_nou> <nume_vechi>" – schimbă numele unui grup de utilizatori deja existent.

Ştergerea conturilor de utilizator și a grupelor de utilizatori

În cazul în care dorim să ștergem fie un cont fie un grup de utilizatori, putem folosi următoarele comenzi:

- userdel [-r] <cont_utilizator> pentru ştergerea contului specificat. Parametrul opțional "–r" menționează că va fi şters şi directorul home al respectivului utilizator.
- groupdel <nume_grup> această comandă şterge grupa specificată din lista de grupuri.

Comenzi adiționale

- usermod modifică detaliile conturilor deja existente. Putem de exemplu, folosind comanda "usermod –L user2" ("-L" de la Lock), să blocăm accesul la un cont. Pentru a debloca un cont, trebuie folosit parametrul "–U" (Unlock).
- adduser, addgroup, deluser, delgroup reprezintă variantele mai uşor de folosit ale comenzilor explicate mai devreme. Diferența constă în faptul că acestea sunt mai "prietenoase" cu utilizatorul, în sensul că în locul parametrilor adiționali, vă vor întreba într-un mod interactiv, detaliile necesare.
- gpasswd este un utilitar folosit pentru administrarea grupurilor de utilizatori. Parametrii acceptați de acest utilitar sunt următorii: -a <user> <grup> - adaugă utilizatorul în grupă -d <user> <grup> - elimină utilizatorul din grupă

Important de menționat este faptul că informațiile despre utilizatori și grupe sunt menținute în două fișiere importante: "/etc/passwd" pentru utilizator și "/etc/group". Aceste fișiere au un format strict, foarte bine definit: fiecare linie reprezintă o intrare, și are mai multe câmpuri separate prin caracterul ":".

Pentru fişierul /etc/passwd formatul este:

"nume:parolă:id_utilizator:id_grup:comentariu:director_home:shell" - **SFAT:** pentru a viziualiza conținutul fişierului /etc/passwd puteți folosi comanda "less /etc/passwd". Ca mențiune, în majoritatea cazurilor, câmpul parolă este completat cu un singur caracter "x", ceea ce simbolizează faptul că parola este criptată și stocată în fișierul /etc/shadow.

	user1@statie1: ~	
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s	<u>H</u> elp	
File Edit View Terminal Tabs root:x:0:0:root:/root:/bin/b daemon:x:1:1:daemon:/usr/sbi bin:x:2:2:bin:/bin:/bin/sh sys:x:3:3:sys:/dev:/bin/sh sync:x:4:65534:sync:/bin:/bi games:x:5:60:games:/usr/game man:x:6:12:man:/var/cache/ma lp:x:7:1p:/var/spool/lpd:/ mail:x:8:8:mail:/var/mail:/b news:x:9:9:news:/var/spool/p uucp:x:10:10:uucp:/var/spool/p proxy:x:13:13:proxy:/bin:/bi www-data:x:33:33:www-data:/v backup:x:34:34:backup:/var/b list:x:38:38:Mailing List Ma irc:x:39:39:ircd:/var/run/ir gnats:x:41:41:Gnats Bug-Repo nobody:x:65534:65534:nobody: libuuid:x:100:101::/var/lib/ Debian-exim:x:101:105::/var/ statd:x:104:109:Avahi mDNS c gdm:x:105:112:Gnome Display haldaemon:x:106:114:Hardware hplip:x:107:7:HPLIP system u user1:x:1000:1000:Ion Popeso	<pre>Help ash n:/bin/sh n/sync s:/bin/sh n:/bin/sh bin/sh ews:/bin/sh /uucp:/bin/sh n/sh ar/www:/bin/sh ackups:/bin/sh nager:/var/list:/bin/sh cd:/bin/sh rting System (admin):/var/lib/gnats:/bin/sh libuuid:/bin/sh spool/exim4:/bin/false nfs:/bin/false un/dbus:/bin/false aemon,,:/var/run/avahi-daemon:/bin/false abstraction layer,,:/var/run/hal:/bin/false ser,,:/var/run/hplip:/bin/false u,,::/home/user1:/bin/bash</pre>	
🖬 💷 user1@statie1: ~		

Pentru fişierul /etc/group formatul este:

"nume_grup:parolă:id_grup:user1,user2,..." – este similar cu fişierul /etc/passwd, dar conține informații despre grupuri. Ultimul câmp al fiecărei intrări reprezintă lista toturor utilizatorilor din grupul respectiv, despărțiți prin virgule.

Se pot trage o serie de concluzii:

- atât utilizatorii cât şi grupurile au un identificator numeric numit "id", folosit pentru reprezentare.
- în momentul în care este creat un nou cont de utilizator, se crează şi o grupă cu acelaşi nume, şi acelaşi id, din care acesta va face parte.

- grupurile pot avea şi ele parole, permiţând existenţa grupurilor "privilegiate"
- putem adăuga şi şterge utilizatori din grupe, exclusiv prin editarea fişierelor descrise mai sus. Acest procedeu nu este recomandat însă, şi este preferabilă folosirea utilitarelor descrise mai devreme.
- Putem restrictiona accesul anumitor utilizatori prin comanda usermod

Putem afla oricând care utilizatori sunt logați folosind comanda "who", care ne va specifica și pe ce consolă sunt aceștia logați.

Administrarea folosind interfață grafică

Administrarea în mod grafic presupune folosirea instrumentelor universal valabile, distribuite cu mediul grafic folosit, în cazul de față, GNOME - un mediu desktop foarte des întâlnit. Aceste instrumente se numesc GNOME System Tools (GST), și reprezintă un set integrat de utilitare ce pot fi folosite în configurarea diverselor aspecte ale sistemului de operare printr-o interfață grafică, foarte uşor de folosit. GNOME System tools se bazează pe un set de module numit "System Tools Backends". Acesta are rolul de a oferi o interfață comună (numită Dbus), independentă de platformă (fie Linux, FreeBSD sau alte sisteme UNIX), prin care se pot citi și modifica diverse setări de sistem. Până la apariția acestui set de module, era practic imposibil să se creeze utilitare care să funcționeze perfect indiferent de platformă, iar în prezent sunt suportate o mulțime de distribuții: Redhat, Mandrake, Debian (și variante precum Ubuntu), Suse, Fedora.

Utilitarul folosit pentru administrarea userilor și grupurilor se numește "usersadmin". Această aplicație trebuie lansată cu privelegii de "root", altfel nu va putea avea acces la diferitele fișiere de system ce trebuiesc modificate. Utilitarul se bazează, după cum am menționat mai devreme, pe existența mediului grafic, iar fără acesta nu va putea rula. Pentru lansarea sa, există mai multe posibilități:

- Din panoul (implicit superior) prin intermediul meniului "System" -> "Administration" -> "Users and Groups".
- Dintr-un terminal virtual, ce rulează în mediul grafic. Această opțiune presupune de asemenea şi folosirea unor utilitare adiționale pentru a vă autentifica drept root (precum su sau sudo, pe care le vom analiza mai târziu). De altfel "shortcut"-urile folosite drept prima alternativă, se bazează pe un astfel de utilitar, numit "gksu". Veți observa de altfel că sunteți rugați să vă autentificați drept "root" înainte ca orice utilitar de configurare să pornească".

Depinzând de distribuția sau chiar de versiunea distribuției pe care o rulați, puteți fi atenționat că GNOME System Tools nu este oficial suportat pentru distribuția dvs. Acest mesaj apare și în cazul versiunii 5.0.1 a Debian-ului, și este datorat unei schimbări neanunțate în cadrul distribuției. Utilizatorul poate trece peste această atenționare în cazul de față, fără teamă că utilitarul în cauză nu va funcționa corect.

👸 Applicat	tions Places System	າ 🔊 🗟			🕼 🐵 Fri Apr 17, 6:20 PM	
		user1@statie1: ~				
<u>File</u> <u>E</u> dit	<u>V</u> iew <u>T</u> erminal Ta	<u>b</u> s <u>H</u> elp			14	
userl@sta	atiel:~\$ gksu user	s-admin			Î	
	2	Users	Settings		\mathbf{X}	
	Name		Login name	Home directory	- A dd User	
	Son Popescu		user1 root	/home/user1 /root	Properties	
					<u> Delete</u>	
					Anage Groups	
	() <u> Help</u>				X Close	
			1			
🖾 💷 usei	r1@statie1: ~	🛽 Users Settings				

În imaginea de mai sus se poate observa lista cu utilizatorii existenți, cât și diverse informații despre aceștia. Interfața utilitarului este una simplă, și permite adăugare sau ștergerea de grupuri și utilizatori cu uşurință.

