

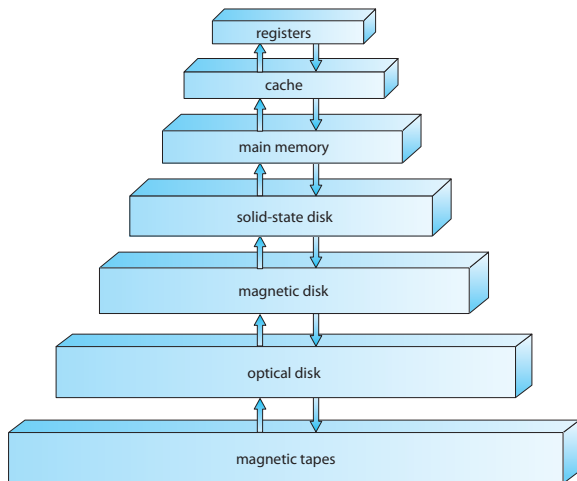
# Sistemul de fișiere

## Utilizarea Sistemelor de Operare

Paul Irofti

Universitatea din București  
Facultatea de Matematică și Informatică  
Department de Informatică  
Email: [paul.irofti@fmi.unibuc.ro](mailto:paul.irofti@fmi.unibuc.ro)

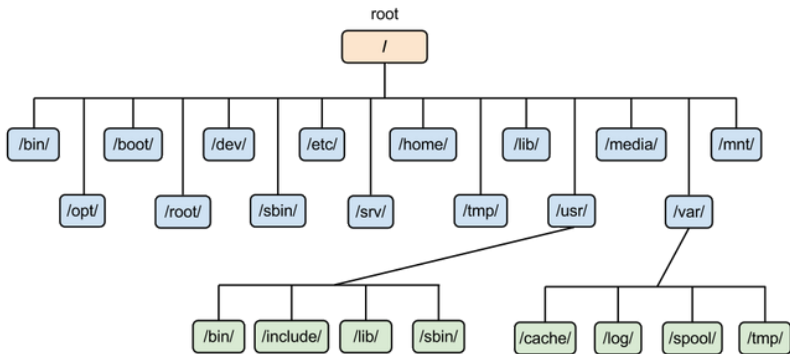
# Memoria



<http://codex.cs.yale.edu/avi/os-book/>

- ▶ stocare permanentă
- ▶ disponibile după întreruperea alimentării
- ▶ date folosite direct de utilizator
- ▶ salvate în memoria secundară (SSD → casete)

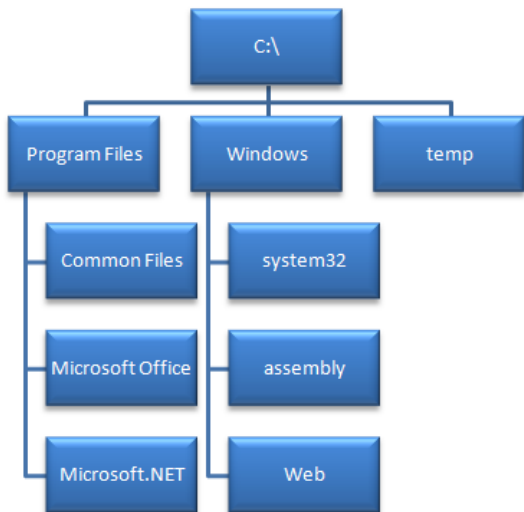
- ▶ un director conține fișiere
- ▶ informație organizată
- ▶ nu avem toate fișierele într-un loc (flat structure)
- ▶ un director poate conține alte directoare



<https://nepalिसupport.wordpress.com/>

- ▶ rădăcina este notată cu / și se mai numește și *root*.
- ▶ conține toate mediile de memorie
- ▶ fișierele în general nu au extensie
- ▶ separatorul folosit este /
- ▶ Exemplu: `/var/log/messages`

# Windows



<https://linuxway.weebly.com/>

- ▶ nu există o rădăcină unică
- ▶ fiecare entitate de stocare are rădăcina lui
- ▶ rădăcina este notată cu literă:
- ▶ indiferent de configurație tot timpul va exista C:
- ▶ C: conține toate fișierele necesare SO
- ▶ fișierele în general au extensie
- ▶ separatorul folosit este \ (invers decât Unix)
- ▶ versiunile mai noi de Windows acceptă și varianta Unix
- ▶ Exemplu: C:\Windows\System32



# Lina de comandă

- ▶ flexibilitate
- ▶ automatizare
- ▶ control la distanță
- ▶ dificil la început
- ▶ mod de lucru: comandă-răspuns
- ▶ are la bază un program de tip `shell`
- ▶ promptul indică faptul că se așteaptă o comandă
- ▶ simbolizat de obicei prin
  - ▶ \$ sau % pentru utilizatorii comuni
  - ▶ # pentru administrator (denumit `root` în Unix).

