

1. NOTIUNI DE BAZA Linux/UNIX

1.1.Unix/Linux - scurt istoric.....	2
1.1.1.Cum a inceput totul.....	2
1.1.2.Distributii Unix/Linux.....	2
1.2.Structura sistemului de operare Linux/UNIX.....	3
1.2.1.Elemente de baza.....	3
1.2.2.Kernel-ul Linux/Unix.....	4
1.2.3.Shell-ul.....	4
1.2.4.Sistemul de fisiere.....	4
1.3.Notiuni de baza despre utilizarea sistemului de operare Linux/UNIX.....	5
1.3.1.Variante de interactiune cu sistemul de operare.....	5
1.3.2.Facilitati si concepte legate de interfata grafica Linux.....	6
1.3.3.Facilitati si concepte legate de shell-ul Linux.....	7
1.3.4.Conturi de utilizator si privilegii.....	9
1.3.5.Oprirea si restartarea sistemului de operare.....	9
1.4.Help sub Linux/UNIX.....	9
1.4.1.Unix Programmer's Manual (man pages).....	9
1.4.2.Help despre comenzile built-in de shell.....	11
1.5.Structura logica a sistemului de fisiere Linux/Unix.....	12
1.5.1.Caracteristici generale.....	12
1.5.2.Tipuri de fisiere Linux/Unix.....	12
1.5.3.Directoarele create la instalare si rolul acestora.....	13
1.6.BIBLIOGRAFIE.....	15
1.7.ANEXA 1 - Exemflu de manpage.....	16

1.1. Unix/Linux - scurt istoric

1.1.1. Cum a inceput totul

UNIX reprezinta un sistem de operare aparut in 1969 la Bell Labs (o divizie de cercetare a AT&T). Revolutionar pentru vremea sa, Unix oferea posibilitatea ca mai multi utilizatori sa foloseasca acelasi computer, profitand de timpii morti aparuti in executia programelor fiecaruia dintre ei - o forma incipienta de multitasking. Istoria sa se impleteste cu cea a limbajului C (creat initial pentru a usura portarea Unix-ului pe diferite configuratii hardware) si cu a stivei de protocoale TCP/IP (adaugare ulterioara ce a oferit Unix-ului capabilitatea de lucru in retea).

Unix a debutat ca un sistem de operare free, distribuit in retea universitara, unde cercetatorii i-au adaugat multiple facilitati. Una dintre primele derivate majore ale Unix-ului original poarta numele de BSD, fiind lansata in 1977 de colectivul de la University of California, Berkeley.

1969	Unix
1977	BSD
1983	SunOS
1984	GNU
1985	Windows
1991	Linux
1992	Solaris (SunOS 5.x)

Unix-ul a captat treptat atentia companiilor, incepand sa fie distribuit comercial, in sistem closed-source. Acest lucru ii deranjeaza pe cei obisnuiti cu libertatea de folosire si imbunatatire a softului; in 1984, Richard Stallman fondeaza GNU (acronim recursiv: GNU's Not Unix), ce isi propune sa creeze un sistem de operare free cat mai asemanator cu Unix. Pentru a se asigura de faptul ca rezultatul muncii sale nu se va putea transforma vreodata in soft proprietar, Stallman creeaza licenta GPL (General Public License) sub care sunt lansate toate aplicatiile GNU, si care dicteaza ca autorul unui astfel de soft este obligat sa distribuie software-ul impreuna cu sursele, astfel incat altii sa-l poata analiza si imbunatati, iar cei care preiau si modifica un soft GNU sunt obligati sa distribuie la randul lor softul impreuna cu sursele. GPL deschide calea imbunatatirii softului prin contributii individuale, interzicand ca, intr-un punct oarecare al acestei cai, softul sa devina closed-source.

In 1991, Linus Torvalds creeaza prima versiune a nucleului unui sistem de operare pentru computere compatibile IBM-PC, ce se doreste a fi unul free si asemanator cu Unix-ul, el primind un nume derivat din cel al creatorului inasa cu o consoana finala ce identifica modelul original: Linux. Beneficiind de software-ul deja creat de GNU, se naste astfel un sistem de operare free, kernel-ul Linux fiind la scurt timp dupa uniune relansat sub licenta GPL.

1.1.2. Distributii Unix/Linux

O distributie Linux include:

- kernel-ul Linux, cu posibile modificari fata de cel original, si optimizat conform scopului distributiei
- un ansamblu de software GNU necesar administrarii si folosirii sistemului de operare
- aplicatii suplimentare, selectate si ele in functie de scopul distributiei
- (optional) un mediu grafic – in general compus din serverul Xorg, un window manager si un desktop environment, plus eventuale aplicatii grafice aditionale

Distributiile Linux se diferentiaza in functie de multi factori, printre care amintim:

- platforma hardware pe care ruleaza – in ziua de astazi Linux poate rula pe aproape orice (computere ce folosesc procesoare Intel, SPARC sau PowerPC, pe dispozitive portabile etc)
- scopul declarat al distributiei - unele distributii isi propun usurinta in utilizare si sunt gandite pentru folosirea computerului ca desktop, pe cand altele sunt destinate rularii de servere. In functie de scop, pot diferi:
 - o software-ul inclus in kitul de instalare
 - o setul de utilitare proprii ale distributiei in cauza (soft dezvoltat de catre autorii distributiei)

- o felul in care este optimizat kernelul
- o felul in care se desfasoara instalarea sistemului de operare

In zilele noastre exista sute de distributii Linux. In functie de usurinta de utilizare, iata cateva exemple de distributii celebre:

- distributii user-friendly
 - o **RedHat** – reprezinta unul dintre primele Linux-uri majore care au avut ca scop usurinta in utilizare. Ultima versiune free a fost RedHat 9; in ziua de astazi compania RedHat produce o versiune comerciala destinata firmelor (RedHat Enterprise Linux - RHEL), iar continuatorul open-source al RedHat este proiectul **Fedora**. De asemenea, pe baza surselor de RHEL este creata distributia **CentOS**, practic o clona de RHEL dar care beneficiaza de stabilitatea si siguranta versiunii comerciale
 - o **Mandrake/Mandriva** – derivat initial din RedHat, Mandrake a devenit una dintre distributiile majore ce se remarcu prin utilizarea facila. Dupa fuzionarea cu Conectiva Linux si-a schimbat denumirea in Mandriva. In ziua de astazi Mandrake/Mandriva nu mai exista, insa viziunea sa a fost preluata de cateva distributii derivate (ex: **Mageia, OpenMandriva, PCLinuxOS**)
 - o **Ubuntu** – fenomenul ultimilor ani, o distributie derivata din Debian si gandita sa ofere maxim de usurinta in utilizare si suport hardware bun
 - o **SuSe/OpenSuse** – o distributie similara cu RedHat (preluand multe elemente din el) insa independenta de acesta, axata pe usurinta in utilizare fara a sacrifica insa din flexibilitate
- distributii pentru specialisti – cu mai putina preocupare pentru aspectul vizual, sunt axate pe flexibilitate, pentru a permite unui utilizator/administrator versat sa-si realizeze scopurile pe calea dorita
 - o **Debian** – cea mai clonata distributie Linux. Un Linux matur, stabil, sigur, compus din software free si open-source, ce dispune de o impresionanta colectie de pachete software
 - o **Gentoo** – o distributie Linux in care abordarea administrarii de software se face diferit fata de multe altele: sistemul de gestionare a software-ului preia aplicatiile sub forma de surse si le poate compila pe loc, adaptandu-le la platforma gazda si personalizandu-le setarile in functie de preferintele administratorului sistemului
 - o **Slackware** – considerata a fi cea mai veche distributie inca “in viata”, Slackware tinteste simplitatea, stabilitatea si asemanarea cu vechiul Unix. Administrarea de sistem se bazeaza pe editare de fisiere text, iar sistemul de management al softului este unul simplificat la maximum

In ziua de astazi exista inca un numar important de sisteme de operare considerate urmasi ai Unix-ului original, majoritatea lor fiind insa proprietare. Dintre ele amintim:

- **Solaris** – Unix-ul firmei Sun Microsystems, unul dintre sistemele de operare foarte bine cotate pentru operarea de servere. Odata cu preluarea Sun de catre Oracle, denumirea s-a modificat in Oracle Solaris
- **AIX** – versiunea de Unix dezvoltata de IBM
- **HP-UX** – Unix-ul companiei HP
- **FreeBSD, OpenBSD, NetBSD** – urmasi ai BSD-ului original

1.2. Structura sistemului de operare Linux/UNIX

1.2.1. Elemente de baza

Sistemele de operare Linux/UNIX sunt compuse din trei elemente de baza:

- **kernel** – asigura functionalitatea de baza a sistemului de operare, cea care permite rulara in paralel a mai multor aplicatii si comunicarea cu dispozitivele hardware
- **shell** – interpretorul de comenzi, ce permite utilizatorului interactiunea cu sistemul de operare

- **sistem de fisiere** – stocheaza structurat majoritatea aplicatiilor si datelor folosite de catre utilizatorul computerului

Fiecare dintre aceste componente va fi prezentata in continuare.

1.2.2. Kernel-ul Linux/Unix

Kernel-ul reprezinta nucleul sistemului de operare; el este primul incarcat la pornirea sistemului de operare si realizeaza gestionarea resurselor computerului, oferind posibilitatea rularii mai multor aplicatii in paralel (multi-tasking). Printre functiile indeplinite de kernel se numara:

- *interactiunea cu hardware-ul prezent in sistem* – driverele ce controleaza diversele dispozitive hardware sunt parte a kernelului
- *gestionarea procesorului/procesoarelor* – in general, intr-un sistem de operare, numarul de procese este mult superior celui de procesoare; de aceea, fiecare procesor reprezinta o resursa partajata intre mai multe procese, fiind necesara gestionarea sa de asa natura incat, in final, fiecare proces sa capete timp de procesor si sa se poata astfel executa
- *gestionarea memoriei* – fiecare proces beneficiaza de propriul sau spatiu de memorie, separat de al celorlalte procese sau de al kernel-ului

1.2.3. Shell-ul

Shell-ul reprezinta interpretorul de comenzi, si pe vremea aparitiei Unix-ului era singura modalitate de interactiune cu un computer, interfetele grafice bazate pe ferestre aparand mai tarziu. Shell-ul constituie interfata intre kernel si utilizator: simpla existenta a kernelului ofera doar *posibilitatea* rularii mai multor procese - este nevoie insa de o modalitate de a interactiona cu sistemul de operare, transmitandu-i operatiile ce se doresc efectuate.

Daca in alte sisteme de operare axate pe linie de comanda interpretorul era unic (ex: command.com in MS-DOS), in Linux/Unix utilizatorul are de ales intre mai multe interpretoare de comenzi posibile – iata cateva exemple:

- **sh** (Bourne Shell) – reprezinta unul dintre primele interpretoare de comenzi folosite in Unix. Fiind destul de lipsit de facilitati, in ziua de astazi nu se mai foloseste, insa pe baza sa au fost create alte shell-uri mai performante
- **ksh** (Korn Shell) – un interpretor cu sintaxa asemanatoare cu sh insa cu facilitati imbunatatite
- **csh** (C Shell) - interpretor cu sintaxa asemanatoare cu cea din C
- **bash** (Bourne Again Shell) – interpretorul implicit din Linux-ul zilelor noastre, cel mai bogat in facilitati dintre cele prezentate

1.2.4. Sistemul de fisiere

Daca sistemul de operare ar dispune numai de kernel si shell, utilizarea computerului ar fi restransa la comenzile cunoscute de catre shell (comenzile built-in). Pentru a asigura extensibilitatea sistemului de operare, este necesar sa se poata adauga noi aplicatii (comenzi), separate fata de cele incluse in shell; acestea sunt memorate in sistemul de fisiere.