Pentru a adăuga un nou utilizator, trebuie apăsat butonul "Add user", care va deschide o nouă fereastră în care vor fi introduse diverse informații despre noul utilizator. Fereastra va avea 3 panouri: "Account", "User privileges" și "Advanced".

Primul panou, conține informațiile generale, precum nume, adresă, telefon și o zonă în care se poate alega parola pentru noul utilizator: manual, sau generată automat. Cel de-al doilea panou specifică drepturile generale ale utilizatorului vizat, iar ultimul panou ne oferă posibilitatea de a specifica setări neconvenționale pentru utilizatorul în cauză, de la alegerea unui director home diferit, până la setarea unui user id manual și chiar selectarea shellului pe care utilizatorul îl va folosi. În mod similar se pot modifica sau șterge utilizatorii selectați prin butoanele "Properties" și "Delete".

Butonul "Manage Groups" este folosit pentru administrarea grupurilor de utilizatori: se pot adăuga noi grupuri de utilizatori sau şterge şi modifica grupurile deja existente. Pentru a modifica un grup, acesta trebuie selectat din lista afişată, urmat de acționarea butonului "Properties".

👸 Applicat	ions Places S	System 👰 👩		🏟 🖗 🛛 🖓 Fri Apr 17,	6:32 PM 📃 💻
		user1@sta	itie1: ~		
<u>File</u> <u>E</u> dit	<u>V</u> iew <u>T</u> ermin	al Ta <u>b</u> s <u>H</u> elp			K
userl@sta	tiel:~\$ gksu	users-admin			
a	2		Users Settings		
	Name	49	Login name Home directory Groups settings	/ 🕹 Add User	
	⊘ root	libuuid			
		crontab			Add Group
		scanner	Group 'user1' Properties		<u>Properties</u>
		ssl-cert	Basic Settings		😭 Delete
		Debian-exim			
		mlocate	Group <u>n</u> ame: user1		
		ssh	Group <u>I</u> D: <u>1000</u>	~	
		messagebus avabi	Group Members		
		netdev	🗆 Ion Popescu		
		Ipadmin	🗆 root	=	
		gdm			
		stb-admin	K III		
		haldaemon	🛛 🔀 <u>C</u> ancel	<u>«"о</u> к	
		powerdev			
ſ	11 Help	user1			
	W Holp	telp 🔁			X Close
🖾 💷 user	1@statie1: ~	🕒 Users Settir	ngs		

Din fereastra nou creată, se poate modifica lista de utilizatori a grupului (prin bifarea, respectiv debifarea utilizatorilor din lista afişată). Putem modifica identificatorul grupului de utilizatori cât și adăuga sau scoate utilizatori în grup, într-o manieră mult mai uşoară.



Sistemul de permisiuni

Toate sistemele de operare Linux / Unix au fost gândite din start drept sisteme multi-user, multi-tasking, capabile să găzduiască mai mulți utilizatori. Din acest motiv, a fost necesar un sistem de securitate, bazat pe permisiuni cu un nivel de granularitate mic, pentru a restricționa accesul (atât la scriere cât și la citire) a anumitor resurse. Fiind un sistem multi-user adevărat, deseori folosit pentru servere, Linux are un sistem de permisiuni încorporat, foarte bine pus la punct. Acest fapt poate cauza probleme pentru utilizatorii noi (mai ales celor obișnuiți cu sistemul implicit "access-all" folosit în Windows), astfel multe erori pot fi cauzate datorită permisiunilor setate incorect. Sistemul de permisiuni folosit de Linux este total independent de distribuție, și nu se schimbă cu aceasta.

Fiecare fişier are ataşat un set de permisuni simple, care dictează accesul la resursa respectivă. Permisiunile asupra fişierelor sunt definite pentru 3 mari grupuri: proprietar, grupă și alți utilizatori. Astfel, se pot defini un set de reguli individuale pentru fiecare din grupurile menționate mai sus, bazate pe un sistem simplu de acces. Drepturile de acces la fişiere se împart și ele în trei categorii: "r" – read (citire), "w" – write (scriere), "x" – execute (execuție); acestea pot fi folosite în orice combinație, pentru definirea oricărui nivel de acces.

Combinații de cele 3 drepturi menționate mai sus trebuiesc specificate pentru fiecare grupă în parte, deci vom avea 3 x 3 specificatori de acces pentru fiecare fişier, prin care vom descrie accesul fiecărei grupe (în ordinea amintită). Menționăm de asemenea, că, pentru ca acest sistem de permisiuni să funcționeze, trebuie să existe o referință asupra identității proprietarului și al grupului – în esență, fiecare fișier va avea definit un utilizator proprietar, un grup proprietar, cât și 9 permisiuni individuale. Pentru a vedea permisiunile unor resurse dintr-un sistem Linux, putem folosi comanda "Is –I".

```
statie1:/home# ls -l
total 12
-rw-r--r-- 1 root root 24 2009-04-17 19:54 informatii
drwxr-xr-x 23 user1 user1 4096 2009-04-17 19:53 user1
drwxr-xr-x 2 1001 1001 4096 2009-04-17 19:39 user2
statie1:/home#
```

Rezultatul afişat poate fi interpretat astfel: directorul curent are 3 fişiere: "informații", "user1" și "user2". Fiecare linie conține informații despre fișierul în cauză: primul caracter semnifică tipul de fișier, secvența de 9 caractere reprezintă permisiunile pentru fiecare grup în parte, următorul număr reprezintă numărul de legături al directorului – implicit, un director nou creat va avea 2. În continuare avem proprietarul, respectiv grupul proprietar al resursei (Observație – în cazul fișierului "user2", nu avem menționate niște nume, ci identificatorul utilizatorului, respectiv grupei), dimensiunea în bytes, data și ora la care fișierul a fost creat, iar în final, numele fișierului.

Pentru a putea înțelege mai bine cum funcționează sistemul de permisiuni, putem să analizăm în detaliu linia pentru fișierul "user1": spre deosebire de

primul fișier, primul caracter este "d" (în loc de "-"), cu semnificația că "user1" este de fapt un director. Ca proprietari, acesta are 2: utilizatorul "user1", și grupul de utilizatori "user1".

În continuare dacă examinăm secvența de 9 caractere pentru directorul "user1" vom vedea că acesta este "rwxr-xr-x", iar dacă împărțim această secvență în grupuri de câte 3 caractere vom obține permisiunile asupra acestui fișier ale fiecărei grupe. Astfel deducem că utilizatorul "user1" (**owner**) are drepturi "rwx", grupul de utilizatori "user1" (**group**) are drepturi "r-x", iar toți ceilalți (**others**), au drepturi "r-x".

Pentru orice fişier, permisiunile au următoarele efecte:

- permisiunea read (r) oferă acces la citirea fişierului; pentru directoare, oferă acces la lista sa de fişiere
- permisiunea write (w) permite modificările în fişierul vizat; pentru directoare, oferă posibilitatea de a şterge sau a crea fişiere în director.
- execute (x) permite execuția fişierului drept o comandă. În cazul directoarelor, această permisiune oferă acces în director, şi trebuie setat pentru a putea accesa conținutul acestuia.

În cazul directoarelor, permisiunile pot crea confuzii mari începătorilor:

- "r--" user-ul, grupul sau others pot lista doar conținutul directorului. Nu se poate face nici o schimbare asupra sa.
- "rw-" permisiunea de write nu are nici un efect în cazul directoarelor în absența permisiunii de execuție (x), astfel, acest mod rezultă cu un comportament identic cu cel din cazul precedent.
- "r-x" permite listare, precum şi accesul la fişierele din cadrul directorului, dar nu oferă posibilitatea modificării acestora. În acest caz, prin acces, înţelegem că putem vizualiza, modifica sau executa fişierele din cadrul directorului exclusiv pe baza permisiunii acestora.
- "--x" fişierele din director pot fi accesate, dar conţinutul directorului nu poate fi listat, astfel încât, trebuie să cunostem în prealabil numele fişierelor conţinute.
- "rwx" avem acces total asupra fişierelor din director, atâta timp cât permisiunile individuale ale fişierelor permit acest lucru.

