

## GHID DE UTILIZARE LINUX (III)

### Lucrul la linia de comandă în UNIX, partea a II-a: Comenzi de bază și sisteme de fișiere (cont.)

Cristian Vidrașcu

`cristian.vidrascu@info.uaic.ro`

Martie, 2025



# Sumar

- Introducere
- UNIX și utilizatorii
- Fișiere în sisteme UNIX (cont.)
- Alte categorii de comenzi simple
- Troubleshooting
- Referințe bibliografice

- Introducere
- UNIX și utilizatorii**
  - Conturi de utilizatori și grupuri de utilizatori
  - Comenzi referitoare la utilizatorii sistemului
- Fișiere în sisteme UNIX (cont.)**
  - Structura fizică a sistemelor de fișiere
  - Montarea sistemelor de fișiere
  - Protecția fișierelor, prin permisiuni de acces
  - Alte comenzi pentru lucrul cu fișiere (cont.)
- Alte categorii de comenzi simple**
  - Procese. Sistemul de procese
  - Comenzi ce oferă diverse informații
  - Alte categorii de comenzi
- Troubleshooting**
  - “Ghid de supraviețuire la linia de comandă”
- Referințe bibliografice**



# Introducere

[Introducere](#)

[UNIX și utilizatorii](#)

[Fișiere în sisteme UNIX \(cont.\)](#)

[Alte categorii de comenzi  
simple](#)

[Troubleshooting](#)

[Referințe bibliografice](#)

În prima parte a acestei prezentări am discutat despre:

- Principalele categorii de comenzi
  - Comenzi de *help*
  - Editoare de texte
  - Compilatoare, depanatoare, ș.a.
  - Comenzi pentru conectarea la distanță pe un server UNIX
- Fișiere în sisteme UNIX
  - Structura logică a sistemelor de fișiere
  - Comenzi de bază pentru lucrul cu fișiere (și directoare)
  - Alte comenzi pentru lucrul cu fișiere

\* \* \*

Acum vom continua prezentarea cu subiectele sumarizate pe *slide*-ul anterior.



# Agenda

Introducere

UNIX și utilizatorii

Conturi de utilizatori și grupuri de utilizatori  
Comenzi referitoare la utilizatorii sistemului

Fișiere în sisteme UNIX (cont.)

Alte categorii de comenzi simple

Troubleshooting

Referințe bibliografice

Introducere

**UNIX și utilizatorii**

Conturi de utilizatori și grupuri de utilizatori  
Comenzi referitoare la utilizatorii sistemului

**Fișiere în sisteme UNIX (cont.)**

Structura fizică a sistemelor de fișiere  
Montarea sistemelor de fișiere  
Protecția fișierelor, prin permisiuni de acces  
Alte comenzi pentru lucrul cu fișiere (cont.)

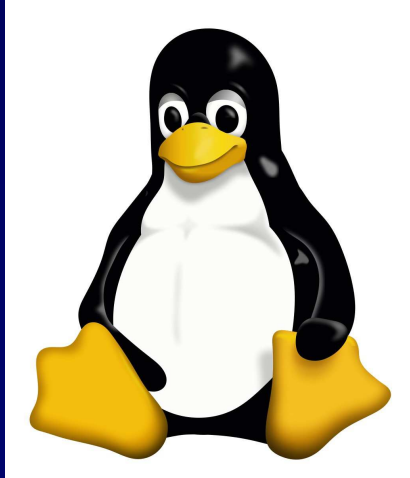
**Alte categorii de comenzi simple**

Procese. Sistemul de procese  
Comenzi ce oferă diverse informații  
Alte categorii de comenzi

**Troubleshooting**

“Ghid de supraviețuire la linia de comandă”

**Referințe bibliografice**



# Conturi de utilizatori și grupuri de utilizatori

Introducere

UNIX și utilizatorii

Conturi de utilizatori și grupuri de utilizatori

Comenzi referitoare la utilizatorii sistemului

Fișiere în sisteme UNIX (cont.)

Alte categorii de comenzi simple

Troubleshooting

Referințe bibliografice

- Fiecare **utilizator**, pentru a putea lucra pe un sistem UNIX, trebuie să posede un cont pe sistemul UNIX respectiv. Contul este identificat printr-un nume (*username*) și o parolă (*password*) asociată, ce trebuie furnizate în momentul conectării la sistem (*i.e.*, operația de autentificare în sistem, numită pe scurt “*login*”). De asemenea, fiecare cont de utilizator are asignat un UID (*User Identification*). Există un utilizator special, numit *root* (sau *superuser*), cu  $UID = 0$ . El are drepturi depline asupra întregului sistem, fiind responsabil cu administrarea sistemului.

