

GHID DE UTILIZARE LINUX (II)

Lucrul la linia de comandă în UNIX, partea I-a:

Comenzi de bază și sisteme de fișiere

Cristian Vidrașcu
vidrascu@info.uaic.ro

Februarie, 2021

Introducere	3
Principalele categorii de comenzi	5
Comenzi de <i>help</i>	6
Editoare de texte	8
Compilatoare, depanatoare, ș.a.	9
Comenzi pentru conectarea la distanță pe un server UNIX	10
Alte categorii de comenzi. ...	11
Fișiere în sisteme UNIX	12
Fișiere și sisteme de fișiere	13
Structura logică a sistemelor de fișiere	15
Comenzi de bază pentru lucrul cu fișiere (și directoare)	16
Alte comenzi pentru lucrul cu fișiere	18
Referințe bibliografice	19

Sumar

Introducere

Principalele categorii de comenzi

Comenzi de *help*

Editoare de texte

Compilatoare, depanatoare, ș.a.

Comenzi pentru conectarea la distanță pe un server UNIX

Alte categorii de comenzi. . .

Fișiere în sisteme UNIX

Fișiere și sisteme de fișiere

Structura logică a sistemelor de fișiere

Comenzi de bază pentru lucrul cu fișiere (și directoare)

Alte comenzi pentru lucrul cu fișiere

Referințe bibliografice

2 / 19

Introducere

Un *interpretor de comenzi* este un program executabil ce realizează interfața dintre utilizator și sistemul de operare, *i.e.* preia comenzile introduse de utilizator, le execută și afișează rezultatele execuției acestora.

În UNIX există mai multe interpretoare de comenzi, ce oferă posibilitatea lucrului la linia de comandă, *i.e.* interfața cu utilizatorul este în mod text, nu în mod grafic.

În sistemele din familia UNIX există două categorii de *comenzi simple*:

- *Comenzi interne*: sunt implementate în interpretoarele de comenzi.
Exemple: `cd`, `help`, ș.a.
- *Comenzi externe*: sunt implementate de sine stătător (*i.e.*, se găsesc fiecare în câte un fișier, având același nume cu comanda respectivă), în:
 - fișiere executabile (*i.e.*, programe executabile obținute prin compilare din programe sursă scrise în C sau în alte limbaje de programare).
Exemple: `passwd`, `ls`, ș.a.
 - fișiere text cu secvențe de comenzi, numite *script-uri*.
Exemple: `.profile`, `.bashrc`, ș.a.

3 / 19

Introducere (cont.)

Forma generală de lansare în execuție a unei comenzi simple:

UNIX> *comanda* [*opțiuni*] [*argumente*]

Observații:

- Textul "UNIX> " este *prompterul* afișat de interpretorul de comenzi, la care acesta așteaptă să tastezi comanda dorită, urmată de apăsarea tastei ENTER.
Notă: textul afișat ca și prompter este configurabil – vom discuta ulterior cum anume.
- Caracterul separator între numele comenzii și parametrii ei, precum și între fiecare dintre parametri, este caracterul SPACE sau TAB.
- Opțiunile și/sau argumentele specificate după numele comenzii pot eventual lipsi (*i.e.*, sunt opționale, lucru indicat printr-o pereche de paranteze '[. . .]'; acestea nu trebuie tastate).
- Prin convenție, opțiunile sunt precedate de caracterul '-' (sau '--', în cazul opțiunilor lungi).
- Semnificația argumentelor depinde de comandă (*e.g.*, cel mai adesea sunt nume de fișiere).
- Comenzile externe pot fi specificate și prin numele complet (*i.e.*, *calea absolută sau relativă*) al fișierului respectiv.
- O comandă mai lungă (*i.e.*, cu mulți parametri) poate fi introdusă pe mai multe linii, caz în care fiecare linie trebuie terminată cu caracterul '\' urmat de ENTER, cu excepția ultimei linii (terminarea acestuia se face apăsând doar tasta ENTER).