Sistemul de fisiere reprezinta o modalitate de stocare structurata, persistenta a informatiei. Structura sa presupune o ierarhie de fisiere si directoare, ce permite identificarea in mod unic a oricarei resurse memorate. Sistemul de operare insusi este stocat in sistemul de fisiere, deoarece el contine, pe langa kernel si shell, o **intreaga suita de aplicatii ce permite exploatarea computerului.**

Contutul sistemului de fisiere poate proveni (partial sau total) din diferite surse:

- de pe hard-disk-ul/hard-disk-urile statiei pe care este instalat sistemul de operare ce ruleaza pe statia in cauza
- din retea – fie stocat pe un server si partajat in retea, fie preluat integral din retea (in cazul clientilor care booteaza din retea)

1.3. Notiuni de baza despre utilizarea sistemului de operare Linux/UNIX

1.3.1. Variante de interactiune cu sistemul de operare

In ziua de astazi, odata cu dezvoltarea interfetelor grafice, utilizatorul are la dispozitie doua modalitati de interactiune cu computerul:

- cea bazata pe o **interfata grafica** – utilizatorul foloseste mouse-ul si tastatura pentru a interactiona cu un sistem grafic compus din ferestre si icon-uri. Este modalitatea de lucru preferata de catre simplul utilizator, a carui necesitate principala este utilizarea computerului, nu administrarea acestuia
 - o avantaje
 - usurinta in utilizare - necesita minim de cunostinte legate de hardware sau arhitectura sistemului de operare
 - o dezavantaje
 - operatiile posibile sunt cele oferite de aplicatiile grafice disponibile; flexibilitatea poate fi mare, dar nu maxima
 - interfata grafica nu este intotdeauna disponibila (ex: pe servere ea in general lipseste)
- cea bazata pe un **interpretor de comenzi** – linia de comanda ramane principala modalitate de interactiune cu serverele Linux/Unix in scopul administrarii sistemului de operare/serverelor. Chiar si atunci cand pe o statie Linux ruleaza o interfata grafica, utilizatorul are posibilitatea de a deschide o fereastră de terminal (asemanatoare unui Command Prompt din Windows) pentru a lucra in linia de comanda.
 - o avantaje
 - ofera flexibilitate maxima – utilizatorul poate combina comenzile disponibile in mai multe feluri pentru a atinge un acelasi scop. Utilizatorii pot astfel folosi sistemul de operare in concordanta cu nivelul lor de cunostinte si pricepere, senzatia rezultata fiind una de libertate in operare si de flexibilitate
 - unele comenzi/operatii pot fi realizate mai rapid din linie de comanda de catre administratorul experimentat
 - deseori pe un server este singura modalitate de administrare (prin conectare de la distanta si folosirea unei linii de comanda in mod text)
 - o dezavantaje
 - incomoda pentru utilizatorul incepator
 - solicita cunostinte mai bogate legate de structura sistemului de operare, sintaxa comenzilor etc.

1.3.2. Facilitati si concepte legate de interfata grafica Linux

1.3.2.1. Comparatie cu Windows

Utilizatorul obisnuit cu modul de interactiune Windows va gasi multiple similaritati intre modul de interactiune cu interfata grafica in Windows si Linux/Unix. Exista insa destule deosebiri (mai mult sau mai putin subtile), o parte dintre acestea fiind punctate in continuare.

Ceea ce numim "interfata grafica" in Linux este de fapt un ansamblu de aplicatii, fiecare cu rolul sau bine stabilit, care nu face parte din kernel. Spre deosebire de Windows, interfata grafica nu este indispensabila functionarii sistemului de operare, ci are regimul unei aplicatii – asa cum se intampla la inceput cu Windows-ul insusi (care era pe vremea aceea o simpla aplicatie pornita din sistemul de operare DOS).

1.3.2.2. Elementele componente ale interfetei grafice

Interfata grafica Linux este compusa din:

- **serverul grafic** – este aplicatia care dispune de drivere pentru diferite tastaturi, mousi, hardware video etc si care ofera infrastructura pentru crearea unei interfete grafice bazate pe ferestre. Serverul grafic are capabilitati native de lucru in retea – de exemplu, este posibil ca o aplicatie grafica sa ruleze pe o statie, inasa output-ul ei sa fie afisat in serverul grafic aflat pe o alta statie.

Serverele grafice Linux/Unix ale zilelor noastre sunt un general derivate din proiectul X (initiat de MIT in 1984). Pana acum cativa ani, standardul de facto in Linux era Xfree86, ultimii ani consacrand inasa Xorg ca server grafic folosit in Linux.

- **window manager** (gestionarul de ferestre) – este aplicatia care se ocupa de plasarea, dimensionarea, desenarea si decorarea ferestrelor. Aplicatiile isi deseneaza output-ul in ferestrele oferite de catre window manager. Exista o intreaga varietate de window manager-e, cu facilitati mergand de la simpla creare de ferestre care nu se suprapun, pana la desktop-uri 3D cu efecte impresionante
- **desktop environment** (mediul de lucru desktop) – reprezinta suita de aplicatii grafice de care utilizatorul are nevoie pentru a-si explata computerul, asociate de obicei cu o anumita suita de componente grafice si un anumit mod de interactiune cu computerul. Un desktop environment poate include aplicatii pentru editare de documente (suita office), redare media, navigare pe internet, messaging, dezvoltare de soft etc.

Iata cateva dintre desktop environment-urile mai cunoscute:

- **KDE** – folosit preponderent in Linux (dar beneficiind si de versiune de Solaris). Oferă o aranjare a spatiului de lucru asemanatoare cu Windows (o bara cu aplicatii in partea de jos a ecranului, un meniu asemanator ca functionalitate cu cel de Start etc). Foloseste ca window manager din oficiu KWin
- **GNOME** – provenit de asemenea din Linux, inasa intalnit astazi pe multe alte Unix-uri (Solaris include un desktop environment numit Java Desktop System, bazat pe GNOME). Oferă o aranjare a spatiului de lucru usor diferita, folosind doua bare (in partea de sus si de jos a ecranului). Foloseste ca window manager default MetaCity (in GNOME 2) sau Mutter (in GNOME 3)

1.3.2.3. Scurta orientare

Pentru utilizatorul nou-venit in Linux, una dintre primele probleme o reprezinta a identifica aplicatiile ce realizeaza operatiile obisnuite. Iata cateva echivalente grafice Windows-Linux.

