

# ANEXA B

## COMENZI UNIX UZUALE

### B.1 GENERALITĂȚI

Sistemul UNIX lucrează în mod linie de comandă.

Accesarea unui sistem UNIX se face pe baza **numelui de utilizator** și a **parolei**:

login: nume\_utilizator <Enter>

password:

Parola va avea minimum șase caractere alfanumerice, cu minimum două alfabetice și minimum unul nealfabetic în primele șase.

După introducerea corectă a acestora și validarea lor, apare **prompterul de sistem**:

\$ pentru Bourne Shell și Korn Shell

% pentru C Shell

# pentru contul administratorului de sistem.

Shell-ul emite prompterul și interpretează comenzile.

Prima fază de interpretare a unei comenzi în sistemul UNIX constă în recunoașterea numelui comenzii sau generarea unui mesaj de eroare referitor de sintaxă incorectă.

Shell-urile Bourne și Korn admit introducerea comenzilor pe mai multe linii, caz în care pe următoarele linii apare prompterul: >.



*Exemplul 4.* Definierea secvenței (Ctrl-H) de ștergere a caracterelor, se face cu comanda:

```
$ stty erase "^H" <Enter>
```

*Exemplul 5.* Secvența (Ctrl-c) de întrerupere a execuției unui program, se definește astfel:

```
$ stty intr "^C" <Enter>
```

*Exemplul 6.* Comanda de afișare a unui mesaj pe ecran are obligatoriu ca argumente șirul de caractere care vor fi reprezentate, eventual încadrate cu apostroafe dacă lungimea mesajului depășește lungimea liniei de pe ecran:

```
$ banner ' BINE ATI <Enter>
```

```
> VENIT' <Enter>
```

Se observă apariția prompterului > pentru continuarea liniei anterioare.

Alte comenzi cu caracter general sunt:

who	comanda de aflare a informațiilor despre utilizatorii activi la un moment dat.
who am i	afișează numele utilizatorului și informații despre terminal.
whoami	afișează numai numele utilizatorului activ la acel moment, pe terminal.
man [-k n] termen	afișează informațiile conținute în manualul de sistem, referitoare la un cuvânt sau o comandă specificată în argumentul "termen", cu opțiunea -k pentru a fișă toate intrările ce conțin acel termen, sau cu specificarea numărului secțiunii dorite din manual (n).
whereis termen	comanda de deducere a secțiunilor care fac referiri la termenul dorit.
mail	comanda de citire a mesajelor din căsuța poștală proprie.
mail [destinatar]	comanda de expediere a unui mesaj prin sistemul de poștă electronică (Expedierea este marcată de combinația de taste ^d).
write [destinatar]	comandă de scriere către un alt utilizator, activ la același calculator.
mesg [y n]	comanda de acordare sau invalidare a permisiunii de scriere la acel terminal (fără opțiuni, afișează setarea anterioară).

### **Observații:**

Tasta <Space> determină afișarea următoarei pagini cu informații utile.

Tasta <Enter> permite afișarea următoarei linii.

Cu tasta <q> (*Quit*) se iese din manual.

Afișarea mesajelor se face pe principiul LIFO (*Last-In First-Out*).

Comanda `mail` își introduce propriul prompter (?) care poate fi urmat de una din următoarele comenzi:

?	se solicită informații ajutătoare ( <i>help</i> ).
d	șterge mesajul afișat din cutia poștală ( <i>delete</i> ).
s [fișier]	salvează mesajul în fișierul specificat ( <i>save</i> ).
<Enter>	determină afișarea următorului mesaj.
+	idem.
-	afișează mesajul anterior.
![comanda]	execută o altă comandă UNIX.
q	ieșire din cutia-poștală cu eventuale modificări ( <i>quit</i> ).
x	ieșire din cutia-poștală fără a schimba conținutul acesteia ( <i>exit</i> ).

## B.2 ORGANIZAREA SISTEMULUI DE FIȘIERE SUB UNIX

Sistemul de fișiere într-un sistem UNIX este organizat ierarhic, sub forma unei diagrame de tip 'arbore', cu un nod-rădăcină (notat: /), cu directoare, subdirectoare și fișiere (Fig.B.1).