4 / 19

Principalele categorii de comenzi

5 / 19

Agenda

Introducere

Principalele categorii de comenzi

Comenzi de *help*

Editoare de texte

Compilatoare, depanatoare, ș.a.

Comenzi pentru conectarea la distanță pe un server UNIX

Alte categorii de comenzi...

Fișiere în sisteme UNIX

Fișiere și sisteme de fișiere

Structura logică a sistemelor de fișiere

Comenzi de bază pentru lucrul cu fișiere (și directoare)

Alte comenzi pentru lucrul cu fișiere

Referințe bibliografice

5 / 19

Comenzi de *help*

- Documentația despre comenzile interne este disponibilă cu ajutorul comenzii interne *help*. Ea se poate apela astfel:
UNIX> *help*
Va afișa lista tuturor comenzilor interne disponibile în interpretorul de comenzi respectiv.
UNIX> *help nume_comandă_internă*
Va afișa pagina de *help* pentru comanda internă specificată.
- Documentația despre comenzile externe, precum și despre funcțiile C din API-ul POSIX și cele din STANDARD C LIBRARY, este disponibilă cu ajutorul comenzilor *man*, *whatis*, *which*, *whereis*, *apropos*, *info* ([2]).
Observație: paginile de manual sunt organizate în 8 secțiuni. **Secțiunea 1** conține documentația despre comenzile uzuale accesibile utilizatorilor obișnuiți, iar **secțiunea 8** pe cea despre comenzile accesibile doar administratorului de sistem. **Secțiunea 2** conține documentația despre apelurile de sistem din Linux (*i.e.*, API-ul POSIX), iar **secțiunea 3** conține funcțiile din biblioteca standard de C. Despre restul secțiunilor și alte detalii puteți citi *aici* și *aici*.

6 / 19

Comenzi de *help* (cont.)

Modul de apelare a comenzilor ce oferă acces la documentația despre comenzile externe și despre funcțiile C din API-ul POSIX și din STANDARD C LIBRARY :

- UNIX> *whatis nume*
Va afișa lista comenzilor și funcțiilor C existente cu numele specificat, precum și secțiunile de manual ce le conțin.
- UNIX> *man [secțiune] nume*
Va afișa pagina de manual (din secțiunea specificată) pentru comanda sau funcția C specificată.
- UNIX> *which nume*
Va afișa locația comenzii.
- UNIX> *whereis nume*
Va afișa locația comenzii și a paginilor de manual asociate.
- UNIX> *apropos cuvânt*
Va căuta cuvântul specificat în descrierile comenzilor, afișând lista rezultatelor găsite.
- UNIX> *info [opțiuni] [cuvânt ...]*
Va afișa documentația, în formatul alternativ TEXINFO, pentru cuvintele specificate.

Demo: a se vedea exemplul [Accesarea paginilor de manual] din suportul de laborator, disponibil *aici*.

7 / 19

Editoare de texte

La fel ca în Windows, și în UNIX există mai multe categorii de editoare:

- Editoare de texte pentru text “obișnuit” (*i.e.*, fără formataări specializate):
 - a) în mod text: `mc`, `pico`, `nano`, `vi`, `vim`, `joe`, ș.a. ([3])
 - b) în mod grafic: `gedit`, `kedit`, ș.a.
- Editorul `emacs`: este un editor în mod text, ce face parte din proiectul GNU, cu facilități puternice, utilizabil ca și mediu integrat de programare.
- Suite de birou (pentru mediul grafic): OPENOFFICE, LIBREOFFICE, ș.a.
- Editoare specializate pentru HTML: acestea permit scrierea de pagini web în limbajul HTML, ce conține instrucțiuni de formatare a textului propriu-zis.
- Editoare specializate pentru TEX / LATEX: un asemenea editor este un *mediu de lucru* ce permite tehnoredactarea de documente științifice în limbajul $\text{T}_{\text{E}}\text{X}$ / $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$.
- Alte tipuri de editoare (*e.g.*, cele cu facilități specializate pentru scrierea de programe în diverse limbaje de programare, adesea făcând parte dintr-un mediu integrat de programare).