“Baza de date” cu informații despre utilizatorii ce au conturi în sistem este stocată în fișierul `/etc/passwd` ([3]), ce conține linii de text conforme cu următorul șablon (format din 7 câmpuri cu informații, separate prin caracterul ‘:’) :

```
username : x : UID : GID : userdata : home directory : login shell
```

unde *UID* este ID-ul utilizatorului, *GID* este ID-ul grupului primar de care aparține, și *userdata* sunt datele personale ale utilizatorului (numai dacă au fost introduse de *superuser* la crearea contului).

*Notă:* acel **x** din al doilea câmp indică faptul că parola asociată contului este stocată, în formă criptată, în fișierul `/etc/shadow` (care nu este accesibil decât *superuser*-ului).



## Conturi de utilizatori și grupuri de utilizatori (cont.)

Introducere

UNIX și utilizatorii

Conturi de utilizatori și grupuri de utilizatori

Comenzi referitoare la utilizatorii sistemului

Fișiere în sisteme UNIX (cont.)

Alte categorii de comenzi simple

Troubleshooting

Referințe bibliografice

- Există **grupuri de utilizatori**, cu ajutorul cărora se gestionează mai ușor drepturile și restricțiile de acces ale utilizatorilor la resursele sistemului. Orice grup de utilizatori are asignat un nume (*groupname*) și un număr GID (*Group Identification*), precum și o parolă (*gpassword*) asociată, utilizabilă pentru “*membership* temporar”.

Fiecare utilizator face parte dintr-un grup primar (indicat de câmpul *GID* din `/etc/passwd`) și poate fi opțional afiliat la alte grupuri suplimentare (pe baza asocierilor din `/etc/group`).

“Baza de date” cu informații despre grupurile de utilizatori din sistem este stocată în fișierul `/etc/group` ([4]), ce conține linii de text conforme cu următorul șablon (format din 4 câmpuri cu informații, separate prin caracterul ‘:’) :

```
groupname : x : GID : list of users
```

unde *GID* este ID-ul grupului, iar ultimul câmp conține o listă, posibil vidă, de *username*-uri separate prin caracterul ‘,’ , prin care se “declară” afilierea respectivilor utilizatori la acel grup.

*Notă:* acel **x** din al doilea câmp indică faptul că parola asociată grupului este stocată, în formă criptată, în fișierul `/etc/gshadow` (care nu este accesibil decât *superuser*-ului).





# Comenzi referitoare la utilizatorii sistemului

Introducere

UNIX și utilizatorii

Conturi de utilizatori și grupuri de utilizatori

Comenzi referitoare la utilizatorii sistemului

Fișiere în sisteme UNIX (cont.)

Alte categorii de comenzi simple

Troubleshooting

Referințe bibliografice

- Comenzi de administrare a conturilor de utilizatori:
  - schimbarea datelor asociate unui cont de utilizator obișnuit (de către acesta): `passwd`, `chsh`, ș.a.
  - schimbarea datelor asociate unui cont de utilizator (doar de către *superuser*): `chage`, ș.a.
- Comenzi ce oferă diverse informații despre utilizatori:
  - aflarea de informații despre utilizatorii sistemului (*i.e.*, despre conturile de utilizator): `whoami`, `id`, `groups`
  - aflarea de informații despre utilizatorii conectați la sistem (și despre sesiunile lor de lucru pe sistem): `users`, `w`, `who`, `finger`, `last`

*Demo:* a se vedea exemplele `[Info user(s) #1]` și `[Info user(s) #2]` din suportul de laborator, disponibil [aici](#).



# Agenda

Introducere

UNIX și utilizatorii

Fișiere în sisteme UNIX (cont.)

Structura fizică a sistemelor de fișiere

Montarea sistemelor de fișiere

Protecția fișierelor, prin permisiuni de acces

Alte comenzi pentru lucrul cu fișiere (cont.)