- ▶ comanda în dreapta promptului
- ▶ rezultatul afișat fără a fi prefixat cu prompt
- ▶ încheierea comenzii → reapariția promptului

## Example

```
$ echo " Hello , World!"  
Hello , World!  
$
```

- ▶ comandă necunoscută → citiți manualul
- ▶ manualul SO este împărțit în secțiuni
- ▶ intrări cu același nume dar în secțiuni diferite
- ▶ secțiuni
  1. general commands
  2. system calls and error numbers
  3. library functions
  4. device drivers

## Example

```
$ man echo
$ man printf
$ man 3 printf
$ man man
```

# Variabile de mediu

- ▶ dictează comportamentul programelor
- ▶ sunt setate de sistem sau utilizator
- ▶ scrise cu litere mari și precedate de simbolul \$
- ▶ la început terminalul vă plasează în directorul \$HOME
- ▶ terminalul caută comenzi în \$PATH
- ▶ pot fi setate sau modificate cu comanda export
- ▶ export este o comandă *internă* a shell-ului

- ▶ `pwd(1)` – print working directory
- ▶ `cd(1)` – change directory
- ▶ `ls(1)` – list directory contents
- ▶ `.` simbolizează directorul curent și `..` directorul părinte

## Example

```
$ cd ../../../../  
$ cd /home/paul  
$ cd $HOME  
$ cd
```

# Redirecționare

- ▶ operatorul `>` redirecționează ieșirea în modul *create*
- ▶ operatorul `>>` redirecționează ieșire în modul *append*
- ▶ operatorul `<` redirecționează intrarea
- ▶ operatorul `|` (denumit și *pipe*): ieșirea comenzii din stânga devine intrarea celei din dreapta

## Example

```
$ echo "lorem ipsum" > foo
```

```
$ echo "more lorem" >> foo
```

```
$ echo "less ipsum" > foo
```

Cum va arăta foo la fiecare pas?

- ▶ `cat(1)` – pentru fișiere text scurte
- ▶ `less(1)` – pentru fișiere text lungi
- ▶ `hexdump(1)` – pentru fișiere binare

## Example

```
$ cat foo
less ipsum
$ less /etc/passwd # exit with 'q'
$ which ls
/bin/ls
$ hexdump -C 'which ls' | less
```

- ▶  $\wedge$  – simbolizează începutul liniei
- ▶  $\$$  – simbolizează sfârșitul liniei
- ▶ un atom este un caracter sau un set de caractere grupat cu [ ] sau ( )
- ▶  $*$  – găsește de 0 sau mai multe ori atomul precedent
- ▶  $+$  – găsește de 1 sau mai multe ori atomul precedent
- ▶  $?$  – găsește de 0 sau 1 ori atomul precedent



## Căutarea în fișiere

- ▶ `grep(1)` – comanda cea mai folosită de căutare
- ▶ folosește tipare și expresii regulate
- ▶ uz tipic: `grep expresie fișiere`
- ▶ opțiuni utile
  - ▶ `-R` – căutare recursivă (intră în toate subdirectoarele)
  - ▶ `-i` – case insensitive
  - ▶ `-n` – indică numărul liniei pe care a fost găsită expresia

### Example

```
$ grep printf *.c  
$ grep printf *. [ch]
```

# Căutarea în directoare

- ▶ `find(1)` – comanda cea mai folosită de căutare
- ▶ uz tipic: `find cale -name expresie`
- ▶ opțiuni utile
  - ▶ `-exec` – execută comanda asupra elementelor găsite
  - ▶ `-type` – limitează rezultatele la un anumit tip (ex. fișiere)
  - ▶ `-iname` – la fel ca `-name` dar case insensitive
  - ▶ `-maxdepth` – nivelul de recursivitate

## Example

```
$ find / -name echo # foarte lent!  
$ find . -iname *.o -exec rm {} \;
```

- ▶ `mkdir(1)` – crează directoare
- ▶ `rmdir(1)` – șterge directoare fără conținut
- ▶ `touch(1)` – crează un fișier gol dacă nu există deja
- ▶ `mv(1)` – mută directoare sau fișiere
- ▶ `cp(1)` – copiază directoare (cu `-r`) sau fișiere
- ▶ `rm(1)` – șterge directoare (cu `-r`) sau fișiere

# Expresii regulate

- ▶ `[0-9]+ [a-z]+ [A-Z]+ [a-zA-Z0-9]+`
- ▶ `[:alnum:] [:cntrl:] [:lower:] [:space:]`
- ▶ `[:alpha:] [:digit:] [:print:] [:upper:]`
- ▶ am fosolit deja o expresie regulată? când?

## Example

```
$ egrep [a-z]+[0-9]+ *.tex
\usepackage[utf8]{inputenc}
\item ... Windows\textbackslash System32}
$ egrep '\([0-9]+' *.tex
\item \code{pwd(1)} -- print ...
\item \code{cd(1)} -- change ...
\item \code{ls(1)} -- list ...
\item \code{cat(1)} -- pentru ...
...
```