

# Laboratorul 1 : Asistența în Linux

**Durată estimată studiu: 100 -120 minute**

---

## 1.1 Introducere

Unul dintre cele mai importante skilluri în lucrul cu Linux este abilitatea de a găsi singur informații despre comenzi. Sistemele Linux includ documentație completă integrată, astfel încât nu este necesară conectarea la internet pentru a afla cum funcționează o comandă.

### De ce este important:

- Lucru în medii izolate/securizate (fără internet)
  - Autonomie în rezolvarea problemelor
  - Înțelegere profundă a sistemului
  - Pregătire pentru certificări Linux / interviu
- 

## 1.2 Ierarhia documentației

Când întâlnim o comandă necunoscută, există o ordine logică de căutare a informațiilor:

### 1. Ajutor rapid (2 secunde):

`command --help`

### 2. Manual de referință (2 minute):

`man command`

`whatis command`

### 3. Căutare în documentație (5 minute):

`apropos keyword`

`ls /usr/share/doc/`

---

## 1.3 Comenzile de ajutor

### 1.3.1 Opțiunea --help

Majoritatea comenzilor moderne acceptă opțiunea `--help` care afișează un rezumat scurt al sintaxei și opțiunilor disponibile. Este modalitatea cea mai rapidă de a obține informații de bază.

*# Afișare opțiuni ls*

`ls --help`

*# Output:*

*# Usage: ls [OPTION]... [FILE]...*

*# -l use a long listing format*

*# -a do not ignore entries starting with .*

*# -h with -l, print sizes in human readable format*

#### Exercițiul 1.1:

Folosiți opțiunea `--help` pentru a afla:

Ce opțiune a comenzii `cd` afișează directorul HOME?

Ce face opțiunea `-r` la comanda `cp`?

Găsiți opțiunea comenzii `mkdir` care creează și directoarele părinte.

#### Exercițiul 1.2:

Deschideți terminalul și executați:

1. `grep --help` - identificați opțiunea pentru căutare case-insensitive
  2. `find --help` - găsiți opțiunea pentru căutare după dimensiune
  3. Notați timpul aproximativ necesar pentru fiecare căutare
-

### 1.3.2 Manual pages (man) - Ghid complet

Sistemul man (manual) este documentația oficială a sistemului Linux. Spre deosebire de `--help` care oferă doar un rezumat, man pages conțin descrieri complete, exemple detaliate, și explicații pentru toate opțiunile disponibile.

#### Organizarea pe secțiuni

Man pages sunt organizate în secțiuni numerotate. Aceeași comandă poate exista în mai multe secțiuni, fiecare documentând un aspect diferit:

Secțiune	Conținut	Exemple
1	Comenzi utilizator general	ls, cp, grep, find
2	System calls (apeluri kernel)	open, read, write, fork
3	Library functions (C library)	printf, malloc, strlen
4	Device files și drivere	null, zero, tty
5	Formate fișiere configurare	/etc/passwd, fstab, crontab
6	Jocuri și screensavere	-
7	Convenții și protocoale	ip(7), tcp(7), regex(7)
8	Comenzi administrare sistem	mount, iptables, systemctl

#### De ce contează secțiunile?

Uneori același nume apare în mai multe secțiuni. De exemplu:

```
passwd (1) - comanda pentru schimbarea parolei
passwd (5) - formatul fișierului /etc/passwd
```

Dacă rulezi doar `man passwd`, vei vedea secțiunea 1 (comanda). Pentru a vedea secțiunea 5 (formatul fișierului), trebuie să specifici explicit:

```
man 5 passwd
```

#### Sintaxă accesare secțiuni

*# Sintaxă generală*

```
man [secțiune] nume_comandă
```

*# Exemple concrete*

```
man ls           # Secțiunea 1 (default - comenzi user)
```

```
man 5 crontab    # Secțiunea 5 (format fișier configurare)
```

```
man 8 mount      # Secțiunea 8 (comenzi admin)
```

```
man 7 regex      # Secțiunea 7 (convenții - expresii regulate)
```

## Căutare în ce secțiuni există o comandă

# Vezi toate secțiunile disponibile pentru "printf"

`man -f printf`

# Output:

# printf (1) - format and print data

# printf (3) - formatted output conversion

# SAU folosește *whatis* (echivalent)

`whatis printf`

## Navigare în man pages

Tastă/Comandă	Acțiune
Space sau Page Down	Pagină următoare
b sau Page Up	Pagină anterioară
Enter sau ↓	O linie în jos
↑	O linie în sus
g sau Home	Salt la început
G sau End	Salt la sfârșit
/pattern	Caută "pattern" înainte (forward search)
?pattern	Caută "pattern" înapoi (backward search)
n	Următorul rezultat căutare
N	Rezultatul anterior căutare
h	Help (ajutor navigare)
q	Ieșire (quit)