Pe primul nivel se găsesc următoarele directoare:

/bin	conține fișierele de execuție a comenzilor pentru operațiile de bază în sistemul de operare și manipulare de fișiere.
/dev	conține driverele componentelor hardware din sistem.
/etc	cuprinde programele accesibile numai administratorului de sistem.
/ux	reprezintă directorul-sistem, care conține sistemul de operare UNIX.
/usr	include comenzi uzuale, altele față de cele incluse în directorul /bin.
/tmp	conține fișiere temporare.
/users	include subdirectoarele de lucru și conturile utilizatorilor.

Structura se ramifică în subdirectoare:

/usr/bin	cuprinde comenzi suplimentare, care nu sunt absolut necesare sistemului.
/usr/mail	include ca subdirectoare "cutiile poștale" ( <i>mailboxes</i> ) ale utilizatorilor.
/usr/local/bin	conține aplicațiile dezvoltate local.
/usr/man	reprezintă manualul cu informații ajutătoare pentru utilizatori.
/users/user1	este directorul creat de către administrator pentru utilizatorul 1.
/users/user2	este directorul creat de către administrator pentru utilizatorul 2 ș.a.m.d.

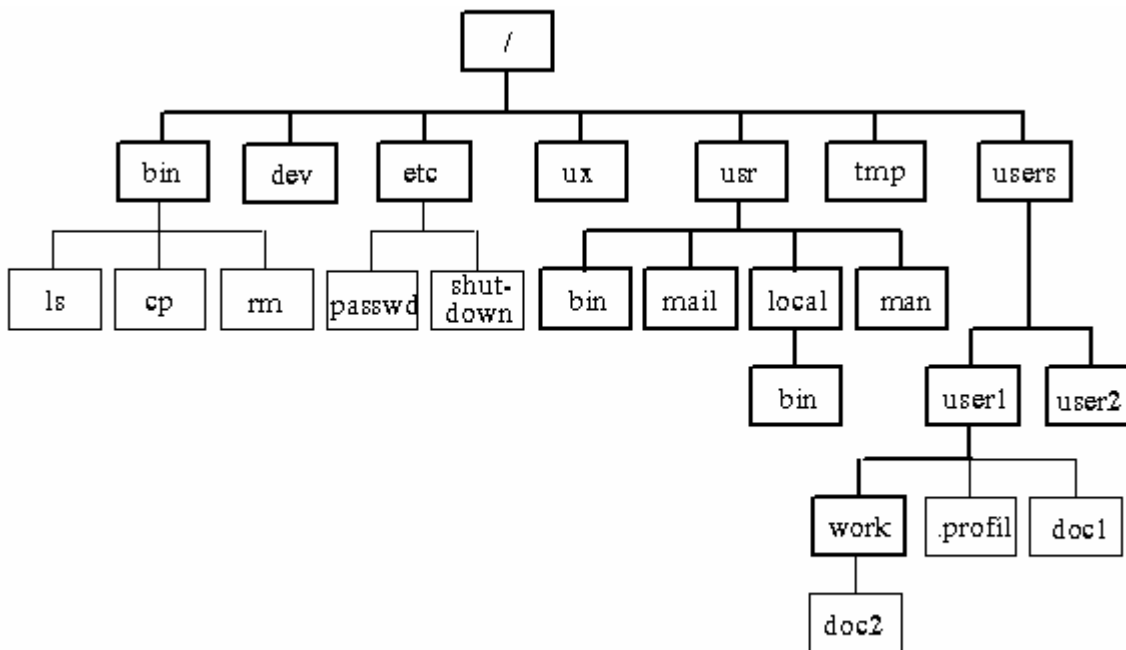


Fig.B.1 Structură de sistem de fișiere sub UNIX

**Observație:**

La crearea unui director, se definesc automat două directoare, notate "." și "..", pentru exprimarea simplificată a căilor relative în graf:

- reprezintă directorul curent.
- .. semnifică nodul ierarhic superior, adică directorul "părinte".

*Exemplul 7.* Din nodul "user1" se definesc următoarele căi cu semnificațiile specificate:

- |          |               |                   |
|----------|---------------|-------------------|
| ·        | echivalent cu | /users/user1      |
| ..       | echivalent cu | /users            |
| ../..    | echivalent cu | /                 |
| ./work   | echivalent cu | /users/user1/work |
| ../user2 | echivalent cu | /users/user2      |
| ../usr   | echivalent cu | /usr              |

**B.3 COMENZI DE OPERARE CU DIRECTOARELE**

- pwd** comanda de determinare a directorului curent (*present working directory*), exprimă calea absolută din graf spre directorul curent.