Alte categorii de comenzi simple

Troubleshooting

Referințe bibliografice

Introducere

## UNIX și utilizatorii

Conturi de utilizatori și grupuri de utilizatori

Comenzi referitoare la utilizatorii sistemului

## Fișiere în sisteme UNIX (cont.)

Structura fizică a sistemelor de fișiere

Montarea sistemelor de fișiere

Protecția fișierelor, prin permisiuni de acces

Alte comenzi pentru lucrul cu fișiere (cont.)

## Alte categorii de comenzi simple

Procese. Sistemul de procese

Comenzi ce oferă diverse informații

Alte categorii de comenzi

## Troubleshooting

“Ghid de supraviețuire la linia de comandă”

## Referințe bibliografice





# Structura fizică a sistemelor de fișiere

Introducere

UNIX și utilizatorii

Fișiere în sisteme UNIX (cont.)

Structura fizică a sistemelor de fișiere

Montarea sistemelor de fișiere

Protecția fișierelor, prin permisiuni de acces

Alte comenzi pentru lucrul cu fișiere (cont.)

Alte categorii de comenzi simple

Troubleshooting

Referințe bibliografice

În sistemele de operare din familia UNIX, există mai multe “tipuri” de sisteme de fișiere, ce au toate o structură logică de tip arborescent, însă diferă între ele prin structura fizică a fiecăruia, adică prin structurile de date folosite pentru reprezentarea datelor (*i.e.*, a informațiilor despre fișierele stocate într-un sistem de fișiere de “tipul” respectiv) și prin formatul de stocare a lor pe disc.

Exemple de “tipuri” de sisteme de fișiere din Linux: `ext2fs`/`ext3fs`/`ext4fs`, `xfs`, `reiserfs`, `btrfs`, `zfs`, `vfat`, `ntfs`, ș.a. Pentru informații despre acestea consultați documentația `man 5 fs`. *Observație:* acestea sunt “tipuri” de sisteme de fișiere utilizate pentru stocarea fișierelor pe disc. Există însă și “tipuri” cu diverse roluri specializate, ce stochează informația nu pe disc, ci în memoria RAM volatilă sau într-o altă formă de stocare (*e.g.*, `tmpfs`, `procfs`, `devfs`, `sysfs`, `unionfs`, `squashfs`, ș.a.), sau sisteme de fișiere distribuite (*e.g.*, NFS, AFS, SMB/CIFS, ș.a.).

**Sistemul de fișiere “principal”:** într-un sistem UNIX avem un singur “disc logic”, numit *the root filesystem*, și referit prin rădăcina sa, cu numele “/” (spre deosebire de Windows, unde avem mai multe unități de “discuri logice”, referite prin C:, D:, ș.a.m.d.).

Acesta este configurat la instalarea sistemului de operare pe acel calculator, iar toate celelalte volume de fișiere, ce se dorește a fi accesate, se “includ” ca și subarbori în acesta, prin operația de *montare*.



# Montarea sistemelor de fișiere

Introducere

UNIX și utilizatorii

Fișiere în sisteme UNIX (cont.)

Structura fizică a sistemelor de fișiere

**Montarea sistemelor de fișiere**

Protecția fișierelor, prin permisiuni de acces

Alte comenzi pentru lucrul cu fișiere (cont.)

Alte categorii de comenzi simple

Troubleshooting

Referințe bibliografice

Un sistem UNIX poate conține mai multe volume de fișiere, fiecare volum fiind un sistem de fișiere de un anumit “tip” și care este stocat pe disc într-o “unitate de stocare” a volumelor de fișiere.

În mod uzual, un calculator poate avea mai multe discuri (*i.e.*, dispozitive fizice de stocare), iar fiecare disc poate fi “împărțit” într-una sau mai multe partiții. Iar fiecare partiție poate stoca un singur volum de fișiere. Însă, mai există și alte scheme de definire a “unității de stocare” (*e.g.*, sistemele RAID).

Fiecare volum de fișiere are, d.p.d.v. logic, un director rădăcină (referit prin “/”, în raport cu “unitatea de stocare” ce-l conține), și poate fi navigat prin încărcarea sistemului de operare de pe acel volum. Dacă totuși vreun sistem de fișiere de pe un dispozitiv de stocare trebuie folosit fără încărcarea sistemului de operare de pe acel dispozitiv, există soluția de **a monta** structura arborescentă a acelui sistem de fișiere în structura arborescentă a sistemului de fișiere “principal”, de pe care s-a încărcat sistemul de operare.