## Tips & Tricks pentru eficiență

### 1. Căutare rapidă în secțiuni specifice:

În loc să citești tot man page-ul, sari direct la secțiunea care te interesează:

# Deschide man ls

`man ls`

# Odată deschis, caută direct secțiunea *OPTIONS*

`/OPTIONS`

# Apoi apasă Enter - sare direct la opțiuni

*# Caută toate aparițiile cuvântului "recursive"*

/recursive

*# Apasă 'n' pentru următorul rezultat*

## 2. Găsire rapidă exemple:

Majoritatea man pages au secțiune EXAMPLES la sfârșit:

`man rsync`

*# Apoi caută:*

/EXAMPLES

## 3. Comparare între secțiuni:

Deschide două terminale și compară:

*# Terminal 1*

`man 1 passwd` *# comanda*

*# Terminal 2*

`man 5 passwd` *# formatul fișierului*

## 4. Export man page pentru citire offline:

*# Salvează ca text*

`man ls > ls_manual.txt`

*# Sau ca PDF (dacă ai ps2pdf instalat)*

`man -t ls | ps2pdf - ls_manual.pdf`

## 5. Căutare în toate man pages:

*# Caută "mount" în toate descrierile*

`man -k mount`

*# SAU mai detaliat, caută în conținutul efectiv*

`man -K "USB device"` *# Atenție: durează mai mult!*

## Structura tipică a unui man page

Aproape toate man pages urmează aceeași structură:

1. **NAME** - Numele și scurtă descriere
2. **SYNOPSIS** - Sintaxa comenzii (cel mai important!)
3. **DESCRIPTION** - Descriere detaliată
4. **OPTIONS** - Toate opțiunile disponibile (aici petreci cel mai mult timp!)
5. **EXAMPLES** - Exemple practice (salt direct aici cu /EXAMPLES)
6. **FILES** - Fișiere de configurare asociate
7. **SEE ALSO** - Comenzi și man pages relacionate
8. **BUGS** - Probleme cunoscute
9. **AUTHOR** - Cine a scris comanda/man page-ul

**Shortcut:** Când deschizi un man page, citește întâi **SYNOPSIS** (sintaxa), apoi sari la **EXAMPLES**, apoi revii la **OPTIONS** pentru detalii.

## Înțelegerea sintaxei din SYNOPSIS

Man pages folosesc convenții speciale în secțiunea SYNOPSIS:

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

*# Legendă:*

*# [text] = opțional*

*# <text> = obligatoriu*

*# ... = poate fi repetat*

*# | = SAU (alegi una)*

*# text = literar (scrii exact așa)*

Exemple practice:

*# cp [OPTION]... SOURCE DEST*

*# SOURCE și DEST sunt OBLIGATORII (nu sunt în [])*

`cp file1.txt file2.txt`

*# ls [OPTION]... [FILE]...*

*# FILE este OPȚIONAL (în [])*

`ls` *# funcționează*

`ls /var/log` *# funcționează și cu argument*

# mount [-t type] device directory

# -t type este *OPȚIONAL*

mount /dev/sdb1 /mnt # funcționează

mount -t ext4 /dev/sdb1 /mnt # funcționează

### Exercițiul 1.3:

Folosind man, răspundeți la următoarele:

1. Deschideți man `ls` și sari **direct** la secțiunea `OPTIONS` (folosind `/OPTIONS`)
2. Găsiți ce face opțiunea `-R` (hint: caută `/recursive`)
3. În man `cp`, găsiți diferența dintre `-r` și `-a`
4. Deschideți man `5 fstab` (format fișier) și citiți primele 3 paragrafe
5. Comparați man `mount` (secțiunea 8) cu man `2 mount` (system call)

### Exercițiul 1.4:

Practică cu secțiuni:

1. Rulați man `-f printf` - câte secțiuni există?
2. Deschideți man `1 printf` - ce face comanda?
3. Deschideți man `3 printf` - ce face funcția din C?
4. Observați diferența dintre cele două

### Exercițiul 1.5:

Workflow eficient:

1. Deschideți man `find`
2. Căutați direct secțiunea `EXAMPLES`: `/EXAMPLES`
3. Găsiți un exemplu care caută fișiere mai mari de 100MB
4. Ieșiți și testați comanda găsită
5. Reveniți în man și căutați opțiunea `-size` pentru detalii: `/-size`

### Exercițiul 1.6:

Challenge interpretare SYNOPSIS:

1. Deschideți man `grep`
2. Citiți secțiunea `SYNOPSIS`
3. Explicați ce înseamnă: `grep [OPTION] ... PATTERN [FILE] ...`
4. Care argumente sunt obligatorii? Care opționale?
5. Testați: `grep "error"` (fără `FILE`) - funcționează? De ce?

### 1.3.3 whatis - Descriere scurtă

Comanda `whatis` afișează o descriere pe o singură linie a unei comenzi, extrasă din `man` page. Utilă pentru a afla rapid ce face o comandă înainte de a citi documentația completă.

`whatis ls`

*# Output: ls (1) - list directory contents*

`whatis mount`

*# Output: mount (8) - mount a filesystem*

`whatis ssh_config`

*# Output: ssh\_config (5) - OpenSSH client configuration file*

Observați numărul din paranteză - acesta indică secțiunea `man` page. Este util pentru a ști dacă este comandă (1), fișier configurare (5), sau comandă admin (8).

#### Exercițiul 1.7:

Folosiți `whatis` pentru a afla ce fac următoarele comenzi:

1. `grep`
2. `chmod`
3. `systemctl`
4. `rsync`

Notați ce secțiune (numărul din paranteză) aparține fiecare.

#### Exercițiul 1.8:

Cercetare:

1. Rulați `whatis rsyslog`
  2. Dacă nu apare rezultat, rulați `sudo mandb` (reconstruiește database `man`)
  3. Reîncercați `whatis rsyslog`
  4. Explicați de ce uneori `whatis` nu găsește rezultate imediat după instalarea unui pachet
-



### 1.3.4 apropos - Căutare în descrieri

apropos caută cuvinte cheie în toate descrierile man pages. Este extrem de util când nu știm exact numele comenzii, dar știm ce funcționalitate căutăm.

*# Caută toate comenzile legate de "compress"*

`apropos compress`

*# Output:*

*# bzip2 (1) - compress or expand files*

*# gzip (1) - compress or expand files*

*# zip (1) - package and compress files*

*# Caută comenzi pentru lucru cu rețeaua (doar admin - secțiunea 8)*

`apropos network | grep "(8)"`

*# Caută tot ce conține "copy" în descriere*

`apropos copy`

Diferența față de `whatis`:

`whatis` caută **nume exact** de comandă

`apropos` caută **cuvinte** în orice descriere

`whatis copy`

*# Output: copy: nothing appropriate*

`apropos copy`

*# Output: multe rezultate (cp, rsync, scp, etc.)*

#### **Exercițiul 1.9:**

Folosiți `apropos` pentru a găsi:

1. Comenzi pentru lucru cu arhive (keyword: "archive")
2. Comenzi pentru management utilizatori (keyword: "user")
3. Comenzi pentru monitoring sistem (keyword: "monitor")

Pentru fiecare, notați cel puțin 3 comenzi găsite și secțiunea lor.

#### **Exercițiul 1.10:**

Rezolvare problemă reală:

1. Trebuie să găsiți o comandă care sincronizează fișiere între directoare
2. Folosiți `apropos sync` sau `apropos copy`
3. Din rezultate, identificați comanda `rsync`
4. Citiți descrierea cu `whatis rsync`
5. Deschideți manualul: `man rsync` și găsiți secțiunea **EXAMPLES**

### 1.3.5 whereis - Localizare comenzi

`whereis` caută și afișează locația executabilului, man page-urilor și fișierelor sursă pentru o comandă. Este util pentru:

- Verificare dacă o comandă este instalată
- Găsire locație man pages comprimate (.gz)
- Debugging când există versiuni multiple instalate
- Diferențiere între comandă și funcție built-in

`whereis python`

*# Output: python: /usr/bin/python3 /usr/share/man/man1/python3.1.gz*

`whereis ls`

*# Output: ls: /bin/ls /usr/share/man/man1/ls.1.gz*

*# Observați: man pages sunt comprimate cu gzip (.gz)*

Diferența dintre `whereis` și `which`:

- `whereis` - găsește **TOATE** locațiile (binary + man pages + source)
- `which` - găsește doar **primul** executabil din \$PATH