Scopul programului	Solutie tipica pentru Windows	Solutie pentru Linux
suita office	Microsoft Office	Open Office, LibreOffice
web browser	Internet Explorer	Mozilla Firefox, Opera, Chrome
messenger	Yahoo messenger	Pidgin, Kopete
client email	Outlook Express	Thunderbird, Kmail, Evolution
manager fisiere	Total Commander	Krusader, Gnome Commander
editor de text	Notepad	Kwrite, Kate, Gedit
linie de comanda	Command Prompt	Konsole, Gnome-terminal
player muzica	Windows Media Player, WinAmp	XMMS, Amarok, Rythmbox
redare filme	Windows Media Player, RealPlayer	Mplayer, Totem/Videos

Nota: posibilitatile de pe coloana de Linux sunt mult mai bogate – au fost incluse in tabel doar cateva variante.

1.3.3. Facilitati si concepte legate de shell-ul Linux

1.3.3.1. Terminal vs consola

Acesti doi termeni apar deseori atunci cand vine vorba de lucrul in linia de comanda Unix. Ei provin de pe vremea cand computerele abia incepusera sa fie dispozitive interactive (ele functionau initial pe baza de cartele perforate) iar interactiunea utilizatorului cu computerul se facea printr-un asa-numit terminal, existand posibilitatea ca la un acelasi computer sa fie conectate mai multe terminale pentru o cat mai eficienta utilizare a resurselor statiei in cauza. Terminalul reprezenta de fapt modalitatea de a furniza input computerului si de a primi output-ul produs de acesta. Terminalele erau compuse in general din tastatura si monitor (sau imprimanta!) si se conectau la computer folosind conexiuni seriale.

Dintre terminalele conectate la un computer, unul avea statut privilegiat: cel desemnat a fi "consola" sistemului, adica terminalul pe care sunt afisate informatiile de diagnostic la bootarea sistemului de operare si singurul de pe care pot fi realizate anumite operatii administrative (ex: shutdown, restart). De aceea, acest terminal era de obicei plasat intr-o locatie protejata fizic.

In ziua de astazi, utilizarea terminalelor fizice este inca posibila, insa rara; in schimb sunt folosite extensiv emulatoarele de terminal – programe care permit conectarea la o statie simuland capabilitatile unui terminal. Iata exemple:

- administrarea serverelor se face in general de la distanta, prin LAN/internet, folosind aplicatii ce se conecteaza la serverul dorit si dau apoi posibilitatea administratorului sa execute comenzi pe server si sa vizualizeze output-ul acestora. Un exemplu tipic sunt softurile cu rol de client SSH
- pe o statie pe care ruleaza mediul grafic, utilizatorul poate obtine o linie de comanda rulant un program ce deschide un pseudo-terminal

In plus, termenul de consola este folosit astazi pentru a desemna tastatura/monitorul direct conectate la computer. Unele Unix-uri ofera astfel o singura consola, pe cand Linux ofera 6 asa-numite *console virtuale*. Ele se numesc virtuale deoarece utilizatorul dispune, practic, de 6 console diferite desi fizic exista o singura pereche tastatura/monitor ale statiei in cauza.

Consolele virtuale Linux sunt accesibile folosind combinatiile de taste CTRL-ALT-F1...CTRL-ALT-F6. Prezenta celor 6 console faciliteaza o mult mai buna organizare a lucrului in linia de comanda; spre exemplu, utilizatorul poate deschide in prima consola documentatia unui server pe care intentioneaza sa-l configureze, in consola 2 poate deschide un editor in care sa aduca modificari fisierului de configurare al serverului, in consola 3 sa dea comenzile de pornire/oprire/etc a serverului, iar in consola 4 sa urmareasca log-urile serverului.

1.3.3.2. Obținerea unei linii de comanda Linux/Unix

Pentru a putea lucra in linie de comanda, este necesara pornirea unei instante de interpretor de comenzi (shell). Acest lucru se poate realiza in diferite moduri:

- daca sistemul Linux a fost configurat sa booteze fara interfata grafica, dupa pornirea sistemului de operare utilizatorul se va gasi in fata unui prompt de login, iar dupa o autentificare reusita va fi pornit automat un shell
- daca sistemul booteaza direct in serverul grafic:
 - utilizatorul se poate loga in modul grafic, pornind apoi o aplicatie de terminal (exemple: konsole, gnome-terminal, xterm)
 - utilizatorul poate folosi combinatiile de taste *CTRL-ALT-F1...CTRL-ALT-F6* pentru a comuta intr-una dintre consolele virtuale, urmand a reveni in serverul grafic in caz de nevoie folosind *CTRL-ALT-F7*

1.3.3.3. Structura unei comenzi Linux/Unix

Sintaxa generala a unei comenzi este urmatoarea:

```
nume_comanda [optiuni] [argument1] [argument 2] ...
```

Nota: spre deosebire de Windows, in Linux/Unix locul optiunilor este in general intre comanda si argumentele acesteia.