*Montarea* unui sistem de fișiere se face cu comanda **mount** (doar de către *superuser*), iar apoi accesarea conținutului său se face prin intermediul punctului de montare.

Iar operația inversă, de *demontare* a unui sistem de fișiere, se face cu comanda **umount**.

Fișiere de configurare: /etc/fstab (pentru montare automată), /etc/mtab, /etc/filesystems.

Pentru detalii despre aceste fișiere consultați documentația **man 8 mount** și **man 5 fstab**.



# Protecția fișierelor, prin permisiuni de acces

Introducere

UNIX și utilizatorii

Fișiere în sisteme UNIX (cont.)

Structura fizică a sistemelor de fișiere

Montarea sistemelor de fișiere

**Protecția fișierelor, prin permisiuni de acces**

Alte comenzi pentru lucrul cu fișiere (cont.)

Alte categorii de comenzi simple

Troubleshooting

Referințe bibliografice

## Modelul de securitate clasic din UNIX:

Fiecare fișier are asociat un anumit utilizator drept **utilizator proprietar**, și un anumit grup de utilizatori drept **grup proprietar** al fișierului.

Utilizatorii sistemului pot fi clasificați astfel în trei categorii în raport cu un fișier:

- proprietarul fișierului – **u** (*user*)
- colegii de grup ai proprietarului – **g** (*group*)
- toți ceilalți utilizatori – **o** (*others*)

Fiecare fișier are asociate câte trei **permisiuni de acces**, pentru fiecare dintre cele trei categorii de utilizatori ale acelui fișier:

- **r** (*read*) – drept de citire a conținutului fișierului
- **w** (*write*) – drept de scriere a conținutului fișierului (inclusiv de ștergere a fișierului)
- **x** (*execute*) – drept de execuție a fișierului





## Protecția fișierelor, prin permisiuni de acces (cont.)

Introducere

UNIX și utilizatorii

Fișiere în sisteme UNIX (cont.)

Structura fizică a sistemelor de fișiere

Montarea sistemelor de fișiere

Protecția fișierelor, prin permisiuni de acces

Alte comenzi pentru lucrul cu fișiere (cont.)

Alte categorii de comenzi simple

Troubleshooting

Referințe bibliografice

Pentru directoare, permisiunile de acces au o semnificație aparte:

- **r** (*read*) – drept de citire a conținutului directorului (*i.e.*, drept de aflare a numelor fișierelor (și subdirectoarelor) din acel director)
- **w** (*write*) – drept de scriere a conținutului directorului (*i.e.*, drept de adăugare de noi fișiere în director și de ștergere sau redenumire a fișierelor din director)
- **x** (*execute*) – drept de traversare a directorului (*i.e.*, drept de acces la fișierele (și subdirectoarele) din acel director)

Modificarea permisiunilor de acces ale unui fișier se poate face cu comanda **chmod**, doar de către proprietarul său (și de către *superuser*).

Schimbarea proprietarului, respectiv a grupului proprietar, al unui fișier se poate face folosind comenzile **chown**, respectiv **chgrp** ([5]). La fel, doar de către proprietarul actual al fișierului (și de către *superuser*).

*Demo:* a se vedea exemplul [Basic file operations #5] din suportul de laborator, disponibil [aici](#).



## Protecția fișierelor, prin permisiuni de acces (cont.)

Introducere

UNIX și utilizatorii

Fișiere în sisteme UNIX (cont.)

Structura fizică a sistemelor de fișiere

Montarea sistemelor de fișiere

Protecția fișierelor, prin permisiuni de acces

Alte comenzi pentru lucrul cu fișiere (cont.)

Alte categorii de comenzi simple

Troubleshooting

Referințe bibliografice

*Verificarea permisiunilor de acces* este efectuată de sistemul de operare la momentul inițierii unei sesiuni de citire/scriere a unui fișier (*i.e.*, la momentul apelului `open()`) sau la momentul inițierii execuției unui fișier (*i.e.*, la momentul apelului `exec()`), astfel:

Presupunem că fișierul ce se dorește a fi accesat este situat la calea absolută:

`/d1/d2/d3/.../file1`,

iar proprietarul comenzii ce dorește să-l acceseze este utilizatorul `Y`.