- cd [director]** comanda de schimbare a directorului curent (*change directory*) în cel exprimat în mod relativ sau absolut ca argument al comenzii.  
Fără argumente, determină întoarcerea în directorul de login (*home directory*).
- mkdir [-p] [director]** comandă de creare a directorului (*make directory*), eventual a celor intermediare dacă nu există (se include opțiunea -p). E admisă crearea simultană a mai multor directoare. Se exprimă ca argument fie calea absolută, fie una relativă pentru fiecare director nou creat.
- rmdir [director]** comandă de ștergere a directorului (*remove directory*) cu condiția să nu conțină fișiere sau subdirectoare. E admisă ștergerea simultană a mai multor directoare.
- ls [optiuni] [cale]** comanda de listare (*list*) a directoarelor și fișierelor subordonate nodului exprimat prin argumentul "cale", relativ sau absolut. Opțiuni:
- a listează toate (*all*) subdirectoarele și fișierele, inclusiv cele de configurare, cu prefixul "."
  - d listează caracteristicile directorului.
  - l listează atributele tuturor fișierelor incluse în director.
  - F listează formatat conținutul directorului (se specifică sufixul / în cazul subdirectoarelor și sufixul \* pentru fișierele executabile).
  - R listează recursiv toate fișierele din directorul curent și din subdirectoarele subordonate acestuia.

**Observație:**

Unele sisteme UNIX admit folosirea comenzilor sub formă prescurtată. Orice comandă se execută numai după acționarea tastei <Enter>.

*Exemplul 8.* Comanda de listare formatată a directoarelor și fișierelor subordonate nodului 'rădăcină' se dă sub forma:

```
$ ls -F<Enter>
```

sau

```
$ lsf <Enter>
```

*Exemplul 9.* Se aplică aceste comenzi pentru nodul "User2". Observați efectele lor.

```
$ pwd
```

```
/users/user2
$ ls .. /user1
work      doc1
$ ls -aF /users/user1
work/     .profil   doc1
$ ls ..
user1     user2
$ ls -F ..
user1/    user2/
$ ls -R ..
user1     user2
./user1:
work      doc1
$ ls user1
user1 not found
$ cd ..
$ pwd
/users
$ cd ..
/
$ cd ..
/
$ cd /usr
$ pwd
/usr
$ ls -F .
bin/      mail/     local/    man/
$ mkdir proba
$ pwd
/usr
$ cd ./proba
$ mkdir dir1 dir2
$ ls -F
```

```

dir1/      dir2/
$ cd ..
$ rmdir proba
rmdir: proba not empty
$ rmdir    proba/dir1  proba/dir2  proba
$ cd ../bin
$ ls -F
ls*  cp*  rm*
```

## B.4 COMENZI DE OPERARE CU FIȘIERELE

Prin **fișier** (*file*) se înțelege o colecție de date, de natură diversă: text, imagine, driver, executabil, director.

Cu un editor de texte, se introduc caractere ASCII și se obține un **fișier obișnuit** (*regular file*).

Dacă se scrie un program folosind un limbaj de programare (C++, Turbo C, Java) se obține un cod-sursă care prin compilare devine **fișier executabil** (*program file*), stocat într-un director de tip /bin.

Mai multe fișiere sunt stocate într-un **director** (*directory*), care reprezintă un fișier special, conținând numele și adresele de memorie ale componentelor, precum și numerele de identificare ale acestora în sistemul de fișiere (*inode number*).

Pentru componentele hardware se folosesc fișiere de tip 'driver', de interfațare cu sistemul de operare (*kernel*), denumite și **fișiere periferice** (*device file*), stocate în directorul /dev.

**Numele** fișierului are maximum 255 de caractere alfanumerice și de punctuație: 'punct' (.), 'cratimă' (-), 'liniuță de subliniere' ( \_ ). În unele variante de UNIX această limită are valori mai mici, de exemplu, 14 în HP-UX.

Ca răspuns la comanda **ls -l**, **tipul** fișierului se specifică pe prima poziție, astfel:

- fișier obișnuit
- d director
- l fișier de legătură
- n fișier de rețea
- c driver pentru un periferic de afișare a caracterelor
- b driver pentru un periferic care lucrează cu blocuri de date



`p` driver pentru un canal de comunicație interproces (pipeline).