Comanda poate corespunde uneia dintre urmatoarele posibilitati:

- *fisier executabil* aflat in sistemul de fisiere. Acestea formeaza majoritatea comenzilor disponibile intr-un terminal/console. Exemplu: comanda *ls*, folosita pentru listarea de directoare
- *comanda built-in* - sunt comenzi recunoscute de catre insusi shell-ul, si care nu au executabile corespondente pe hard-disk. Ele formeaza nucleul necesar pentru gestionarea executiei celor din prima categorie. Exemplu: comanda *set*, folosita pentru setarea unor parametri ai shell-ului curent
- *alias* sau *functie* - sunt comenzi "virtuale", create pe baza altora deja existente din primele doua categorii. Ele vor fi prezentate in capitolul dedicat scripting-ului

Optiunile se recunosc dupa faptul ca incep cu - sau -- si au rolul de a modifica/personaliza executia comenzii. Ele nu sunt obligatorii, fiind utilizate numai atunci cand se doreste modificarea modului de operare al comenzii. Optiunile pot fi:

ls -a	Comanda ls si optiunea scurta -a
ls -a -R	Comanda ls si optiunile scurte -a si -R
ls -aR	Comanda ls si optiunile scurte -a si -R, concatenate

- scurte (de o litera) – exemplu: *ls -F*. Atunci cand se doreste folosirea mai multor optiuni simultan, majoritatea comenzilor permit concatenarea acestora (exemple in tabelul alaturat)
- lungi (de tip GNU) – incep cu -- si contin unul sau mai multe cuvinte separate prin - .
Exemplu: *ls --almost-all*

Argumentele comenzii sunt determinate de scopul acesteia si reprezinta resursele carora li se aplica comanda: ele pot fi nume de fisiere sau directoare, adrese IP, nume de statii etc.

1.3.3.4. Combinatii de taste utile in linia de comanda

- *CTRL-ALT-F1...CTRL-ALT-F6* – comuta intre consolele virtuale
- *CTRL-ALT-F7* – revenirea in serverul grafic dintr-o consola virtuala
- *CTRL-d* – paraseste shell-ul curent (echivalentul executarii comenzii *exit*)

- **CTRL-c** – opreste executia comenzii curente (la fel ca in DOS)
- **CTRL-s** – sisteaza temporar output-ul aplicatiei curente (de exemplu, pentru programe care genereaza output bogat, daca dorim sa analizam o anumita portiune a sa inainte de a disparea de pe din ecran)
- **CTRL-q** – reluarea output-ului oprit in urma comenzii anterioare
- **CTRL-u** – sterge pana la inceputul liniei

1.3.4. Conturi de utilizator si privilegii

Indiferent de varianta aleasa/disponibila pentru interactiunea cu sistemul de operare, utilizarea acestuia presupune folosirea unui cont de utilizator (username+parola) pentru a “intra in sistem”. Ca si in Windows, fiecare cont are asociata o lista de privilegii, iar din acest punct de vedere avem urmatoarea impartire a utilizatorilor:

- userul **root** - reprezinta unicul cont cu drepturi depline in sistemul de operare. El este folosit pentru administrarea de sistem; este contraindicata utilizarea sa pentru lucrul de zi cu zi in sistemul de operare. Multe Linux-uri fie nu permit login-ul ca root in mediul grafic, fie avertizeaza ca login-ul grafic cu acest cont nu este recomandat
- useri neprivilegiati - toti ceilalti useri din sistem au drepturi mai reduse decat root; sistemul de privilegii clasic din Linux/Unix este de asa natura organizat incat nu se poate crea un alt user egal in drepturi cu root.

Atunci cand suntem intr-un shell, nivelul nostru de acces este de obicei indicat prin caracterul aflat in stanga cursorului - respectiv # pentru root si \$ pentru user neprivilegiat, ca in exemplul de mai jos:

```
student@server:~$ ← login ca user neprivilegiat
root@server:/etc# ← login ca root
```

1.3.5. Oprirea si restartarea sistemului de operare

Pentru a opri sau restarta sistemul de operare, utilizatorul are la dispozitie urmatoarele comenzi:

- pentru oprirea sistemului de operare:
 - **poweroff** - inchide sistemul de operare si stinge calculatorul, fara a afisa niciun prompt sau fereastra de confirmare
 - **shutdown -h now** - echivalent cu poweroff, insa comanda permite si o oprire programata (ex: peste 5 minute → shutdown -h +5 sau la ora 22:15 → shutdown -h 22:15)
- pentru restart
 - **reboot** - reporneste sistemul de operare, fara a afisa prompt sau fereastra de confirmare
 - **shutdown -r now** - echivalent cu reboot, insa restartul poate fi de asemenea programat, ca in exemplele de mai sus

1.4. Help sub Linux/UNIX

1.4.1. Unix Programmer's Manual (man pages)

1.4.1.1. Descriere si mod de organizare

Principala sursa de informatie in linia de comanda o reprezinta Programmer's Manual (asa-numitele *man pages*, de la utilitarul *man* folosit pentru vizualizare), fisiere informative in care creatorii aplicatiilor clarificau functionarea acestora. Abordarea documentatiei era una sincera, existand o sectiune in care creatorul descria inclusiv bug-urile (hibele) programului.

Paginile man cuprind informatii atat pentru utilizatorul sau administratorul Linux (ex: help despre comenzile disponibile sau fisiere de configurare) cat si pentru programatorul Linux (ex: biblioteci de functii, apeluri de sistem, definitii de functii etc).

Manpage-urile sunt structurate in **sectiuni**, care reprezinta seturi (categorii) de manpage-uri grupate dupa un anumit domeniu. Iata structura acestora in Linux:

- *Sectiunea 1* – Comenzi disponibile utilizatorilor
- *Sectiunea 2* –Apeluri de sistem (adresate kernelului)
- *Sectiunea 3* – Biblioteci de functii si subrutine; sectiunile 2 si 3 sunt adresate programatorilor
- *Sectiunea 4* – Fisiere speciale
- *Sectiunea 5* – Formate de fisiere
- *Sectiunea 6* - Jocuri
- *Sectiunea 7* – Diverse (“miscellaneous”) - manpage-uri care nu au putut fi incadrate in alta sectiune
- *Sectiunea 8* – Comenzi de administrare a sistemului

Locatia de baza a manpage-urilor in Linux este `/usr/share/man`, insa fiecare aplicatie instalata suplimentar fata de sistemul de operare are posibilitatea de a-si instala propriile manpage-uri in ce director doreste, comanda `man` avand optiuni ce permit specificarea cailor din sistemul de fisiere in care sunt cautate manpage-urile.