`whereis python`

*# Output: /usr/bin/python2.7 /usr/bin/python3.10 /usr/share/man/...*

`which python`

*# Output: /usr/bin/python3 (doar primul din PATH)*

#### Exercițiul 1.11:

Folosiți `whereis` pentru:

- Verificați dacă `nginx` este instalat în sistem
- Găsiți locația man page-ului pentru (va fi .gz)
- Comparați output-ul `whereis python` cu `which python`
- Explicați diferența

#### Exercițiul 1.12:

Investigație versiuni multiple:

- Rulați `whereis python`
  - Notați toate versiunile găsite
  - Rulați `which python` - care versiune este folosită default?
  - Rulați `ls -la $(which python)` - este link simbolic? Către ce versiune?
-

## 1.4 Documentație în /usr/share/doc/

Directorul `/usr/share/doc/` conține documentație extinsă pentru pachetele instalate: README files, CHANGELOG, exemple de configurare, licențe. Este locul unde găsești informații care nu sunt în man pages, precum:

- Ghiduri de configurare pas-cu-pas
- Exemple funcționale de configurații
- Known issues și soluții
- Diferențe specifice distribuției (README.Debian, README.Fedora)

*# Listare pachete documentate*

```
ls /usr/share/doc/
```

*# Citire README pentru nginx*

```
less /usr/share/doc/nginx/README.Debian
```

*# Găsire exemple configurare*

```
ls /usr/share/doc/nginx/examples/
```

*# Exemple practice*

```
cat /usr/share/doc/rsyslog/examples/remote-logging.conf
```

`/usr/share/doc/package-name/`

- |— README
- |— README.Debian      *# Specific distro*
- |— changelog.Debian.gz    *# Istoric modificări*
- |— copyright            *# Licență*
- |— examples/            *# Exemple configurări*
  - |    |— config1.conf
  - |    |— config2.conf
- |— NEWS.gz              *# Noutăți versiuni*

### Exercițiul 1.13:

Explorare documentație:

1. Listați conținutul `/usr/share/doc/`
2. Alegeți un pachet instalat (ex: `sudo`, `openssh-client`, sau )
3. Explorați directorul său: `ls -la /usr/share/doc/nume-pachet/`

4. Citiți fișierul README (folosind `less` sau `cat`)
5. Notați ce informații găsiți care NU sunt în man page

**Exercițiul 1.14:**

Găsire exemple de configurare:

1. Verificați dacă există `/usr/share/doc/rsyslog/examples/`
  2. Listați fișierele de exemplu disponibile
  3. Afișați conținutul unui fișier `.conf` de exemplu
  4. Identificați ce configurează (remote logging? local filtering?)
  5. Notați unde sunt comentariile explicative în fișier
- 

## 1.5 Practică integrată - Workflow real

**Exercițiul 1.15 - Workflow complet:**

**Sarcină:** Trebuie să configurați serviciul `cron` (task scheduler Linux), dar nu știți nimic despre el.

**Pași de urmat:**

1. Folosiți `what is cron` - obțineți descrierea de bază (ce face?)
2. Folosiți `man cron` - citiți primele 2 paragrafe din DESCRIPTION
3. Căutați în man unde sunt stocate job-urile (hint: căutați `/var/spool`)
4. Folosiți `apropos cron` - descoperiți comenzi related (`crontab` apare?)
5. Citiți `man 5 crontab` - format fișier (diferit de `man 1 crontab`!)
6. Explorați `/usr/share/doc/cron/` - există exemple?
7. Folosiți `whereis crontab` - unde este executabilul?

**Timp alocat:** 10 minute

**Rezultat așteptat:** Înțelegere completă `cron` fără Google/internet

---

## 1.6 Quick Reference - Comandă pe scenariu

Scenariu	Comenzi recomandate	Ordinea de utilizare
Comandă nouă, necunoscută	<code>what is → man → --help</code>	1. Află ce face 2. Citește detaliat 3. Vedere rapidă opțiuni
Nu știi numele comenzii	<code>apropos keyword</code>	Caută în descrieri
Verificare instalare	<code>whereis command</code>	Vezi dacă există binary
Configurare serviciu	<code>man service</code> <code>/usr/share/doc/service/</code>	1. Man pentru sintaxă 2. Doc pentru exemple
Căutare în man page	<code>/keyword în man</code>	Ctrl+F echivalent
Comparare opțiuni	<code>man command + --help</code>	Man pentru detalii --help pentru recall rapid