Observație: Utilizatorul "root" nu este afectat în nici un fel de către permisiunile impuse asupra unui fișier, putând astfel să accese, modifice sau execute orice fișier. Acest fapt, spre deosebire de sistemele de operare Microsoft Windows, conferă user-ului "root" putere deplină în sistem, ceea ce poate constitui un mare avantaj sau un mare pericol de securitate.

Sistemul de permisiuni numerice (sau octal)

Deşi sistemul de permisiuni se referă implicit la același lucru, există mai multe feluri de reprezentare. Printre acestea se numără sistemul octal de permisiuni, care folosit corespunzător de către un utilizator experimentat, poate eficientiza procesul de modificare al permisiunilor într-un sistem Linux. Sistemul octal este unul simplu și presupune folosirea a trei cifre ce pot lua valori între 0 și 7 (8 valori distincte, de unde vine de altfel și denumirea

reprezentării), fiecare cifră combinând permisiunile individuale pentru cele 3 grupuri menționate mai devreme, respectiv drepturile de acces pentru **owner, group** și **others** (în această ordine). Din moment ce o singură cifră este folosită pentru a reprezenta trei atribute diferite, trebuie folosit un sistem de separare: permisiunile sunt reprezentate prin suma puterilor lui 2.

Astfel valorile 1 (execuție), 2 (scriere) și 4 (citire) pot fi combinate în orice fel astfel încât permisiunile particulare să fie identificabile (de exemplu, dacă dorim permisiune de scriere și citire dar nu de execuție, adunăm valorile 2 cu 4, iar pentru citire și execuție am aduna 1 cu 4). Sistemul devine mult mai clar în momentul în care se folosește o reprezentare binară a fiecărui set de permisiuni:

Cifră	Echivalent text	Binară
0		000
1	X	001
2	-W-	010
3	-wx	011
4	r	100
5	r-x	101
6	rw-	110
7	Rwx	111

Din tabelul anterior, se poate observa clar, că distribuția celor 3 biți ai fiecărei cifre (fiecare reprezentând o putere a lui 2) este de așa natură încât se pot specifica permisiunile individuale. Astfel primul bit reprezintă permisiunea de citire (activată doar pentru valori de 1), al doilea bit reprezintă permisiunea de scriere, iar ultimul bit controlează permisiunea de execuție.

Cazuri particulare de aplicare ale sistemului de permisiuni

Pe lângă scopul sistemului de permisiuni specific unui sistem de operare precum Linux, în care acestea sunt folosite exclusiv pentru restricții la sistemul de fișiere local (din motive de securitate), există și cazuri speciale.

Serverele web atribuie în mod normal permisiunile uşerului ce reprezintă acest serviciu clientului web care se conectează la serverul în cauză. În acest context, user-ul ce reprezintă serviciul web nu va fi de regulă în acelaşi grup cu nici un alt user de pe sistem, deci i se vor aplica restricțiile corespunzătoare grupului "others".

Ca urmare imediată, fișierele oferite prin protocolul **HTTP** – fișiere html, imagini, etc. - necesită de regulă, permisiuni 644, din moment ce user-ul în cauză are nevoie de acces de citire la fișier. În cazul fișierelor de tip **script**

permisiunile standard sunt 755, din moment ce grupurile de permisiuni "others" și "group" au nevoie și de drepturi de execuție.

Fişierele de pe un server **FTP** devin proprii utilizatorului care le încarcă. Acestea primesc implicit nivelul 644, dar în funcție de setările serverului FTP, acestea pot lua alte valori.

Schimbarea proprietarului sau grupului unui fişier

După cele menționate anterior, este evident faptul că orice fișier sau director din sistemul de fișiere Linux va avea un user și un grup proprietar, ceea ce permite definirea unor permisiuni particularizate asupra resurselor disponibile.

În mod implicit, creatorul unui fișier sau director va fi și proprietarul acestuia, iar grupul proprietar va fi grupul din care face parte utilizatorul respectiv, însă aceste proprietăți se pot schimba (numai de către utilizatorul "root").

Pentru schimbarea proprietarului unui fişier se foloseşte comanda chown.
 Sintaxa comenzii este următoarea:

chown [-R] [user][:][group] tinta1 [...]

Primul parametru specifică o schimbare de proprietar recursivă în cazul în care ținta este un director. Cel de-al doilea parametru este folosit pentru alegerea utilizatorului, care va fi proprietarul țintei (însă poate fi folosit și pentru atribuirea grupului proprietar). După acești parametrii necesari, urmează cel puțin un fișier sau director ținta, însă pot fi alese oricâte ținte (separate prin spații).

 Pentru schimbarea grupului proprietar asociat unei resurse se mai poate folosi şi comanda chgrp (prescurtare de la "change group"). Deşi acest rol poate fi îndeplinit cu succes prin comanda chown, aceasta este rezervată uşerului "root" spre deosebire de chgrp, care poate fi folosit de către orice utilizator (cu limitarea schimbării grupului într-unul din care utilizatorul face parte). Sintaxa comenzii este următoarea:

chgrp grup tinta1 [...]

Primul parametru specifică grupul cu care resursa țintă va fi asociată, al doilea parametru reprezintă resursa țintă, putând urma oricâți alți parametrii ce specifică fișiere sau directoare adiționale pentru care este dorită schimbarea respectivă.

Schimbarea permisiunilor

Permisiunile asociate unui fișier sau director într-un sistem Linux, pot fi schimbate folosind comanda **chmod**. În contextul interfeței grafice, acestea se pot schimba de regulă prin intermediul "**File Manager**"-ului folosit – de exemplu, în cazul în care se folosește **Nautilus**, care este folosit implicit de către **GNOME**, permisiunile se schimbă folosind tab-ul "Permissions" al ferestrei de proprietăți asociate elementului vizat.

Sintaxa comenzii **chmod** este următoarea:

computer1:/# chmod [opțiuni] permisiuni țintă

Opțiunile pot fi:

- -R, permisiunile vor fi setate recursiv
- -f, mod forțat
- -v, mod "verbose" (cu afişare pentru fiecare fişier procesat)
- -c, se afişează informații numai în cazul în care sunt efectuate schimbări asupra fişierului țintă

Permisiunile pot fi:

- Permisiuni în format octal
- Permisiuni în format text

În cazul permisiunilor text, se folosește următoarea sintaxă:

[grup_permisiuni][operație][permisiuni]

- grup_permisiuni reprezintă grupul de permisiuni asupra căruia se va efectua modificarea şi poate lua următoarele valori:
 - "u" pentru grupul de permisiuni "owner"
 - "g" pentru grupul de permisiuni "group"
 - "o" pentru grupul de permisiuni "others"
 - "a" pentru toate grupurile de mai sus
- operație reprezintă modul în care va fi efectuată schimbarea:
 - "=" atribuie în mod direct
 - "-" înlătură permisiunile specificate
 - "+" adaugă permisiunile specificate
- permisiuni reprezintă drepturile efective ce urmează a fi schimbate:
 - "r" permisiune de citire
 - "w" permisiune de scriere
 - "x" permisiune de execuție
 - "t" setează bitul "sticky"
 - "s" setează SUID sau SGID

Exemplu:

- Comanda "chmod ug+rw,o+r fişier" va seta permisiunile de citire şi de scriere pentru grupurile "owner" şi "group" şi permisiunea de citire pentru grupul "others".
- Comanda "chmod 664 fişier" va avea acelaşi efect ca şi anterioară. Se atribuie permisiunea de 6 (2+4, adică scriere şi citire) grupurilor
 owner" și group" și permisiunea de 4 (doar citire) grupului others"

"owner" și "group" și permisiunea de 4 (doar citire) grupului "others". Tipuri de fișiere

Aşa cum am observat mai devreme, primul caracter din şirul de permisiuni simbolizează tipul de element cu care avem de-a face. Astfel "d" însemna că este vorba de un director, pe când "-" reprezintă un fişier. Pe lângă acestea există și alte tipuri de fişiere:

- **d** director
- fişier
- I legătură
- s socket UNIX
- p pipe
- **c și b** fișiere dispozitiv

Sistemul de fişiere

Ca orice sistem de operare, Linux are nevoie de un sistem de fişiere (numit **File System**) pentru a putea gestiona informațiile și datele stocate în sistem. Un sistem de fişiere – în termeni științifici – nu reprezintă nimic altceva decât o metodă de organizare și stocare a datelor într-un calculator. Sistemele de fișiere pot fi folosite pentru dispozitive propriu-zise, precum hard discuri sau unități CD-ROM, sau pot facilita accesul la resurse disponibile într-o locație diferită printr-un mediu de comunicare cum ar fi o rețea de calculatoare (protocoale de genul NFS sau SMB). Desigur, conform ultimelor tendințe în materie de organizare a sistemelor informatice, există și sisteme de fișiere destinate accesări unor resurse virtuale, cum este de exemplu, **procfs** (sau Process file system).