Ca atribute ale unui fișier se mai specifică, în afară de tipul acestuia, drepturile de acces (citire, scriere), număr de noduri subordonate, nodul ierarhic superior, dimensiunea fișierului exprimată în octeți, data și ora când a fost accesat ultima oară și în final, numele fișierului.

Afișarea pe ecran a conținutului unui fișier se poate face fie în întregime, prin comanda `cat`, având obligatoriu ca argument numele fișierului, fie segmentat, pagină cu pagină, folosind comanda `more`.

Dacă se dorește **afișarea ultimelor n linii** dintr-un fișier, atunci se dă comanda:

```
tail [-n] [nume fișier]
```

Alte comenzi de operare cu fișierele sunt următoarele.

**Comanda de copiere:**

```
cp [-optiuni] [fișier1] [director|fișier2]
```

Se copie fișierul 1 într-un nou fișier 2, creat în același director dacă nu se precizează alt director, sau într-un fișier cu același nume (`fișier1`) dar în directorul specificat de cel de-al doilea argument al comenzii, eventual sub un alt nume (`fișier2`) dacă este dat în comandă.

Primul argument poate fi multiplu, cu condiția ca destinația să fie un director.

Opțiunile comenzii pot fi:

- `i` de operare interactivă, pentru a se cere acordul în cazul suprascrierii unui fișier deja existent.
- `r` de operare recursivă, în cazul în care dorim să copiem în întregime un director în alt director.

**Mutarea sau redenumirea** unui fișier (director) se face cu comanda:

```
mv [-optiuni] [director|fișier1] [director|fișier2]
```

Și în acest caz, ca argumente se folosesc căi absolute sau relative ale unor fișiere sau directoare.

Opțiunea de operare interactivă - `i` ne avertizează în cazul în care există deja la destinație un fișier cu același nume ce urmează a fi rescris.

Comanda de mutare a fișierelor diferă de cea de copiere prin faptul că nu se creează fișiere noi, numărul total de fișiere nu se modifică.

**Comanda de ștergere** a unui fișier sau a conținutului unui director este:

```
rm [-optiuni] [director|fișier]
```

În acest caz, este indicată folosirea opțiunii de operare interactivă - `i`.

Ștergerea tuturor fișierelor dintr-un director este posibilă prin includerea în linia de comandă a opțiunii de operare recursivă - r, fiind recomandată exprimarea ambelor opțiuni - ir. Spre deosebire de comanda rmdir în acest caz, directorul există în continuare, dar nu mai conține nimic.

Opțiunea - f determină ștergerea necondiționată, fără avertismente a tuturor fișierelor specificate în lista de argumente.

*Exemplul 10.* Se consideră nodul curent user1. Observați efectele următoarelor comenzi:

```
$ pwd
```

```
/users/user1
```

```
$ ls -F
```

```
work/      doc1
```

```
$ ls work
```

```
doc2
```

```
$ cp doc1 work
```

```
$ ls work
```

```
doc1 doc2
```

```
$ cp doc1 doc3
```

```
$ ls -F
```

```
work/      doc1      doc3
```

```
$ cat doc1
```

Acesta este un fisier de test.

Se foloseste pentru exemplificarea comenzilor.

```
$ cat doc3
```

Acesta este un fisier de test.

Se foloseste pentru exemplificarea comenzilor.

```
$ tail -1 doc3
```

Se foloseste pentru exemplificarea comenzilor.

```
$ ls .. /user2
```

```
ls: user2 is empty
```

```
$ cp doc1 doc3 .. /user2
```

```
$ ls .. /user2
```

```
doc1 doc3
```

```
$ cp -ri work .. /user2
$ cd .. /user2
$ ls -FR
work/      doc1      doc3
./work:
doc1      doc2
$ mv doc3 work
$ ls -F
work/      doc1
$ ls work
doc1      doc2      doc3
$ mv work test
$ ls -F
test/      doc1
$ ls test
doc1      doc2      doc3
$ rm -i doc1
doc1? y
$ cat doc1
cat: cannot open doc1
$ ls -F
test/
$ rm test
rm: test directory
$ rm -r test
$ ls test
test is empty
$ ls -F
test/
$ rmdir test
$ ls
user2 is empty
$ cd ..
```

```
$ mv user1 user2
```

```
$ ls user1
```

```
user1 is empty
```

```
$ ls user2
```

```
work      doc1
```

***Observație:***

În cazul operării interactive, utilizatorul poate răspunde afirmativ, tastând y (*yes*), ori negativ, prin n (*no*) sau <Enter>.