

Instrumente și tehnici de bază în informatică

Curs 1

Horațiu Cheval

Sistemul de fișiere

- Organizat ierarhic într-o structură arborescentă
- Directoarele conțin fișiere sau alte directoare
- UNIX: un singur nod rădăcină numit *root* (notat prin /); separate prin /
- Windows: pot exista mai multe rădăcini (C:, D:, etc.); separate prin \

Linia de comandă

- modul cel mai elementar prin care interacționăm cu sistemul de operare.

- comandă → răspuns

```
$ echo "Hello world"
```

```
Hello world
```

```
$date
```

```
vineri 3 octombrie 2025, 08:36:08 +0300
```

- Terminale: GNOME terminal, konsole, etc.
- Shell-uri: sh, **bash**, zsh

Comenzi esențiale:

- `pwd(1)`: afișează directorul curent
- `cd(1)`: schimbă directorul curent
- `ls(1)`: afișează conținutul directorului curent

Exemplu:

- `pwd(1)`: afișează directorul curent
`/home/horatiu/Documents$ pwd`
`/home/horatiu/Documents`

Exemplu:

- `cd(1)`: schimbă directorul curent

```
/home/horatiu/Documents$ cd itbi
```

```
/home/horatiu/Documents/itbi$
```

`..`: se referă mereu la directorul curent

`...`: se referă la directorul părinte

- `ls(1)`: afișează conținutul directorului curent

```
/home/horatiu/Documents$ ls
```

```
# toate fișierele din directorul Documents
```

Linia de comandă - navigarea

Orice fișier este identificat printr-o **cale (path)** către acesta. Calea poate fi:

- absolută (începând de la root), identificabilă prin /:
`/home/horatiu/Documents/itbi/2025/Curs/Curs1/curs1.pdf`
- relativă (la directorul curent):
`2025/Curs/Curs1/curs1.pdf`

Linia de comandă

De cele mai multe ori, pot primi un număr de **argumente** și **opțiuni (flag-uri)**

Putem gândi sintaxa generală a comenzilor shell sub forma

```
cmd -flag1 ... -flagM arg1 ... argN
```

De obicei, ordinea flag-urilor nu contează.

Uneori, flag-urile reprezintă parametri care primesc valori (numere, fișiere, etc)

```
$ head -n 3 hello.c
```

```
# printeaza primele 3 randuri ale fisierului hello.c
```

```
$ gcc hello.c -o my_hello_exec
```

```
# compileaza programul hello.c,
```

```
# numind executabilul produs my_hello_exec
```

Manualul

- `man comanda` – afișează manualul pentru o comandă

Exemplu:

```
$ man ls
```

```
LS(1)
```

NAME

```
ls - list directory contents
```

SYNOPSIS

```
ls [OPTION]... [FILE]...
```

DESCRIPTION

List information about the FILES (the current directory if none are given).

Mandatory arguments to long options are mandatory for short options unless otherwise noted.

```
-a, --all
```


Variabile de mediu (environment variables)

Exemplu:

- `cd` (fără niciun argument) ne mută în directorul “home” al utilizatorului
- `firefox` va porni browser-ul

Q: Cum știe shell-ul care este directorul curent și unde se găsește executabilul?

Variabile de mediu (environment variables)

Exemplu:

- `cd` (fără niciun argument) ne mută în directorul “home” al utilizatorului
- `firefox` va porni browser-ul

Q: Cum știe shell-ul care este directorul curent și unde se găsește executabilul?

A: Environment variables

Variable de mediu (environment variables)

- Variabila \$HOME: stabilește directorul *home*

```
/home/horatiu/Documents$ echo $HOME  
/home/horatiu  
/home/horatiu/Documents$ cd  
/home/horatiu$
```

- Variabila \$PATH: o listă de directoare (separate prin :) în care shell-ul caută executabile aferente unei comenzi

```
$ which firefox  
/usr/bin/firefox  
$ echo $PATH  
/home/horatiu/.opam/default/bin:/home/horatiu/  
.nix-profile/bin ...  
# (o lista lunga de directoare intre care se afla /usr/bin)
```

Comenzi esențiale:

- `cp(1): cp src dst` copiază fișierul `src` în `dst`
- `mv(1): mv src dst` mută fișierul `src` în `dst`
- `rm(1): rm file` șterge fișierul `file`
- `touch(1): touch file` creează un fișier gol cu numele `file`
- `mkdir(1): mkdir dir` creează un director cu numele `dir`
- `rmdir(1): rmdir dir` șterge directorul `dir`, dacă acesta este gol

Redirecționări și pipe-uri

- > – redirecționează ieșirea standard într-un fișier
- >> – adaugă la sfârșitul unui fișier
- | – leagă ieșirea unei comenzi de intrarea alteia

Exemplu:

```
$ ls > list.txt
```

```
$ ls /usr/bin | less
```

grep(1): grep expr file1 ... fileN caută expr în fișierele argument.

Exemplu:

- toate liniile care conțin int x

```
$ grep -n "int x" main.c
```

- toate liniile care conțin int x, cu unul sau mai multe spații

```
$ grep -E "int[[:space:]]+x" main.c
```

opțiunea -E indică faptul că folosim o *expresie regulată* (vedeți man find pentru mai multe opțiuni).

`find(1)`: find path options caută în path fişierele care se potrivesc specificaţiei din options

Exemplu:

- toate fişierele cu extensia `.tex` din directorul curent

```
$ find . -name "*.tex"
```

- toate directoarele din `/usr`

```
$ find /usr -type d
```

- şterge toate fişierele cu extensia `.aux`

```
$ find . -type f -name "*.aux" -exec rm {} \;
```