Putem concluziona aşadar, faptul că un sistem de fişiere nu este nimic altceva decât o metodă de organizare, stocare şi accesare a datelor informatice prezente pe un mediu de calcul. Sistemul de fişiere Linux urmează toate aceste principii, cu mențiunea că este oarecum diferit față de sistemele de fişiere utilizate de alte sisteme de operare.

Putem caracteriza orice sistem de tip UNIX printr-o afirmație foarte simplă: Dacă ne referim la un sistem de tip UNIX, orice entitate care nu este un fișier, este un proces.

Utilizând această ideologie foarte simplă, orice sistem Linux poate fi descris pe scurt foarte precis. Exact ca și un sistem UNIX, Linux nu deosebește un fișier de un director în nici un fel, acesta nefiind altceva decât un fișier ce conține numele fișierelor incluse. Urmând această structură, programele (și evident și serviciile), fișiere text sau binare, până și dispozitive precum tastatura și mouse-ul, sunt considerate fișiere.

Pentru o organizare eficientă a acestor fişiere, s-a optat pentru structurarea lor într-o manieră ierarhică, precum a altor sisteme de operare. Pentru simplitate, vom reprezenta imaginea unui sistem de fişiere Linux precum cea a unui arbore, cu ramuri care la rândul lor au alte ramuri (directoare) care se termină în frunze (fişiere), însă pe măsură ce avansăm, vom observa că această reprezentare nu este în întregime corectă.

Tipuri de fişiere

Marea majoritate a fişierelor dintr-un sistem Linux sunt fişiere obişnuite (sau fişiere normale, de date) care conțin date precum text, imagini, fişiere audio sau programe. Urmând ideea că totul într-un sistem Linux este un fişier, există însă excepții pe lângă fişierele obişnuite:

- Directoarele (d) sunt fişiere ce conțin o listă a numelor fişierelor conținute de directorul în cauză.
- Fişiere speciale (c) sunt în general folosite pentru INPUT (introducere de date) sau OUTPUT (extragere de date). Aceste fişiere reprezintă "devices" (sau dispozitive) şi se găsesc în general în directorul "/dev"

- Fişiere SOCKET (s) reprezintă fişiere speciale folosite pentru comunicarea inter-procese printr-un mediu de rețea. Aceste fişiere sunt folosite pentru comunicații între diferite calculatoare sau rețele de calculatoare.
- Fişiere PIPE (p) sunt un fel de fişiere SOCKET cu rolul de a facilita comunicarea între procesele în desfăşurare.
- Legăturile (I) sunt folosite pentru a crea o legătură între un director sau fişier astfel încât aceştia să fie vizibili şi în alte locații.

În lista de mai sus, se observă că fiecare tip de fişiere are o literă delimitată de paranteze. Aceasta reprezintă prescurtarea tipului respectiv de fişier, şi poate fi observat în prima literă a şirului de permisiuni afişat prin comanda "**Is**".

Partiții

Principalul mediu de stocare al datelor este discul magnetic (sau hard disk) datorită vitezei mari de acces a datelor și posibilității de scriere și citire aleatoare, precum și datorită persistenței și siguranței datelor.

Majoritatea utilizatorilor au idee despre ceea ce înseamnă partiționarea unui disc, dat fiind faptul că aceasta este o facilitate oarecum independentă de sistemul de operare, și fiindcă orice sistem de operare posedă capabilitatea de a crea sau șterge o partiție.

Principalul motiv pentru care Linux folosește în mod implic mai multe partiții pentru o singură instalare este o mai bună organizare și securizare a datelor, astfel prin împărțirea acestora pe mai multe partiții, se asigură într-o oarecare măsură siguranța acestora în cazul în care ceva neașteptat se petrece. Această abordare datează de la începuturile sistemului Linux, când sistemul de fișiere nu posedă nici un fel de sistem de jurnalizare și orice pană de curent ar fi condus la comprimiterea datelor cu care se lucra în mod activ. Împărțirea sistemului pe mai multe partiții avea ca rol izolarea datelor astfel încât, în cazul petrecerii unui accident, măcar o parte din date să rămână neatinse.

OBSERVAȚIE: Este important de ştiut faptul că sistemul de jurnalizare nu protejează împotriva defectelor fizice alte mediului de stocare (cum ar fi blocurile stricate sau erorile logice), ci doar împotriva întreruperii neașteptate în funcționarea sistemului.

Tipuri de partiții

În Linux, putem identifica două mari tipuri de partiții: cele dedicate stocării de date, care de altfel sunt cele mai comune într-un sistem și partiții de tip SWAP (sau memorie virtuală) destinate suplinirii unei lipse de memorie a calculatorului.

În cadrul sistemului Linux, se folosește în general comanda **fdisk** pentru partiționarea (automat în momentul instalării) spațiului disponibil. Ca informație suplimentară, sistemul de fișiere Linux codifică partițiile de date

(care pot fi de tip ext3 cu jurnalizare, sau ext2 și versiuni mai vechi, fără jurnalizare) cu ajutorul codului numeric 83 și partițiile de tip SWAP folosind codul 82.

Pe lângă aceste tipuri de sisteme de fişiere, Linux mai oferă suport și pentru altele precum Reiser, NFS, NTFS sau FAT. De altfel, putem observa la unele sisteme aflate în producție, existența mai multor partiții de tipuri diferite: motivul pentru această diversitate ar putea fi pentru asigurarea unei compatibilități cu alte sisteme de operare precum MS Windows – deși Linux știe să interpreteze sisteme de fișiere tipice Windows (de exemplu NTFS sau FAT), reciproca nu este valabilă, o partiție de tip **ext** nefiind vizibilă unui sistem Windows.

Partițiile de tip SWAP sunt de regulă invizibile în mod normal, și sunt accesbilie exclusiv sistemului. Motivul pentru care se folosesc astfel de partiții este pentru a asigura continuarea funcționării sistemului în cazul în care acesta nu posedă suficientă memorie disponibilă. Existența unei partiții SWAP pentru suplinirea deficiențelor de memorie fizică ce pot să apară în funcționarea sistemului a fost inventată odată cu sistemul UNIX, fiind preluat de majoritatea competitorilor.

OBSERVAȚII:

- Comanda fdisk poate duce la pierderea datelor! Se recomandă citirea cu atenție a paginii de manual asociată cât şi efectuarea unui back-up în eventualitatea efectuării operației de partiționare.
- Deşi partiţia SWAP va asigura funcţionarea sistemului în parametrii normali chiar şi în eventualitatea în care acesta nu beneficiază de suficientă memorie fizică, este important de ştiut faptul că foloseşte hard discul pentru acest rol, şi ca urmare viteza memoriei virtuale astfel obţinute va fi mult mai mică comparativ cu viteaza obţinută de memoria fizică.
- Deşi dimensiunea recomandată a partiției de SWAP este în general dublul dimensiunii memoriei fizice, există cazuri particulare unde soluții generale nu vor funcționa optim. De exemplu, în cazul folosirii a două discuri, o partiție de SWAP pe fiecare disc va aduce performanțe mai bune decât o singură partiție de SWAP pentru amândouă, în cazul unei funcționări ce va efectua multe operații de intrare/ieşire. În general se recomandă citirea documentației fiecărui produs software folosit pentru determinarea dimensiunii optime a partiției de memorie virtuală.