Sistemul de operare “parcurge” pe rând (*i.e.*, “de la stânga spre dreapta”) directoarele din acea cale (*i.e.*, `/`, `d1`, `d2`, `d3`, ...) și verifică prezența dreptului `x` pentru categoria din care face parte `Y` în raport cu proprietarul fiecărui director. Iar pentru fișierul propriu-zis, `file1`, se verifică prezența dreptului (sau drepturilor) corespunzător tipului de acces declarat în apelul `open()`, respectiv a dreptului `x` în cazul apelului `exec()`, pentru categoria din care face parte `Y` în raport cu proprietarul acelui fișier.

Dacă cel puțin una dintre aceste verificări individuale eșuează (*i.e.*, dacă dreptul de acces respectiv este interzis), atunci apelul respectiv (`open()` sau `exec()`) eșuează, returnând valoarea `-1` și, drept cauză a erorii, asignează în variabila `errno` o valoare numerică cu semnificația “Permission denied”.



## Alte comenzi pentru lucrul cu fișiere (cont.)

Introducere

UNIX și utilizatorii

Fișiere în sisteme UNIX (cont.)

Structura fizică a sistemelor de fișiere

Montarea sistemelor de fișiere

Protecția fișierelor, prin permisiuni de acces

Alte comenzi pentru lucrul cu fișiere (cont.)

Alte categorii de comenzi simple

Troubleshooting

Referințe bibliografice

- Comenzi pentru diverse prelucrări ale conținutului fișierelor ([6]):
  - `sort` – pentru sortarea unui fișier text (sau mai multe), după diverse criterii
  - `wc` – oferă anumite statistici despre conținutul fișierelor text (e.g., numărul de caractere, de cuvinte sau de linii de text)
  - `tr` – filtru pentru “translatarea” anumitor caractere specificate
  - `uniq` – filtru ce elimină liniile duplicat consecutive
  - `rev` – filtru pentru “inversarea” fiecărei linii de text
- Alte comenzi utile pentru fișiere:
  - `find` – pentru căutarea de fișiere (de orice tip), după diverse criterii ([6])
  - `cmp`, `comm`, `diff` – pentru compararea a două fișiere, după diverse criterii
  - `sum`, `md5sum`, `sha1sum`, `sha256sum` – calculează diverse sume de control (Notă: sunt utile pentru verificarea integrității fișierelor stocate pe perioade mai îndelungate.)

*Demo:* a se vedea exemplele `[sort #1]`, `[sort #2]`, `[wc #1]`, `[find #1]` și `[find #2]` din suportul de laborator, disponibil [aici](#).





# Agenda

Introducere

UNIX și utilizatorii

Fișiere în sisteme UNIX (cont.)

Alte categorii de comenzi simple

Procese. Sistemul de procese  
Comenzi ce oferă diverse informații

Alte categorii de comenzi

Troubleshooting

Referințe bibliografice

Introducere

## UNIX și utilizatorii

Conturi de utilizatori și grupuri de utilizatori

Comenzi referitoare la utilizatorii sistemului

## Fișiere în sisteme UNIX (cont.)

Structura fizică a sistemelor de fișiere

Montarea sistemelor de fișiere

Protecția fișierelor, prin permisiuni de acces

Alte comenzi pentru lucrul cu fișiere (cont.)

## Alte categorii de comenzi simple

Procese. Sistemul de procese

Comenzi ce oferă diverse informații

Alte categorii de comenzi

## Troubleshooting

“Ghid de supraviețuire la linia de comandă”

## Referințe bibliografice



# Procese. Sistemul de procese

[Introducere](#)

[UNIX și utilizatorii](#)

[Fișiere în sisteme UNIX \(cont.\)](#)

[Alte categorii de comenzi  
simple](#)

[Procese. Sistemul de procese](#)

[Comenzi ce oferă diverse  
informații](#)

[Alte categorii de comenzi](#)

[Troubleshooting](#)

[Referințe bibliografice](#)

**Definiție:** un *proces* este o instanță de execuție a unui program.

Procesele dintr-un sistem sunt organizate logic într-o *ierarhie arborescentă*, bazată pe un mecanism de “*moștenire genetică*”: fiecare proces din sistem are un proces care l-a creat, numit proces *părinte*, și de la care “moștenește” un anumit ansamblu de caracteristici (cum ar fi proprietarul, permisiunile de acces, ș.a.), și poate crea, la rândul lui, unul sau mai multe procese *fii*.