Fiecare manpage este impartit la randu-i in niste sectiuni (capitole); in tabelul de mai jos sunt prezentate cateva exemple, impreuna cu scopul fiecarei sectiuni. Dintre ele, doar NAME, SYNOPSIS, DESCRIPTION si SEE ALSO sunt obligatorii, restul pot lipsi.

NAME	Numele manpage-ului si descrierea scurta (cea in care cauta man -k)
SYNOPSIS	Prezinta sintaxa de utilizare (ex: pentru comenzi, modul de apelare si optiunile posibile, fara a detalia scopul lor)
DESCRIPTION	Descrierea pe larg a subiectului manpage-ului - fie el comanda, fisier de configurare, functie etc.
OPTIONS	Prezenta in general doar pentru comenzi (sectiunile 1 si 8); listeaza si explica optiunile posibile ale comenzii
FILES	Prezinta fisierele utilizate de catre comanda/functia ce face subiectul manpage-ului
BUGS	Listeaza bug-urile (defectele) sau limitarile cunoscute ale programului
EXAMPLE	Ofera exemple de utilizare
SEE ALSO	O lista de manpage-uri care au legatura cu cel curent

Consultati Anexa 1 pentru un exemplu de manpage.

1.4.1.2. Utilizare

Sintaxa comenzii **man** este urmatoarea:

```
man optiuni nume_manpage
```

Spre exemplu, o introducere in sistemul de operare Linux poate fi obtinuta cu comanda **man intro**.

Numele de manpage poate fi o comanda, un nume de functie, un fisier de configurare sau orice altceva care are un manpage prezent in sistem.

Evidentiem urmatoarele optiuni de interes:

- **-k sablon_nume** (keyword) - cautarea unui cuvânt cheie în numele și descrierea scurtă a manpage-urilor disponibile; echivalent cu a scrie *apropos nume_manpage*. Numele căutat poate fi unul exact sau un fragment; atunci când se dorește căutare după nume exact, acesta trebuie încadrat între ^ și \$ (vezi exemplele de mai jos). Dacă numele este . (punct) vor fi afișate toate manpage-urile disponibile
- **-s sectiune** – cautarea într-o anumită secțiune, utilă în cazul în care există pagini cu același nume în secțiuni diferite
- **-a (all)** - atunci când un nume de manpage se regăsește în mai multe secțiuni, această opțiune va determina afișarea pe rând a tuturor manpage-urilor disponibile cu acel nume

```
# toate manpage-urile ale caror nume sau descriere scurta contin cuvantul secventa "xorg"
student@server:~$ man -k xorg
mousedrv (4)          - Xorg mouse input driver
Xorg (1)              - X11R7 X server
xorg.conf (5)        - configuration files for Xorg X server
xorg.conf.d (5)      - configuration files for Xorg X server
XOrgFoundation (7)   - X.Org Foundation information
xvidtune (1)         - video mode tuner for Xorg
# manpage-urile numite xorg, cu exactitate
student@server:~$ man -k ^xorg$
Xorg (1)              - X11R7 X server
# passwd are mai multe manpage-uri, in diferite sectiuni
student@server:~$ man -k ^passwd$
passwd (1)            - change user password
passwd (1ssl)         - compute password hashes
passwd (5)            - the password file
# vizualizarea tuturor manpage-urilor passwd
student@server:~$ man -a passwd
# vizualizarea manpage-ului passwd din sectiunea 5
student@server:~$ man -s 5 passwd
# listarea tuturor manpage-urilor disponibile
student@server:~$ man -k .
# vizualizarea tuturor manpage-urilor disponibile in sectiunea 1
student@server:~$ man -k -s 1 .
```

Nota: în exemplele de mai sus, `man -k` poate fi înlocuit cu comanda **apropos**. Astfel, spre exemplu, lista de manpage-uri ce conțin `xorg` poate fi obținută cu **apropos xorg**

Odată deschis un manpage, utilizatorul poate folosi următoarele shortcut-uri:

- săgețile sau tastele PageUp/PageDown pentru a parcurge conținutul
- ieșirea din manpage se realizează apăsând **q**
- căutarea unui șir de caractere se face apăsând `/sir_cautat` urmat de ENTER; aceasta va avea ca efect poziționarea automată pe prima apariție a șirului căutat. Se poate apoi merge către următoarea apariție apăsând **n** sau către precedenta apăsând **N**

Nota: pentru afișarea controlată a informației, **man** folosește utilitarul `less`, care va fi prezentat într-un capitol ulterior. Shortcut-urile menționate aparțin de fapt de `less`

1.4.2. Help despre comenzile built-in de shell

Informațiile despre comenzile recunoscute de către shell-ul însuși se găsesc fie în manpage-ul shell-ului (ex: `man bash`), fie folosind comanda **help** urmată de numele comenzii built-in dorite:

```
student@server:~$ help alias
alias: alias [-p] [name[=value] ... ]
       Define or display aliases.

       Without arguments, `alias' prints the list of aliases in the reusable
       form `alias NAME=VALUE' on standard output.

       Otherwise, an alias is defined for each NAME whose VALUE is given.
       A trailing space in VALUE causes the next word to be checked for
       alias substitution when the alias is expanded.

Options:
  -p      Print all defined aliases in a reusable format

Exit Status:
alias returns true unless a NAME is supplied for which no alias has been
defined.
```

1.5. Structura logica a sistemului de fisiere Linux/Unix

1.5.1. Caracteristici generale

Sistemul de fisiere Linux/Unix se prezinta utilizatorului sub forma unui arbore de fisiere si directoare, asemanator pana la un punct (ca principiu) cu cel din sistemele de operare Windows. Exista insa cateva diferente notabile care trebuie mentionate de la bun inceput:

- in Windows, atunci cand pe hard-disk sunt definite mai multe partitii - fiecare dintre ele continand, dupa cum se stie, propriul sistem de fisiere - utilizatorul lucreaza cu mai multi arbori distincti (denumiti C:, D: etc), fiecare dintre ei avand propria radacina notata cu \ (de aici C:\, D:\ etc). In Linux/UNIX exista intotdeauna un arbore unic de fisiere, chiar si atunci cand sistemul de fisiere este distribuit pe mai multe partitii (cum anume se realizeaza acest lucru se va explica intr-un capitol ulterior)
- radacina sistemului de fisiere Linux/Unix este notata "/" si este unica
- separatorul elementelor unei cai in sistemul de fisiere Linux/Unix este / (in Windows el era \). Calea `c:\www\index.html` din Windows s-ar scrie in Linux `/www/index.html`.
- denumirile fisierelor pot contine . (punct) pe orice pozitie. Din acest motiv, in Linux/Unix exista multe fisiere atipice pentru Windows (spre exemplu, majoritatea executabilelor nici nu au extensie) sau fisiere care nici nu ar putea fi create in Windows (ex: directorul `.ssh`)
- **se face distinctie intre literele mici si mari**, in aproape toate cazurile! Acest lucru se aplica comenzilor, optiunilor, numelor de fisiere etc. Iata cateva implicatii:
 - in acelasi director pot coexista fisiere al caror nume difera doar prin marimea literelor (ex: fisierele `Nume2`, `nume2` si `nuMe2`)
 - optiunile `-a` si `-A` sunt considerate distincte – nu pot fi folosite interschimbabil, deoarece pot avea efecte mult diferite
 - numele unei comenzi trebuie dat exact, tinand cont de litere mici si mari. Exemplu: comanda `ls` va functiona, pe cand comanda `LS` va genera o eroare
- totul in Linux/Unix este fisier – inclusiv consolele virtuale, dispozitivele hardware detectate in sistem etc. sunt prezente sub forma de fisiere. Fisierele pot fi insa de diferite tipuri (vezi sectiunea urmatoare)

1.5.2. Tipuri de fisiere Linux/Unix

Sistemul de operare Linux/Unix expune toate resursele sale sub forma de fisiere – si daca pentru fisierele obisnuite (documente, poze, muzica etc) acest lucru nu ne mira, lucrurile se schimba cand descoperim placa de

sunet, unitatea optica CD/DVD, hard-disk-ul, conexiunile de retea sub forma de fisiere si – mai mult – putem citi din sau scrie in aceste fisiere! Ca exemplu, o copie de partitie in Linux se poate reduce la a citi dintr-un fisier si a scrie in altul (fiecare dintre cele doua fisiere corespunzand cate unei partitii de pe hard disk).

Chiar daca totul in Linux/Unix este fisier, exista categorii de fisiere, dupa cum urmeaza:

- **fișiere standard** – pot fi text, date, executabile sau orice alt fisier al carui nume are in spate informatie folositoare utilizatorului stocata pe hard disk
- **directoare** – reprezinta “containere” in care sunt grupate fisiere. Desi spunem ca un director “contine” fisiere, de fapt pe hard-disk el nu ocupa un spatiu egal cu suma dimensiunilor fisierelor componente; fisierul de tip director cuprinde numai lista de fisiere continute, directorul fiind artificiul necesar pentru crearea structurii ierarhice a sistemului de fisiere
- **symbolic link-uri** (symlink) – reprezinta “shortcut-uri” catre alte fisiere sau directoare. Spre deosebire de shortcut-ul de Windows, symlink-ul de Linux/Unix nu este gandit doar pentru double-click – de fapt, nici nu are legatura cu interfata grafica – ci este o redirectionare catre un alt fisier care functioneaza inclusiv din linia de comanda
- **fișiere speciale** - sunt fisiere care nu au continut propriu, ci reprezinta o modalitate de a accesa un dispozitiv hardware, o conexiune de retea etc. Exista mai multe tipuri de fisiere speciale:
 - fisiere dispozitiv (“device files”) – toate dispozitivele hardware (“device”) in UNIX sunt vazute ca fisiere, comunicatia cu ele realizandu-se prin accesarea fisierului corespunzator. Fiecarui dispozitiv i se asociaza un *major number* – care ii spune kernelului ce driver sa foloseasca, si un *minor number*, ce reprezinta un indicativ al dispozitivului fizic (sau modalitatea de discriminare intre mai multe dispozitive fizice controlabile cu acelasi driver). Exista doua tipuri de fisiere de acest fel:
 - fisiere dispozitiv de tip **caracter** (“character device”) - folosite pentru accesarea directa, byte-cu-byte a dispozitivelor hardware. Exemplu: consolele virtuale
 - fisiere dispozitiv de tip **bloc** (“block device”) – utilizate pentru accesul prin intermediul unui buffer (in blocuri de date) al dispozitivelor. Exemplu: hard-disk-ul
 - fisiere de tip **named pipe** – intalnite in UNIX-urile tip AT&T; reprezinta o cale de comunicatie bidirectionala intre doua procese, prin intermediul unei cozi FIFO (First In First Out)
 - fisiere de tip **socket** – un “frate” al pipe-ului, dar introdus de BSD; mai este numit “UNIX domain socket” sau “local domain socket” – pentru a fi deosebit de socket-urile TCP, care permit comunicatia intre procese aflate la distanta unul de altul, pe computere conectate intr-un internet TCP/IP. Socket-ul poate fi folosit pentru scriere/citire numai de catre procesul care l-a deschis, spre deosebire de un named pipe care poate fi creat de catre administratorul sistemului pentru a interconecta doua procese ulterior crearii sale.