Kernel-ul (sau inima sistemului Linux) este cel mai important fişier, şi ca atare, poate beneficia de o partiție proprie **/boot.** Restul de spațiu disponibil poate fi partiționat în funcție de zonele importante ale sistemului Linux, însă de multe ori acestea se regăsesc pe o singură partiție pentru simplitate.

OBSERVAȚIE: Modificarea dimensiunii unei partiții deja existente și în folosință este posibilă prin intermediul unor utilitare specializate, însă din această operație pot rezulta pierderi de date. Se recomandă efectuarea unui back-up înainte de orice astfel de operații.

Puncte de montare

În ideea celor menționate până în acest moment, este important de precizat faptul că sistemul Linux folosește un algoritm clar pentru denumirea dispozitivelor sistemului, și hard discurile nu fac excepție. În capitolul aferent dispozitivelor în Linux, vom studia acest aspect mai în amănunt, însă pe moment, pentru simplitate, nu vom menționa nimic despre metodologia de numire a dispozitivelor.

Într-un sistem Linux, toate partițiile sunt conectate la sistem printr-un punct numit "punct de montare" care nu reprezintă altceva decât o locație în cadrul sistemului de fișiere unde partiția va fi "montată".

Pentru a ataşa o nouă partiție în sistem, utilizatorul trebuie să se asigure în primul rând, că locația în care această partiție urmează să fie montată există, și este, preferabil, un director gol. În urma procesului de "montare" (pe care îl vom explica în capitolele următoare), utilizatorul va putea să acceseze conținutul partiției noi în directorul în care a fost montat. Acest proces este transparent și în esență face o legătură între dispozitivul care trebuie accesat, cu o locație din sistem. Ca exemplu, unitatea de CD-ROM se montează de obicei în directorul /media/cdrom.

Procesul de montare al dispozitivelor poate fi modificat, astfel încât să corespundă cerințelor adminstratorului de sistem, prin modificarea conținutului fişierului /etc/fstab. Unele partiții nu se montează automat odată cu pornirea sistemului, altele nu sunt conectate mereu (disc-uri flash) iar altele nu pot fi montate decât de utilizatorul "root". Detalii despre modalități de modificare ale fișierului de configurare vor fi menționate în secțiunile următoare.

Pentru a afla detalii despre partițiile montate pe un sistem care funcționează, putem folosi comanda **df** pentru a afişa informații despre capacitatea, numele și punctul de montare al fiecărei partiții.

user1@computer1:~\$	df	-h				
Filesystem		Size	Used	Ava i l	Use%	Mounted on
/dev/sda1		7.5G	2.46	4.8G	34%	/
tmpfs		506M	0	506M	0%	∕lib⁄init/rw
udev		10M	80K	10M	1%	∕dev
tmpfs		506M	0	506M	0%	/dev/shm
/dev/hda		80M	80M	0	100%	∕media/cdrom0
user1@computer1:~\$	_					

În imaginea anterioară, se poate observa rezultatul apelării funcției **df** cu parametrul **-h** (prescurtarea de la "human", care face aranjarea informațiilor într-un mod mai prietenos). Se pot vedea informații despre partițiile existente, punctul de montare, spațiul ocupat și spațiul liber, cât și partițiile SWAP inactive. Pentru descoperirea partiției pe care se află directoul curent, se poate apela tot comanda **df** cu parametrul adițional ".": "**df -h**."

Structura implicită de directoare

În marea majoritate a sistemelor Linux, structura de directoare implicită este "RH Linux FHS Compliant", adică respectă standardele FHS (Filesystem Hierarchy Standard). Ca urmare, o structură identică, sau similară va putea fi identificată pe foarte multe sisteme Linux. În contextul în care folosim modelul arbore de reprezentare al sistemului de fișiere, atunci directorul rădăcină este indicat prin caracterul "/" (slash), care mai este numit și "root directory" sau "root" - a nu se face confuzia cu utilizatorul "root", fiind două lucruri total diferite!

user10	computer	1:/\$ ls	-d *∕						
bin/ boot/ user10	cdrom/ dev/ computer	etc/ home/ 1:/\$ _	lib∕ lost+found∕	media∕ mnt∕	opt/ proc/	root∕ sbin∕	selinux/ srv/	sys/ tmp/	usr/ var/

Structurarea respectivă are rolul de a facilita folosirea sistemului Linux prin întocmirea unor standarde. Fiecare director, aparținând convenției FHS, are un rol bine definit:

- /bin conține de regulă programele prezente pe sistem.
- /boot fişiere de pornire cât şi kernel-ul sistemului. În distribuții mai noi, acest director poate conține şi fişierele de configurare al "boot loader"-ului grub.
- /dev conține referințe către toate dispozitivele periferice, reprezentate prin fişiere.
- /etc conține majoritatea fişierelor de configurare.
- /home directorul home conține directoarele home ale fiecărui user din sistem.
- /opt conține de obicei programe adiționale.
- /proc conține fişiere cu informații despre resursele sistemului. Se pot găsi astfel o sumedenie de informații despre resursele sistemului. Pentru mai multe informații, poate fi apelată comanda man proc.
- /root directorul home al utilizatorului "root".
- /var locul în care sunt ținute fişiere temporare, cum ar fi jurnale, liste de mesaje, etc.

Reprezentarea sistemului de fişiere folosind arbori

În ciuda uşurinței cu care utilizatorii înțeleg sistemul de organizare ierarhică a sistemului de fişiere, pornind de la un director rădăcină, și intrând în subdirectoare ce conțin alte directoare și fișiere, lucrurile nu stau așa și în realitate.

Deși modelul de reprezentare prin arbore este unul relativ exact, nu are nici un fel de relevanță când este raportat la modul în care datele sunt stocate pe disc în mod fizic.

Fiecare partiție a unui disc poate avea propriul sistem de fișiere, fapt clar în urma celor discutate anterior, iar acestea pot fi puse împreună pentru a forma un tot unitar. În realitate, fiecărui fișier îi corespunde un identificator numeric numit **inode**, identificator care va conține informații despre conținutul, proprietarul și locația fișierului pe disc.

Revenind la scenariul în care există mai multe partiții, este bine de știut că fiecare partiție are propriul său set de **inode**-uri, putând să apară astfel conflicte, unde două fișiere diferite pot avea același **inode**. Fiecare identificator descrie structura de date pe disc, propritatile fișierului cât și locația resursei la care face referință.

De regulă, la momentul inițializării sistemului, se crează un număr fix de **inode**-uri pe fiecare partiție, acest număr reprezentând numărul maxim de fișiere (de orice tip – chiar și directoare) care pot co-exista pe acea partiție la un moment dat.

Urmând acest sistem, când un fișier nou este creat, acestuia i se atribuie un **inode** marcat ca fiind liber. Identificatorul în cauză va primi următoarele informații: proprietar, tipul de fișier, permisiuni, ora și data la care fișierul a fost creat, ora și data ultimei modificări al **inode**-ului curent, numărul de legături la fișierul în cauză, o adresă fizică pe disc cât și dimensiunea fișierului. Pentru a vedea inode-ul unui fișier, se poate folosi comanda **Is** cu parametrul **-i**.

Folosind aceste informații, și completându-le cu numele fișierului și al directorului (care de altfel nu sunt incluse în **inode**), sistemul Linux poate construi și reprezenta o structură de fișiere ierarhică pe care utilizatorul să o înțeleagă mai ușor.

OBSERVAȚIE: Este bine de știut faptul că inode-urile fac parte din sistemul de fișiere și au spațiul lor dedicat pe disc.

Organizarea structurii de directoare

Structura de directoare tipică unui sistem Unix este cea discutată anterior, și poate să difere într-o oarecare măsură în funcție de distribuție, însă indiferent care ar fi distribuția pe care o folosim, este foarte important ca utilizatorul (mai ales adminstratorul) să poată naviga cu succes structura de directoare. Acest fapt se datorează în primul rând abilității de a accesa fișiere de configurație specifice (care se află în locuri bine stabilite în cadrul sistemului de fișiere) sau de a executa programe care nu se află în directorul curent.