Fiecare proces are asignat un PID (acronim ce provine de la *Process IDentification*), ce este un număr întreg pozitiv și care este unic pe durata vieții acelui proces (*i.e.*, în orice moment, nu există în sistem două procese cu același PID).

Există un proces special, cel cu PID = 0, care este creat atunci când este inițializat (*boot-at*) sistemul UNIX pe calculatorul respectiv. Acesta nu are proces părinte, fiind rădăcina arborelui de procese ce se vor crea pe parcursul timpului (până la oprirea calculatorului).

- comenzi ce oferă informații despre procesele active în sistem: `ps`, `pstree`, `top`, `jobs`, `fg/bg`
- comenzi pentru planificarea / gestiunea execuției proceselor: `kill/killall`, `at`, `nice`, `time/times`, `timeout`

*Demo:* a se vedea exemplul [\[Info running programs\]](#) din suportul de laborator, disponibil [aici](#).



## Comenzi ce oferă diverse informații

[Introducere](#)

[UNIX și utilizatorii](#)

[Fișiere în sisteme UNIX \(cont.\)](#)

[Alte categorii de comenzi simple](#)

[Procese. Sistemul de procese](#)  
[Comenzi ce oferă diverse informații](#)

[Alte categorii de comenzi](#)

[Troubleshooting](#)

[Referințe bibliografice](#)

- informații despre timpul curent (dată + oră) și despre calendar: `date`, `cal`/`ncal`
- informații despre timpul de când este pornit sistemul curent: `uptime`
- informații despre terminale: `tty`, `stty`
- informații despre numele sistemului (calculatorului) curent – numele scurt, numele FQDN (*i.e.*, numele complet în sistemul DNS), adresa IP, ș.a.: `hostname`, `hostnamectl`, plus fișierele de configurare `/etc/hostname` ([7]), `/etc/machine-id`, `/etc/machine-info` și `/etc/hosts`
- informații despre nucleul (Linux sau alt UNIX) instalat pe sistemul curent: `uname`
- informații diverse despre componentele hardware ale calculatorului: `lshw`, `lscpu`, `lsmem`, `lspci`, ș.a.<sup>1</sup>
- informații despre distribuția de Linux instalată pe sistemul curent: fișierul de configurare `/etc/os-release` ([7]) **sau** `lsb_release`<sup>2</sup>

*Demo:* a se vedea exemplul [\[System info\]](#) din suportul de laborator, disponibil [aici](#).

<sup>1</sup> Acestea sunt comenzi specifice doar pentru sistemul de operare Linux.

<sup>2</sup> Această comandă este posibil să nu fie prezentă în unele distribuții de Linux.



## Alte categorii de comenzi

[Introducere](#)

[UNIX și utilizatorii](#)

[Fișiere în sisteme UNIX \(cont.\)](#)

[Alte categorii de comenzi simple](#)

[Procese. Sistemul de procese](#)  
[Comenzi ce oferă diverse informații](#)

[Alte categorii de comenzi](#)

[Troubleshooting](#)

[Referințe bibliografice](#)

- scrierea de mesaje pe ecran (*i.e.*, stdout): `printf` (mesaje formatate), `echo` (mesaje neformatate); ștergerea tuturor mesajelor de pe ecran: `clear`
- editoare neinteractive: `ed` (*line-oriented text editor*), `sed` (*stream editor*)
- limbaje de *scripting*: `awk`, `perl`, `python`, ș.a.
- arhivare, comprimare, codificare de fișiere: `gzip/gunzip`, `zip/unzip`, `tar`, `fsarchiver`, `compress/uncompress`, `uuencode/uudecode`, ș.a.
- scrierea de mesaje pe ecranul altui utilizator: `write`, `talk`
- conectarea la/deconectarea de la un sistem UNIX: `ssh`, `telnet`, `login` / `logout`, `exit`<sup>3</sup>
- programe pentru diverse protocoale INTERNET: `ftp`, `sftp`, `scp`, `mail/pine`, `lynx/links`, `host`, ș.a. (*Deprecated protocols*: `finger`, `talk`)
- utilitare pentru depanare și diagnosticare: `strace` și alte unelte înrudite (`ltrace`, `perf-trace`, `trace-cmd`, ș.a.)<sup>4</sup>

<sup>3</sup>Apelată doar din instanța de login a *shell*-ului.