---

## 1.7 Detectarea versiunii și sistemului

După ce am învățat să găsim informații despre comenzi, este esențial să înțelegem sistemul pe care lucrăm: ce versiune de kernel rulează, ce distribuție folosim. Aceste cunoștințe sunt fundamentale pentru:

- Verificarea compatibilității software-ului
- Troubleshooting specific distribuției
- Înțelegerea unde să cauți fișiere de configurare
- Lucrul eficient în medii de producție

### 1.7.1 Identificarea versiunii kernel

Kernel-ul este nucleul sistemului de operare Linux. Versiunea kernel-ului poate influența compatibilitatea cu hardware-ul și anumite funcționalități.

*# Versiune kernel scurtă*

```
uname -r
```

*# Output: 5.15.0-91-generic*

*# Informații complete sistem*

```
uname -a
```

*# Output: Linux hostname 5.15.0-91-generic #101-Ubuntu SMP x86\_64 GNU/Linux*

Decompoziția versiunii 5.15.0-91-generic:

**5.15** - versiunea majoră kernel

**0** - versiunea minor

**91** - patch number

**generic** - varianta kernel (generic, lowlatency, etc.)

*# Doar numele sistemului de operare*

```
uname -s
```

*# Output: Linux*

*# Arhitectura procesorului*

```
uname -m
```

*# Output: x86\_64 (64-bit) sau aarch64 (ARM 64-bit) sau i686 (32-bit)*

*# Hostname-ul sistemului*

```
uname -n
```

*# Output: server01*

### Exercițiul 1.17:

Identificați sistemul vostru:

1. Rulați `uname -r` - notați versiunea kernel
2. Rulați `uname -m` - este sistem 32-bit sau 64-bit?
3. Rulați `uname -a` - identificați distribuția din output
4. Comparați versiunea kernel cu un coleg - sunt identice?

### Exercițiul 1.18:

Informații detaliate kernel:

1. Citiți fișierul `/proc/version`: `cat /proc/version`
2. Observați informații despre compilator (gcc version)
3. Comparați cu output-ul `uname -a` - ce informații suplimentare găsiți?

## 1.7.2 Identificarea distribuției Linux

Fiecare distribuție Linux are caracteristici specifice: package manager diferit, locații config diferite, cicluri de release diferite.

*# Metoda standard - fișierul os-release*

`cat /etc/os-release`

*# Output (exemplu Ubuntu):*

*# NAME="Ubuntu"*

*# VERSION="22.04.3 LTS (Jammy Jellyfish)"*

*# ID=ubuntu*

*# ID\_LIKE=debian*

*# PRETTY\_NAME="Ubuntu 22.04.3 LTS"*

*# VERSION\_ID="22.04"*

Informații importante din `/etc/os-release`:

**NAME** - numele distribuției

**VERSION** - versiunea și codenum

**ID** - identificator scurt (ubuntu, debian, centos, etc.)

**ID\_LIKE** - familia din care face parte (debian, rhel, arch)

**VERSION\_ID** - doar numărul versiunii

*# Informații sistem complete (modern)*

hostnamectl

*# Output:*

*# Static hostname: server01*

*# Operating System: Ubuntu 22.04.3 LTS*

*# Kernel: Linux 5.15.0-91-generic*

*# Architecture: x86-64*

*# LSB info (dacă este instalat lsb-release)*

lsb\_release -a

*# Output:*

*# Distributor ID: Ubuntu*

*# Description: Ubuntu 22.04.3 LTS*

*# Release: 22.04*

*# Codename: jammy*

### 1.7.3 Fișiere specifice distribuției

Fiecare familie de distribuții are propriile fișiere de identificare:

*# Debian/Ubuntu specific*

`cat /etc/debian_version`

*# Output: 12.4 (pentru Debian) sau jammy/sid (pentru Ubuntu)*

*# RHEL/CentOS/Fedora specific*

`cat /etc/redhat-release`

*# Output: CentOS Linux release 8.5.2111*

*# Toate fișierele \*-release disponibile*

`cat /etc/*-release`



### Exercițiul 1.19:

Identificare completă distribuție:

1. Rulați `cat /etc/os-release` - notați NAME și VERSION\_ID
2. Rulați `hostnamectl` - verificați dacă informațiile coincid
3. Încercați `lsb_release -a` - funcționează? (poate să nu fie instalat)
4. Verificați dacă există `/etc/debian_version` sau `/etc/redhat-release`
5. Din informațiile colectate, determinați familia distribuției (Debian-based sau RHEL-based)