1.5.3. Directoarele create la instalare si rolul acestora

Asa cum in Windows, imediat dupa instalare, gasim in radacina partitiei C: cateva directoare “standard” cu roluri bine stabilite (Windows, Program Files, Documents and Settings sau Users etc), in acelasi fel in radacina (unica!) a arborelui de fisiere Linux/Unix exista un ansamblu de directoare, mai numeroase, ce servesc fiecare cate unui scop. Iata-le in continuare pe cele principale:

- **/home** – contine directoarele personale ale utilizatorilor (echivalent cu *Documents and Settings* sau *Users* din Windows)
- **/root** – directorul personal al utilizatorului *root* (administratorul sistemului)
- **/dev** – contine fisiere speciale corespunzatoare dispozitivelor din sistem (hard-disk-uri, placa de sunet etc)

- **/etc** – contine fisierele de configurare ale sistemului de operare si aplicatiilor. Este echivalent cu registry-ul din Windows, cu deosebirea ca aproape toate fisierele din /etc sunt text si pot fi usor modificate cu orice editor de text
- **/tmp** – director pentru fisiere temporare, cu permisiuni de scriere/citire pentru oricine. In unele distributii el este sters automat la reboot, de aceea nu se recomanda stocarea de informatii ce se doresc pastrate pe termen lung in acest director
- **/opt** – software instalat suplimentar, ce nu face parte din sistemul de operare ("3rd party software")
- **/boot** – contine un minim de fisiere necesare bootarii sistemului de operare – o parte a bootloader-ului/boot manager-ului si kernelul/kernel-urile
- **/proc** – contine un pseudo-sistem de fisiere, care de fapt nu exista pe hard-disk. Toate resursele prezente dedesubtul lui /proc reprezinta informatii din kernel pe care acesta le prezinta utilizatorului sub forma de fisiere. Astfel, utilizatorul poate avea acces la o multitudine de informatii legate de hardware-ul detectat, procesele care ruleaza, setarile de retea etc. Unele dintre aceste "fisier" sunt modificabile, efectul scrierii intr-un astfel de fisier fiind modificarea unui parametru de kernel
- **/var** – date cu caracter variabil – fisiere cu durata de viata mica sau al caror continut se modifica des. Cateva exemplu:
 - subdirectorul **mail** - contine casutele de mail in cazul in care pe statia respectiva ruleaza un server e-mail
 - subdirectorul **log** - contine fisierele jurnal, cele in care sistemul de operare si aplicatiile isi consemneaza erorile si informatiile produse in timpul functionarii
- **/bin** – contine executabile ("binaries") de uz general – cele strict necesare utilizarii si depanarii sistemului de operare, chiar si cand acesta lucreaza in mod avarie
- **/sbin** – contine executabile ce constituie comenzi pentru administrarea sistemului (inaccesibile in general utilizatorilor neprivilegiati)
- **/lib** – contine biblioteci partajate de sistem si module de kernel, necesare pentru bootarea sistemului si pentru rularea comenzilor din /bin si /sbin
- **/mnt** – folosit pentru montarea temporara a altor sisteme de fisiere
- **/usr** – locul unde sunt instalate aplicatiile destinate utilizatorilor (echivalent cu Program Files din Windows). In acest director se regaseste o parte a structurii de directoare din radacina, deoarece si aici informatiile sunt structurate:
 - **bin, lib** – comenzile disponibile utilizatorilor si bibliotecile de care acestea depind
 - **sbin** – utilitare de administrare pentru root care nu sunt critice pentru o minima administrare a sistemului (cele critice stau in /sbin)
 - **local** – folosit pentru instalarea de soft disponibil doar pe masina locala (in conditiile in care /usr ar fi preluat din retea, read-only)
 - **share** – date independente de arhitectura si care nu se modifica (ex: manpage-uri)

Desi pot exista diferente de la o distributie Linux la alta, in general se incearca aderarea la aceleasi standarde. Pentru sistemele Linux exista **FHS** (Filesystem Hierarchy Standard), disponibil la <http://www.pathname.com/fhs/>, ce specifica modalitatea (si ratiunile) de structurare a sistemului de fisiere Linux. FHS este parte a unei specificatii mai largi (LSB – Linux Standard Base) ce stabileste un set minim de criterii ce trebuie indeplinite de catre distributiile Linux astfel incat o aplicatie sa poate rula (in forma sa nemodificata) pe orice sistem compatibil LSB.

1.6. BIBLIOGRAFIE

- <http://www.linuxnix.com/2010/02/file-types-in-linux.html>
- <http://man7.org/linux/man-pages/man1/man.1.html> (sau comanda *man intro*)
- <http://man7.org/linux/man-pages/man7/man-pages.7.html> (sau comanda *man man-pages*)
- <http://man7.org/linux/man-pages/man1/man.1.html> (sau comanda *man man*)

InfoAcademy

1.7. ANEXA 1 - Exemplu de manpage

Prezentam in continuare manpage-ul comenzii **pwd** din sectiunea 1.

```
PWD(1)                                User Commands                                PWD(1)

NAME
    pwd - print name of current/working directory

SYNOPSIS
    pwd [OPTION]...

DESCRIPTION
    Print the full filename of the current working directory.

    -L, --logical
        use PWD from environment, even if it contains symlinks

    -P, --physical
        avoid all symlinks

    --help display this help and exit

    --version
        output version information and exit

NOTE: your shell may have its own version of pwd, which usually super-
sedes the version described here. Please refer to your shell's docu-
mentation for details about the options it supports.

AUTHOR
    Written by Jim Meyering.

REPORTING BUGS
    Report pwd bugs to bug-coreutils@gnu.org
    GNU coreutils home page: <http://www.gnu.org/software/coreutils/>
    General help using GNU software: <http://www.gnu.org/gethelp/>
    Report pwd translation bugs to <http://translationproject.org/team/>

COPYRIGHT
    Copyright © 2013 Free Software Foundation, Inc. License GPLv3+: GNU
    GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
    This is free software: you are free to change and redistribute it.
    There is NO WARRANTY, to the extent permitted by law.

SEE ALSO
    getcwd(3)

    The full documentation for pwd is maintained as a Texinfo manual. If
    the info and pwd programs are properly installed at your site, the com-
    mand

        info coreutils 'pwd invocation'

    should give you access to the complete manual.

GNU coreutils 8.21                    January 2015                                PWD(1)
```