<sup>4</sup>În prezent, aceste unelte sunt disponibile doar pentru sistemul de operare Linux.





# Agenda

Introducere

UNIX și utilizatorii

Fișiere în sisteme UNIX (cont.)

Alte categorii de comenzi simple

*Troubleshooting*

“Ghid de supraviețuire la linia de comandă”

Referințe bibliografice

Introducere

## UNIX și utilizatorii

Conturi de utilizatori și grupuri de utilizatori

Comenzi referitoare la utilizatorii sistemului

## Fișiere în sisteme UNIX (cont.)

Structura fizică a sistemelor de fișiere

Montarea sistemelor de fișiere

Protecția fișierelor, prin permisiuni de acces

Alte comenzi pentru lucrul cu fișiere (cont.)

## Alte categorii de comenzi simple

Procese. Sistemul de procese

Comenzi ce oferă diverse informații

Alte categorii de comenzi

## *Troubleshooting*

“Ghid de supraviețuire la linia de comandă”

## Referințe bibliografice



# “Ghid de supraviețuire la linia de comandă”

[Introducere](#)

[UNIX și utilizatorii](#)

[Fișiere în sisteme UNIX \(cont.\)](#)

[Alte categorii de comenzi  
simple](#)

[Troubleshooting](#)

[“Ghid de supraviețuire la linia  
de comandă”](#)

[Referințe bibliografice](#)

Ce să faceți dacă vi se “blochează” o comandă ? Urmăți pașii de mai jos:

- Mai întâi, așteptați un timp rezonabil, poate totuși programul nu este “blocat”, ci doar ocupat cu calcule intensive. Dacă totuși nu apare prompterul, atunci:
- Apăsăți (simultan) tastele **CTRL + C** . Aceasta determină trimiterea semnalului de întrerupere SIGINT programului respectiv. Dacă tot nu apare prompterul, atunci:
- Apăsăți (simultan) tastele **CTRL + \** . Aceasta determină trimiterea semnalului de terminare SIGQUIT programului respectiv. Dacă tot nu apare prompterul, atunci:
- Apăsăți (simultan) tastele **CTRL + Z** . Aceasta determină suspendarea programului respectiv (*i.e.*, trecerea lui în starea *SUSPENDED*) și afișarea prompterului.

Mai departe, pentru a opri acel program (el este doar suspendat, nu și terminat), procedați în felul următor:

Folosind comanda **ps** aflați PID-ul acelui program care vi se blocase, iar apoi tastați comanda  
UNIX> **kill -9 *pid***

unde *pid* este PID-ul aflat anterior.

Ca urmare a acestei comenzi procesul în cauză este omorât (*i.e.*, terminat forțat).





# Bibliografie obligatorie

[Introducere](#)

[UNIX și utilizatorii](#)

[Fișiere în sisteme UNIX \(cont.\)](#)

[Alte categorii de comenzi simple](#)

[Troubleshooting](#)

[Referințe bibliografice](#)

[1] Cap. 2, §2.1 și §2.2 din cartea “*Sisteme de operare – manual pentru ID*”, autor C. Vidrașcu, editura UAIC, 2006. *Notă*: este accesibilă, în format PDF, din pagina disciplinei “Sisteme de operare”:

- <https://edu.info.uaic.ro/sisteme-de-operare/SO/books/ManualID-SO.pdf>

[2] Suportul de laborator online asociat acestei prezentări:

- [https://edu.info.uaic.ro/sisteme-de-operare/SO/support-lessons/bash/suport\\_lab2.html](https://edu.info.uaic.ro/sisteme-de-operare/SO/support-lessons/bash/suport_lab2.html)

## Bibliografie suplimentară:

[3] Documentația despre `/etc/passwd`: `man 5 passwd` și `man 5 shadow`

[4] Documentația despre `/etc/group`: `man 5 group` și `man 5 gshadow`

[5] `man 1 chmod`, `man 1 chown` și `man 1 chgrp`

[6] `man 1 sort`, `man 1 wc`, `man 1 tr`, `man 1 uniq`, `man 1 rev` și `man 1 find`

[7] Documentațiile despre `/etc/hostname`, `/etc/os-release`, ș.a.: `man 5 hostname`, `man 5 machine-id`, `man 5 machine-info` și `man 5 os-release`

Plus documentațiile tuturor celorlalte comenzi uzuale prezentate, accesibile cu comanda `man`.