### Exercițiul 1.20:

Arhitectura sistemului:

1. Rulați `uname -m` - notați arhitectura (x86\_64, aarch64, etc.)
2. Dacă sunteți pe Debian/Ubuntu: `dpkg --print-architecture`

## 1.8 Structura distribuțiilor Linux

### 1.8.1 Familii principale

Distribuțiile Linux se grupează în familii bazate pe distribuția originală. Fiecare familie are caracteristici comune:

#### Debian-based:

Distribuții: Debian, Ubuntu, Linux Mint, Kali Linux, Pop!\_OS

Package manager: `apt`, `dpkg`

Fișiere pachete: `.deb`

Directoare config: `/etc/apt/`

#### Red Hat-based:

Distribuții: RHEL (Red Hat Enterprise Linux), CentOS, Fedora, Rocky Linux, AlmaLinux

Package manager: `yum` (vechi), `dnf` (modern), `rpm` (low-level)

Fișiere pachete: `.rpm`

Directoare config: `/etc/yum.repos.d/`

**Notă:** RHEL este standard în telecomunicații și enterprise

#### Arch-based:

Distribuții: Arch Linux, Manjaro

Package manager: `pacman`

Rolling release (updates continue)

#### SUSE-based:

Distribuții: openSUSE, SUSE Enterprise

Package manager: `zypper`

## 1.8.2 Identificare familie distribuție

*# Verificare ID\_LIKE în os-release*

`grep ID_LIKE /etc/os-release`

*# Output exemple:*

*# ID\_LIKE=debian → familia Debian*

*# ID\_LIKE="rhel fedora" → familia Red Hat*

*# ID\_LIKE=arch → familia Arch*

### Comenzi identificare:

*# Debian/Ubuntu - verificare package manager*

`which apt`

*# Output: /usr/bin/apt (dacă există)*

*# RHEL/CentOS - verificare package manager*

`which yum dnf`

*# Output: /usr/bin/dnf (modern) sau /usr/bin/yum (vechi)*

*# Verificare ce sistem de init folosește*

`ps -p 1`

*# Output: systemd (modern) sau init/upstart (vechi)*

### 1.8.3 Diferențe importante între familii

Aspect	Debian-based	Red Hat-based
Instalare pachete	<code>apt install package</code>	<code>dnf install package</code> sau <code>yum install package</code>
Căutare pachete	<code>apt search keyword</code>	<code>dnf search keyword</code>
Update sistem	<code>apt update &amp;&amp; apt upgrade</code>	<code>dnf update</code>
Listare pachete	<code>dpkg -l</code>	<code>rpm -qa</code>
Info pachet	<code>dpkg -L package</code>	<code>rpm -ql package</code>
Web server	Apache = <code>apache2</code>	Apache = <code>httpd</code>
Config locations	<code>/etc/apache2/</code>	<code>/etc/httpd/</code>

#### De ce contează:

Documentația online e adesea specifică distribuției  
 Scripts de automatizare trebuie adaptate  
 Troubleshooting diferă (log locations, service names)  
 Package names diferă pentru același software

#### Exercițiul 1.21:

Identificare familie și caracteristici:

1. Din `/etc/os-release`, identificați ID și ID\_LIKE
2. Determinați familia: Debian, Red Hat, Arch, sau altă?
3. Verificați ce package manager aveți: `which apt dnf yum pacman`

#### Exercițiul 1.22:

Explorare diferențe:

1. Găsiți unde sunt stocate listele de repository-uri:  
`/etc/apt/sources.list` (Debian) SAU  
`/etc/yum.repos.d/` (RHEL)
2. Listați conținutul: `ls -la /etc/apt/` sau `ls -la /etc/yum.repos.d/`
3. Citiți un fișier de repository cu `cat` sau `less`
4. Identificați URL-urile de unde se descarcă pachetele

## 1.9 Quick Reference - Comenzi sistem

Comandă	Informație
<code>uname -r</code>	Kernel version
<code>uname -m</code>	Architecture
<code>uname -a</code>	Complete system info
<code>cat /etc/os-release</code>	Distribution info
<code>hostnamectl</code>	System info complet
<code>lsb_release -a</code>	LSB info (dacă instalat)
<code>grep ID_LIKE /etc/os-release</code>	Familia distribuției
<code>which apt dnf yum</code>	Package manager disponibil