8 / 19

Compilatoare, depanatoare, ș.a.

În sistemele UNIX există compilatoare și interpretoare pentru majoritatea limbajelor de programare existente, mai noi sau mai vechi: C, C++, Pascal, Fortran, Java, Ada, ș.a.

Pentru limbajele C și C++, cel mai popular este compilatorul din suita GCC (*the GNU Compiler Collection*) ([4]), care se utilizează astfel:

- Compilarea unui program C se face cu comanda:
`UNIX> gcc sursa.c [-o executabil] [-Wall]`
care realizează compilarea (inclusiv *link*-editarea codului obiect) fișierului sursă specificat (*atenție*: este obligatorie extensia `.c`) în fișierul executabil cu numele specificat prin opțiunea `-o` (în lipsa ei, numele implicit al executabilului este `a.out`).
- În mod asemănător, compilarea unui program C++ se face cu comanda:
`UNIX> g++ sursa.cpp [-o executabil] [-Wall]`

Pentru depanarea programelor se poate utiliza depanatorul GNU DEBUGGER, accesibil prin comanda `gdb`.

9 / 19

Comenzi pentru conectarea la distanță pe un server UNIX

■ Sesiune de lucru interactivă la distanță, *i.e.* pe un server UNIX (sau Linux):

- cu comunicații necriptate:

```
UNIX> telnet [optiuni] [calculator [port]]
```

- cu comunicații criptate ([5]):

```
UNIX> ssh [optiuni] [[username@] calculator]
```

Notă: sub Windows se poate folosi (și) aplicația Putty pentru conexiunea la un server UNIX.

■ Sesiune interactivă pentru transfer de fișiere la distanță:

- cu comunicații necriptate:

```
UNIX> ftp [optiuni] [calculator [port]]
```

- cu comunicații criptate ([5]):

```
UNIX> sftp [optiuni] calculator
```

O variantă neinteractivă este comanda scp:

```
UNIX> scp [opts] [[user@]host1:]file1 [[user@]host2:]file2
```

Notă: sub Windows se poate folosi (și) aplicația WinSCP pentru transfer de fișiere.

Observație: în cadrul FII există un server Linux pentru uzul studenților, pe care puteți lucra până vă veți instala Linux pe calculatorul personal. Numele serverului este `students.info.uaic.ro` și permite doar conexiuni criptate. Autentificarea pe acest server se face utilizând numele de cont și parola pentru contul dvs. de email de la facultate (*i.e.*, contul pe care l-ați primit la începutul anului universitar).

10 / 19

Alte categorii de comenzi...

Observație:

Mai întâi vom introduce conceptele legate de *fișiere* și *sisteme de fișiere*, iar apoi vom continua cu prezentarea altor categorii de comenzi disponibile în sistemele UNIX.

(va urma)

11 / 19

Agenda

Introducere

Principalele categorii de comenzi

- Comenzi de *help*
- Editoare de texte
- Compilatoare, depanatoare, ș.a.
- Comenzi pentru conectarea la distanță pe un server UNIX
- Alte categorii de comenzi...

Fișiere în sisteme UNIX

- Fișiere și sisteme de fișiere
- Structura logică a sistemelor de fișiere
- Comenzi de bază pentru lucrul cu fișiere (și directoare)
- Alte comenzi pentru lucrul cu fișiere

Referințe bibliografice

12 / 19

Fișiere și sisteme de fișiere

La fel ca și în Windows, în sistemele de operare din familia UNIX datele și programele utilizatorilor sunt păstrate pe disc în *fișiere*. Fiecare fișier este identificat printr-un *nume* atribuit de *proprietarul* său (*i.e.*, utilizatorul care l-a creat).

Fișierele sunt organizate (*i.e.*, grupate) în “volume” pe disc, numite *sisteme de fișiere*.

Numele fișierelor în sistemele de fișiere din UNIX sunt *case-sensitive*, spre deosebire de sistemele de fișiere din Windows (FAT și NTFS).