O observație interesantă este aceea că majoritatea programelor importante ale sistemului (și nu numai) pot fi executate din orice locație. Ca un exemplu, programul **Is** poate fi executat fără nici un fel de informație referitoare la adresa sa în cadrul sistemului de fișiere, deși acesta se află doar în directorul /**bin/Is**. Motivul pentru care acest lucru este posibil, este existența unei variabile de sistem, care specifică interpretorului de comenzi (sau shell – în cazul de față **bash**) unde anume să caute un program. Această variabilă de sistem poartă numele de **PATH** și este semnalizată (drept o variabilă) prin caracterul **\$** ce o precede. Conținutul variabilelei **\$PATH** poate fi afişat cu ajutorul comenzii **echo**:

```
user1@computer1:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/games
user1@computer1:~$ _
```

La prima vedere, variabila de cale arată ciudat, dar la o privire mai atentă, putem observa că de fapt nu este vorba de o singură cale, ci de o listă de căi menționate, separate prin caracterul ":".

Ca urmare, în exemplul de mai sus, bash va cauta programul ce trebuie să fie executat succesiv în directoarele "/usr/local/bin", "/usr/bin", "/bin" şi în "/usr/games", urmând ca, în cazul în care acesta nu se află în niciun director, să afişeze mesajul "**command not found**". Dacă însă programul căutat a fost găsit în lista de căi, căutarea va lua sfârșit, putând să cauzeze erori – în cazul în care există două versiuni ale unui program, cu același nume, în locații diferite (amândouă incluse în cale), shell-ul va lansa primul program.

Pentru a căuta adresa unui program în cadrul sistemului de fișiere, putem folosi comanda "**which**".

user1@computer1:~\$ which man /usr/bin/man user10computer1:~\$ _

Rezultatul apelării acestei funcții, este afişarea pe ecran a căii către programul solicitat prin parametru. \$PATH nu este altceva decât o variabilă de sistem, și poate fi schimbată, iar alte variabile create prin intermediul funcției **export**.

```
user1@computer1:/etc$ echo $temp
user1@computer1:/etc$ export temp="Acesta este un test"
user1@computer1:/etc$ echo $temp
Acesta este un test
user1@computer1:/etc$ _
```

În exemplul de mai sus, se poate vedea diferența între două încercări succesive de afişare al conținutului variabilei temp. Prima tentativă relevă faptul că aceasta nu este setată, pe când a două (după o încărcare cu ajutorul funcției export) denotă schimbarea. În același fel se poate modifica și variabila \$PATH, însă utilizatorul trebuie să fie conștient că aceasta ar putea duce la inabilitatea sa de a mai executa orice program fără specificarea căii complete. Din fericire orice schimbări asupra variabilelor de sistem prin funcția **export** sunt temporare, și vor fi păstrate doar atât timp cât sesiunea curentă există. Dacă utilizatorul dorește să adauge temporar un director în cadrul variabilei \$PATH, tot ceea ce trebuie să facă este să execute

comanda **export \$PATH=\$PATH:/cale_nouă**. Comanda poate fi interpretată ca o atribuire simplă, unde variabilei îi va fi atribuită valoarea sa curentă alături de noua cale.

Pentru schimbări permanente asupra setărilor de acest fel, trebuie modificate fişierele de configurare ale aplicației shell.

Pentru o mai bună înțelegere despre orientarea în cadrul sistemului de fişiere, trebuie să menționăm și conceptele de căi **relative** și **absolute**. Pe scurt, amândouă reprezintă o adresă în structura ierarhică a sistemului de fişiere, cu mențiunea că o cale relativă se referă la o extensie a căii curente și va varia în funcție de directorul curent, pe când o cale absolută va reprezenta mereu aceeași locație, și va avea ca punct de referință directorul rădăcină. O cale absolută va începe mereu cu caracterul "/" reprezentând directorul de referință.

Interacțiunea cu fișiere

Precum am observat în imaginile prezentate până acum folosirea comenzii Is are ca efect afișarea fișierelor și directoarelor. O observație pertinentă poate fi accea că această comandă folosește o codificare în culori pentru a descrie tipul de fișier cu care avem de a face, pentru o interpretare mai ușoară:

- albastru directoare
- alb fişiere text
- roz imagini
- verde fișiere executabile
- mov deschis fişiere de tip legătură
- galben fişiere de tip dispozitiv
- roşu arhive

Un aspect important este și acela că ls nu va afisa implicit fișierele care încep cu caracterul ".", acesta fiind un mecanism folosit pentru a simplifica lucrul cu directoarele și de a ascunde fișiere care nu trebuie să fie accesibile ușor. Pentru afișarea tutoror fișierelor, chiar și a celor ascunse (hidden) se folosește comanda "**Is -a**".

Crearea directoarelor se face folosind comanda **mkdir** urmată de numele directorului ce urmează a fi creat. Pentru schimbarea directorului curent se va folosi comanda **cd** urmat de adresa directorului ce urmează a fi schimbat, iar pentru aflarea căii curente se poate folosi comanda **pwd**.

Copierea fişierelor se face folosind utilitarul **cp** urmat de mai mulți parametri:

 cp [-R] sursa destinație – parametrul R este opțional și provine de la recursiv, și va avea ca efect copierea completă cu tot cu subdirectoare. Ultimii doi parametrii simbolizează fișierul sau directorul sursă respectiv fișierul sau directorul destinație. **Mutarea fişierelor** se face folosind comanda **mv**, aceasta putând să fie folosită și pentru redenumirea fișierelor. Comandă va primi doi parametri, primul fiind fișierul ce trebuie mutat (cunoscut și drept sursă) iar al doilea fiind destinația. În cazul în care sursă și destinația se referă la același director, fișierul sursă va fi redenumit în fișierul destinație.

Comanda "**mv /home/user1/text1.doc /home/user1/documente/**" va avea ca efect mutarea fişierului text1.doc în directorul documente aflat la aceeaşi locație.

Ştergerea fişierelor sau a directoarelor se face folosind comanada **rm**. Aceasta poate să primească mai mulți parametri, ultimul reprezentând fişierul sau directorul tinta. Comanda nu poate să șteargă implicit decât un singur fişier, însă, folosind parametrul **-r** (ștergere recursivă), instrucțiunea poate să șteargă întregi ierarhii de directoare. Se recomandă atenție sporită în cazul apelării cu parametrul **-r**, acesta putând să producă pagube foarte mari în sistem.

Pentru a șterge un director se poate folosi și comanda **rmdir**, însă acesta trebuie să fie gol.

Printre comenziile de localizare disponibile într-un sistem Linux, putem aminti câteva mai importante:

comanda which este folosită pentru căutarea unui program în cadrul variabilei de cale \$PATH. Aceasta caută în lista de directoare inclusă, fişierul executabil primit ca parametru şi va returna locația fişierului căutat, în cazul în care acesta există. Comanda mai poate primi parametrul opțional -a care va verifica dacă o comandă este de fapt un alias pentru altă comandă. Un alias este de fapt o scurtătură către o comanda anume.

user1@computer1: 7/Desktop\$ alias	
alias ls='lscolor=auto'	
user1@compu <mark>ter1:</mark> ~/Desktop\$ home	
-bash: home: command not found	
user1@computer1:~/Desktop\$ alias home='	cd ~'
user1@computer1:~/Desktop\$ pwd	
/home/user1/Desktop	
user1@computer1:~/Desktop\$ home	
user10computer1:"\$ pwd	
/home/user1	
user1@computer1:~\$ _	

În exemplul de mai sus, putem observa atât o listă a alias-urilor curente, cât și o metodă simplă de a crea astfel de scurtături. Am creat astfel o scurtătură numită **home**, care de fapt ascunde o schimbare a directorului curent în directorul home al utilizatorului curent. Se poate observa de asemenea că până și comanda **Is** este de fapt un alias, la ea însăși: cu mențiunea că aceasta primește niște parametrii (ce permit afișarea colorată) pe care nu iar fi primit în mod normal.