Numele fișierelor pot avea până la 255 de caractere și pot conține oricâte caractere ‘.’ (*i.e.*, nu sunt împărțite sub forma 8.3, *nume.extensie*, ca în sistemul de fișiere FAT din MS-DOS sau Windows), singura restricție fiind nefolosirea caracterelor NULL și ‘/’, precum și a altor câteva caractere speciale, *e.g.* LF (*Line Feed*), CR (*Carriage Return*), ș.a.

Notă: deși sistemele UNIX nu impun nici o convenție privitoare la numirea fișierelor, există totuși sufixe utilizate în mod standard, precum ar fi: *.c* și *.h* pentru fișiere sursă în limbajul C, *.cpp* și *.h* pentru fișiere sursă în limbajul C++, *.txt* pentru fișiere cu text simplu, neformatat, *.sh* pentru fișiere cu secvențe de comenzi (numite *script-uri*), *.tar/.gz/.zip* pentru arhive în format tar/gz/zip, ș.a. Aceste sufixe se numesc *extensii* și sunt utile d.p.d.v. al utilizatorilor, pentru a le “sugera” ce fel de conținut are fișierul respectiv, la fel ca și în Windows.

13 / 19

Fișiere și sisteme de fișiere (cont.)

În sistemele de fișiere din UNIX, fișierele se clasifică în următoarele 6 *tipuri de fișiere*:

- fișiere normale – sunt fișiere cu conținut ordinar / obișnuit (date sau programe);
- directoare – sunt “cataloge” de fișiere;
- *link*-uri (legături), *hard* sau simbolice – sunt un fel de *alias*-uri, *i.e.* alte nume “sinonime” pentru fișiere deja existente;
- fișiere speciale, în mod bloc sau în mod caracter – sunt drivere de periferice (fizice sau logice);
- fișiere *fifo* – sunt mecanisme de comunicație locală între procese;
- fișiere *socket* – sunt folosite pentru comunicația la distanță între procese, prin rețea (*i.e.*, între programe rulate pe sisteme diferite).

Atenție: a nu se confunda extensia din numele unui fișier cu tipul acestuia!

Toate exemplele de extensii amintite pe *slide*-ul anterior sunt folosite cu fișiere de tipul normal.

Notă: cu primele 3 tipuri de fișiere sunteți deja familiarizați din Windows; *e.g.*, echivalentul *link*-urilor sunt așa-numitele “*shortcut*-uri” (pe desktop). De fapt, și celelalte tipuri de fișiere au echivalent în Windows, doar că, probabil, încă nu le-ați întâlnit în practică (Le veți întâlni în anul 3, la opționalul CSSO).

14 / 19

Structura logică a sistemelor de fișiere

În UNIX, fiecare **sistem de fișiere** este organizat d.p.d.v. *logic* (*i.e.*, d.p.d.v. al modului în care este “văzut” de utilizatori și de programele ce accesează fișierele) într-o *ierarhie arborescentă*, bazată pe ideea de *director* (*i.e.*, un “container” de fișiere):

la fel ca în Windows, sistemul de fișiere este organizat ca un arbore format recursiv din directoare, ce pot conține (sub)directoare și fișiere propriu-zise (*i.e.*, de tip normal și celelalte tipuri de fișiere).

Însă, spre deosebire de Windows, într-un sistem UNIX avem un singur “arbore logic”, numit *the root file-system*, iar rădăcina acestui arbore este referită prin numele “/”.

Numele complet al unui fișier, indiferent de tipul său, este dat de reprezentarea liniară a căii de la rădăcina “/” și până la fișierul respectiv, în acest “arbore logic”.

Exemple: `/usr/bin/bash` , `/etc/passwd` , `/dev/sda` , `/home/username/.profile` , ș.a.

Notă: această reprezentare liniară, numită și *calea absolută* a acelui fișier, utilizează caracterul ‘/’ ca și separator între numele (sub)directoarelor din calea respectivă (spre deosebire de Windows, care utilizează caracterul ‘\’ ca și separator).

De fapt, la fel ca în Windows, și în sistemele UNIX fișierele pot fi accesate (specificate) în două moduri:
– relativ la rădăcina “/” sistemului de fișiere (*i.e.*, specificare prin **cale absolută**);
– sau relativ la *directorul curent de lucru* (*i.e.*, specificare prin **cale relativă la directorul curent**).

15 / 19

Comenzi de bază pentru lucrul cu fișiere (și directoare)

■ Comenzi de bază pentru manipularea fișierelor de tipul director:

- `mkdir` – pentru crearea unui director (sau mai multe)
- `rmdir` – pentru ștergerea unui director (sau mai multe)
- `ls` – pentru afișarea conținutului unui director (sau mai multe)
- `pwd` – pentru aflarea directorului curent de lucru
- `cd` – pentru schimbarea directorului curent de lucru

Demo: a se vedea exemplele [Basic file operations #1], i) și [Basic file operations #2], iii) din suportul de laborator, disponibil [aici](#).

16 / 19

Comenzi de bază pentru lucrul cu fișiere (și directoare) (cont.)

■ Comenzi de bază pentru manipularea fișierelor de orice tip:

- `ln` – pentru crearea unui fișier (sau mai multe) de tipul *alias*
- `mkfifo` – pentru crearea unui fișier (sau mai multe) de tipul *fifo*
- `mknod` – pentru crearea unui fișier (sau mai multe) de tipul *device*
- pentru crearea unui fișier (sau mai multe) de tipul normal, putem folosi mai multe comenzi: orice editor de texte obișnuit, sau `touch`, `truncate`, ș.a.
- `rm` – pentru ștergerea fișierelor de orice tip
- `cp` – pentru copierea de fișiere de orice tip
- `mv` – pentru mutarea și/sau redenumirea de fișiere de orice tip
- `rename` – pentru redenumirea multiplă de fișiere
- `realpath` – pentru afișarea căii absolute a unui fișier (sau mai multe)
- `mc` – utilitarul “Midnight Commander”, având 2 *panel*-uri cu grafică text, pentru manipulări diverse ale fișierelor

Demo: a se vedea exemplele [Basic file operations #1], ii)-iv) și [Basic file operations #2], i)-ii) din suportul de laborator, disponibil [aici](#).

17 / 19

Alte comenzi pentru lucrul cu fișiere

■ Comenzi pentru diverse prelucrări ale conținutului fișierelor:

- `cat`, `tac`, `more`, `less`, `head`, `tail`, `mcview` – pentru afișarea, în diverse formate, a conținutului unui fișier (sau mai multe) de tipul normal
- `file` – oferă informații despre “tipul de informație” dintr-un fișier, pe baza inspecției conținutului acestuia
- `stat` – pentru afișarea atributelor (*i.e.*, informațiile asociate) despre un fișier (*Notă:* este o comandă mai “puternică” decât comanda `ls -l`.)
- `grep` – pentru selectarea liniilor de text, ce conțin un anumit șablon, dintr-un fișier (sau mai multe) de tipul normal (*i.e.*, face “selecție pe orizontală”)
- `cut` – pentru selectarea anumitor coloane de text dintr-un fișier (sau mai multe) de tipul normal (*i.e.*, face “selecție pe verticală”)

Demo: a se vedea exemplele [Basic file operations #3], [Basic file operations #4], [cut #1] și [grep #1] din suportul de laborator, disponibil [aici](#).

Referințe bibliografice

19 / 19

Bibliografie obligatorie

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

• <https://profs.info.uaic.ro/~vidrascu/SO/books/ManualID-SO.pdf>

[2] Documentația comenzilor de *help*: `man 1 man`, `man 1 whatis`, ș.a.

[3] Documentația editoarelor de texte obișnuite: `man mcedit`, `man nano`, `man vim`, ș.a.

[4] Informații despre *GCC (the GNU Compiler Collection)*

[5] Documentația comenzilor de conectare la distanță: `man 1 ssh`, `man 1 scp`, ș.a.

Plus documentația tuturor celorlalte comenzi uzuale prezentate, accesibilă cu comanda `man`.

19 / 19