 comanda find este un utilitar foarte puternic, folosit pentru căutarea unor fişiere în cadrul structurii de directoare, utilizatorul putând să descrie un set foarte specific de filtre pentru limitarea numărului de rezultate găsite. Deși modul în care această comandă este cel mai des folosită se rezumă la căutarea într-un anume loc al unui fișier al cărui nume este inclus în parametru.

```
user1@computer1:~$ ls
aa b bb Desktop new file
user1@computer1:~$ find . -name b
./b
user1@computer1:~$ find . -name "b*"
./bb
./b
./gconf/apps/panel/toplevels/top_panel_screen@/background
./.gconf/apps/panel/toplevels/bottom_panel_screen@
./.gconf/apps/panel/toplevels/bottom_panel_screen@/background
./.gconf/apps/panel/toplevels/bottom_panel_screen@/background
./.gconf/apps/panel/toplevels/bottom_panel_screen@/background
user1@computer1:~$ _
```

În exemplul de mai sus putem observa mai multe feluri de utilizare a comenzii **find**, prima având ca rezultat afişarea tuturor fişierelor din directorul curent (incluzând subdirectoarele) al căror nume este "b". Al doilea exemplu ne arată cum putem folosi "wildcards" pentru a afişa toate fişierele ale căror nume **încep** cu litera "b". Filtrele ce pot fi alese pentru restricționarea numărului de rezultate permit căutarea după conținutul fişierelor, data creării sau modificării, dimensiunea lor, etc.

Crearea de fişiere legătură

Prin legătura (sau scurtătură) putem să descriem un mecanism de legare a două sau mai multe fișiere la o singură resursă fixă (alt fișier). Există două tipuri de legături (sau link-uri):

Hard link – reprezintă un mecanism de asociere a două sau mai multor fişiere cu același inode. Rezultatul imediat va fi că două sau mai multe fişiere diferite vor fi legate la același inode. Ca urmare a acestui fapt, acest tip de legături nu pot fi pe partiții diferite, din moment ce inode-urile sunt unice pentru fiecare partiție.

Soft link – mai este cunoscut și drept symbolic link, și nu este altceva decât un fișier care arătă spre un alt fișier, conținând informații despre numele, calea și locația fizică a fișierului spre care arată și, drept urmare, pot aparatine unor partiții diferite.

Crearea link-urilor se face foarte simplu, utilizând comanda Is:

In -s /home/user1/b /home/user1/legătură

Apelarea funcției precedente va avea ca rezultat crearea unei conexiuni către fișierul "b" numită "legătură". Tipul de legătură este **soft link**, atestat prin prezența parametrului **-s**. În cazul în care nu se dorește crearea unei legături de tip **soft**, ci din contră se dorește crearea unei legături **hard link**, atunci trebuie exclus parametrul **-s** din comandă prezentată anterior.

Montarea dispozitivelor la sistemul de fişiere

Pentru a putea fi accesate, dispozitivele și implicit partițiile acestora trebuie montate într-un sistem Linux. Spre deosebire de alte sisteme de operare (Windows de exemplu), sistemele Linux nu fac o asociere între fiecare partiție și o literă.

Pentru a accesa conținutul unei partiții, aceasta trebuie să urmeze un proces numit **mounting**, prin care respectiva partiție va fi legată de sistemul de fișiere. Această legătură se face prin asocierea unui dispozitiv cu un director din sistemul de fișiere, care va putea fi folosit pentru accesarea conținutului acestuia. Acest director se numește **mount point**, iar procedeul de montare al unei partiții se numește **mounting**.

Procedeul de montare implică aşadar, două entități: un director reprezentând calea de acces și numele partiției ce se va monta. În general, marea parte a distribuțiilor Linux au în rădăcina un director numit **/mnt**, care este de regulă folosit pentru montarea diferitelor dispozitive. Utilizatorul nu este însă obligat să folosească acest director pentru montare, având libertatea de a folosi orice director dorește.

Urmând aceste reguli, putem deduce că orice partiție care este folosită și activă, a fost întâi montată. Excepție de la această regulă nu face nici partiția activă pe care sistemul este instalat, diferența fiind că aceasta este montată în mod automat în cadrul procesului de bootare pentru a evita diferite capcane ce pot să apară.

Urmând algoritmul de denumire al dispozitivelor și partițiilor într-un sistem Linux prezentate mai devreme, și având stabilit un director în care monteze o partiție, utilizatorul trebuie să folosească comanda **mount**:

mount [nume_partiție] [director]

Un exemplu bun de montare comună este aceea a unității CDROM / DVDROM, însă pentru acesta trebuie stabilit numele unității (conform schemei de numire Linux) și punctul de montare. În cazul în care unitatea optică este singura conectată pe magistrala IDE, aceasta va fi denumită /dev/hda. Distribuția Debian îl poate ajuta pe utilizator să determine cu acuratețe numele diferitelor dispozitive din sistem, prin crearea a unor link-uri simbolice generale în directorul /dev. O astfel de legătură simbolică este /dev/cdrom care odată examinată prin comanda Is –I /dev/cdrom, va arăta fişierul real spre care arată.



În urma executării comenzii **mount** /dev/hda /media/cdrom sistemul ne avertizează că dispozitivul este accesibil doar în modul read-only, și ca urmare va fi montat în acest fel. Structura de fișiere a discului prezent în unitatea optică va putea fi accesat în directorul /media/cdrom. Executarea comenzii **mount** fără nici un fel de parametru va afișa o listă cu dispozitive reale și pseudo-dispozitive atașate la sistemul de fișiere în momentul curent.

O greșeală curentă a multor utilizatori este înlăturarea mediilor fizice (discuri optice, hard discuri externe, etc) când acestea sunt încă montate, și poate avea consecințe grave: pierderea datelor scrise. În cazul unităților CDROM înlăturarea discului cât timp timp acesta este montat nu este posibilă – tăvița unității nu se va deschide până când aceasta nu este demontată.

Demontarea unei partiții se face folosind comanda **umount** (atenție, **nu** este "unmount") urmată fie de numele dispozitivului (în exemplul anterior /dev/hda), fie de numele directorului în care a fost montat.

umount /dev/hda sau umount /media/cdrom

Comanda **mount** mai poate fi folosită și cu parametrul **-t** pentru a specifica manual tipul de sistem de fișiere pe care îl folosește resursa ce va fi montată. În lipsa acestui parametru, sau dacă este folosit împreună cu tipul **auto**, sistemul va încerca să determine automat tipul necesar. Pentru mai multe detalii despre tipurile de sisteme de fișiere suportate, consultați **man mount**.

OBSERVAȚIE: În exemplul anterior, s-a folosit modul auto pentru determinarea sistemului de fişiere folosit pe suportul optic. În caz că detecția automată eşuează, se poate încerca comanda împreună cu parametrul **–t iso9660**, formatul standard pentru majoritatea dispozitivelor optice. A se observa de asemenea ca procesul de montare al unui dispozitiv necesită drepturile asociate contului root.

Montarea automată a partițiilor

În cazul resurselor fixe, precum partiții de pe alte discuri, se poate adopta o schemă de montare automată, astfel încât utilizatorul să nu mai fie nevoit să monteze o resursă pentru fiecare instanță a sistemului de operare.

Configurarea partițiilor în vederea montării automate se face prin intermediul fişierului de configurare **/etc/fstab**, prescurtat de la **file systems table**. Acesta conține de regulă o listă cu toate discurile și partițiile disponibile pe un sistem, și indică cum vor fi inițializate.

<pre># /etc/fstab: s #</pre>	tatic file syste	m inform	nation.		
# <file system=""></file>	<mount point=""></mount>	<type></type>	<options></options>	<dump></dump>	<pass></pass>
proc	/proc	proc	defaults	Θ	Θ
/dev/sda1	/	ext3	errors=remount	-ro 0	1
/dev/sda5	none	swap	SW	Θ	Θ
/dev/hda	/media/cdrom0	udf,iso	9660 user, noauto	o 0	Θ
/dev/fd0	/media/floppy0	auto	rw,user,noauto	Θ	Θ
/etc/fstab (END)				

Poza anterioară reprezintă un exemplu de cum ar putea arăta fișierul /etc/fstab. Se poate observa o organizare pe linii (similar cu multe alte fișiere de configurație), împărțite pe coloane separate prin spațiu sau tab. În continuare vom analiza pe rând scopul fiecărei coloane.

Prima coloană semnalează dispozitivul pe care se află respectivul sistem de fişiere. A doua coloană menționează directorul în care s-a făcut montarea (sau **mount point**), iar următoarea coloană conține tipul de sistem de fişiere.

Coloana numărul patru conține opțiunile suplimentare cu care se montează partiția respectivă. Cea de-a cincea coloană este citită de către utilitarul **dump** pentru a stabili momentele în care să efectueze un back-up al sistemului. Ultima coloană este folosită de utilitarul **fsck** pentru repararea erorilor sistemului de fișiere al unei partiții. Acest câmp poate avea 3 valori posibile:0 înseamnă că partiția nu va fi verificată, iar 1 **\$2** reprezint ă ordinea în care utilitarul va verifica. Se poate observa că directorul root are prioritate la verificare; în cazul în care utilizatorul doeste verificarea și celorlalte partiții, acesta ar trebui să seteze câmpurile respective la valoarea 2.

Un exemplu de modificare al fișierului **/etc/fstab** ar fi pentru montarea automată a discului partiționat într-unul din exemplele anterioare **/dev/sdb**. Pentru aceasta, utilizatorul ar trebui să adauge la finalul fisierulu **fstab** următoarea linie: **/dev/sdb1 /mnt/hdd2 auto auto 0 0**

Această linie poate fi intrerpretata în felul următor: setează partiția simbolizată de /dev/sdb1 să fie montată în mod automat în locația /mnt/hdd2 (acest director trebuie să existe în momentul în care se încearcă montarea), detectând automat sistemul de fişiere folosit, cât și opțiunile suplimentare. De asemenea această partiție va fi ignorată de utilitarul dump în vederea efectuării unui back-up, și nu va fi verificată pentru erori de către utilitarul fsck. Pentru a vedea o listă cu toate opțiunile posibile (pentru coloana a patra), consultați secțiunea dedicată parametrului –o în pagina de manual man mount.

Administrarea discurilor

Partiționarea discurilor în cadrul unui sistem Debian se poate face folosind o serie de programe. Urmând structura de până acum, vom analiza un program ce rulează în mod grafic și unul ce rulează în mod text.

Utilitarul **cfdisk** vine instalat cu orice distribuție Debian cât de cât recentă. Acesta program funcționează în mod text, și reprezintă o alternativă mult mai ușor de folosit decât tradionalul **fdisk**.

Utilitarul se lansează prin executarea comenzii **cfdisk /dev/[dispozitiv]** unde **dispozitiv** se referă la unitatea fizică, descrisă printr-un fișier de tip device. Interfața pusă la dispoziția utilizatorului, deși este una de tip text, se folosește relativ ușor, chiar și de către utilizatori mai puțin experimentați. Rularea aplicației fără nici un parametru, va determina alegerea primului disc de tip IDE, simbolizat prin /dev/hda. Considerând progresele tehnologice, hard discurile IDE nu mai sunt cea mai potrivită alegere pentru multe din sistemele mai noi. Ca urmare, dispozitivul /dev/hda ajunge de multe ori să reprezinte dispozitive de tip CD-ROM sau DVD-ROM și orice încercare de partiționare a mediilor de tip read-only va fi marcat printr-un mesaj de avertisment.

OBSERVAȚIE: Pentru a lansa **cfdisk**, utilizatorul curent trebuie să posede drepturi de super user (adică root). Se recomandă atenție sporită când se efectuează orice fel de schimbări asupra discurilor fizice, din moment ce există riscul pierderii datelor.

Alternativa grafică este **gparted**, un utilitar din suita GNOME. Prescurtarea sa vine de la GNOME Partition Editor, și este folosit pentru o serie largă de operațiuni pe discuri: creare și ștergere de partiții, redimensionarea partițiilor active, schimbarea ordinii fizice pe disc a partițiilor și formatarea acestora. Programul **gparted** poate fi lansat fie dintr-un terminal grafic, fie accesând meniul **System -> Administration -> Partition Editor**.

Interfața acestuia este una similară cu multe alte programe ce îndeplinesc funcții similare:

(10)				Record	
		/dev	/sdb - GParted		
<u>G</u> Parted <u>E</u> o	dit <u>∨</u> iew <u>D</u> ev	ice <u>P</u> artition <u>H</u> elp			
New Del	ete Resize/N	Aove Copy Paste	Undo Apply	/de	ev/sdb (8.00 GiB) ≎
		/dev/sdb1 6.05 GiB			/dev/sdb2 1.95 GiB
Partition	Filesystem	Size	Used	Unused	Flags
/dev/sdb1	. 📕 ext3	6.05 GiB	237,55 MiB	5.82 GiB	
/dev/sdb2	ext3	1.95 GiB	66.33 MiB	1.88 GiB	

Modul de funcționare este similar cu Synaptic Package Manager, mai ales datorită sistemului de înșiruire al comenzilor. Operațiile vor fi adăugate într-o coadă, și executate în ordinea corectă, numai după apăsarea butonului **Apply.** Așadar, orice schimbare va fi nulă în cazul în care utilizatorul părăsește programul înainte a acționa butonul respectiv. Pentru selectarea unui alt dispozitiv de stocare, trebuie acționat butonul din partea dreaptă superioară – vizibil în poză cu textul "/dev/sdb..". Se recomandă folosirea numai cu drepturi de root.

Modelul folosit pentru PC-uri este în așa fel gândit încât nu permite existența a mai mult de 4 partiții primare. Dintre acestea, una poate fi de tip extended, adică poate conține mai multe partiții logice. Prin intermediul **gparted**, aceste partiții pot fi formatate conform unui sistem de fișiere, utilizatorul având posibilitatea să aleagă dintr-o gamă foarte largă: **ext2**, **ext3**, **fat**, **hfs**, **ntfs**, **reiser**, **ufs**, **xfs**.

Algoritmul de denumire al dispozitivelor

Aşa cum probabil ați observat, Linux folosește fișiere de tip device pentru a interfața orice fel de dispozitive. Pentru a le putea identifica, a trebuit conceput un sistem de numire a acestora, bine stabilit, astfel să se elimine orice fel de confuzie.

Sub Linux, discurile și partițiile sunt denumite într-un mod total diferit față de alte sisteme de operare. Dacă în MS Windows, fiecare partiție este identificabilă printr-o literă (exemplu **c:**\), în Linux, dispozitivele sunt denumite după următorul algoritm:

- Hard Discul IDE setat ca master va fi denumit mereu /dev/hda
- Hard Discul IDE setat ca slave va fi denumit /dev/hdb

- Primul disc ATA sau SCSI (conform identificatorului său) va fi denumit /dev/sda
- Următoarele discuri ATA sau SCSI vor avea denumirea /dev/sdb, sdc, şi aşa mai departe.
- Dispozitivele de tip CDROM pe ATA sau SCSI vor fi /dev/scd* sau /dev/sr*.

Odată stabilite numele dispozitivelor fizice, mai rămâne doar un mod de a numi și partițiile de pe fiecare disc în parte. Acest fapt este realizat prin concatenarea unei număr zecimal după numele dispozitivului – dacă de exemplu am avea un hard disc conectat pe ATA cu trei partiții, acesta s-ar numi sda, iar partițiile ar fi identificabile după numele **sda1**, **sda2**, **sda3**.

Linux folosește cifrele 1, 2, 3 și 4 pentru identificarea partițiilor primare, pe când partițiile logice încep numărătoarea de la 5. În exemplul anterior, presupunând că cea de-a treia partiție ar fi fost una logică, aceasta nu s-ar fi numit **sda3**, ci **sda5**.

OBSERVAȚIE: Reamintim pe această cale, faptul că dacă un sistem are o partiție extinsă, aceasta este folosită doar pentru a conține partițiile logice, și ca urmare, nu este utilizabilă. De asemenea, convenția prin care atât dispozitivele conectate pe ATA cât și cele pe SCSI folosesc același nume nu este valabilă pentru versiuni de kernel mai vechi de 2.6 și este posibil ca alte distribuții să consolideze această excepție.

er dill