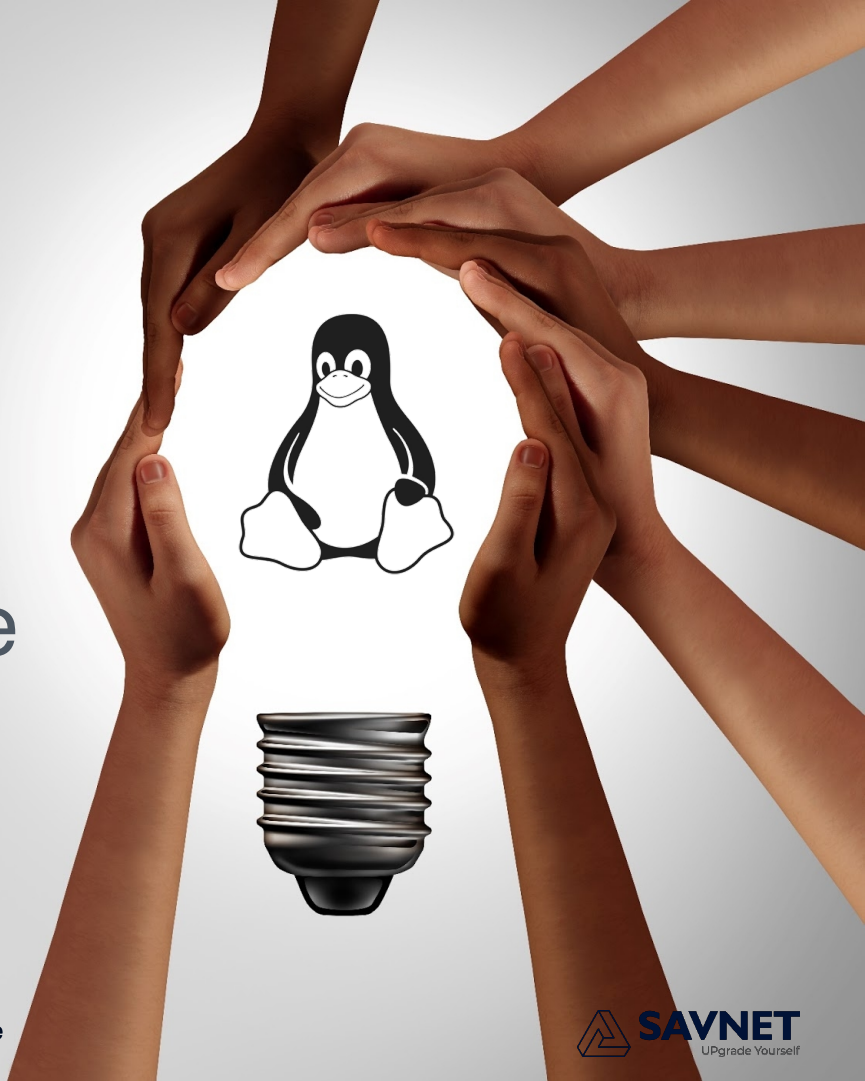


Linux

Administrare și Securitate



De ce LINUX ?

LOCURI DE MUNCĂ

- Administrare LINUX
- Ingineri de Securitate.
- Suport Tehnic.
- Dezvoltator de Sistem Linux.
- Dezvoltatori de Kernel.
- Drive pentru Dispozitive.
- Dezvoltatori de Aplicații.



Companii

Oracle
RedHat
Google
IBM
Facebook
Amazon
DELL
Samsung
Microsoft

...

Copyright © 2019 Network Development Group Inc.

Ce spun statisticile?

Servere & Cloud:

- 49.2% din workload-urile cloud la nivel global rulează pe Linux
- Linux deține 44.8% din piața sistemelor de operare pentru servere
- 59.4% din toate site-urile web folosesc Linux
- 92% din mașinile virtuale pe AWS, Google Cloud și Azure rulează pe Linux

Supercomputing:

- 100% din top 500 supercomputere din lume rulează pe Linux

Dezvoltatori:

- 78.5% din dezvoltatorii la nivel global folosesc Linux
- 90.1% dintre dezvoltatorii cloud-native preferă Linux

Embedded & IoT:

- 39.5% din sistemele embedded folosesc Linux
- 61.3% din smart TV-uri folosesc Linux (Tizen, webOS)

Smartphone:

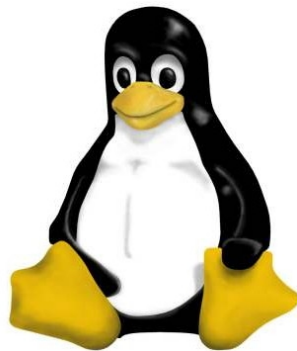
80% din totalul smartphone-urilor la nivel global sunt alimentate de kernel-ul Linux (Android 72.72% - cota de piata)

Istoric LINUX

- Linux a început în 1991 ca un proiect hobby al lui **Linus Torvalds**, un informatician de origine finlandeză care studia la Universitatea din Helsinki.
- Proiectul **GNU** a fost dezvoltat de **Richard Stallman** în 1983.
- Programatorii Linux au putut incorpora instrumentele GNU pentru a oferi un sistem de operare complet.



Linus
Torvalds



Richard Stallman

Linus Torvalds: "I'll Spend Next 25 Years To Help Linux Beat Windows In Desktop War"
... and there GOD said "Let there be FREE Software!"
... and then Richard Stallman said "Let there be FREE Software!"

Distribuții de LINUX



Filozofia Open -Source





Cum se desfășoară cursul?

Experiență Educațională

Teorie	Învățare Individuală	Testare / Examene
<ul style="list-style-type: none">• 4 Sesiuni de Învățare• 10 ore/ predare	<ul style="list-style-type: none">• Exerciții practice studiu individual	<ul style="list-style-type: none">• Evaluare finala - test grila

“Opportunities don’t happen. You create them.” (Chris Grosser)

Ce vom învăța în acest curs?

- Obținerea de Ajutor și utilizarea paginilor manual
- Detectarea versiunilor
- Structura distribuției
- Sistem de fișiere pentru documentare și locații



Capitolul 1: Asistență în Linux



CLI – Command Line Interface

De ce CLI (Command Line Interface)?

Viteză de Execuție

Operațiuni de până la 20x mai rapide decât interfețele grafice

Consum Minimal de Resurse

Utilizare redusă de RAM și CPU - ideal pentru servere și sisteme vechi

Portabilitate Universală

Funcționează pe Linux, macOS, Windows (WSL), BSD și sisteme embedded

Automatizare Completă

Scripturi pentru taskuri repetitive și integrare CI/CD

Control Precis și Granular

Acces la funcționalități avansate indisponibile în GUI

Administrare Remotă Eficientă

Conectare SSH pe conexiuni lente - bandwidth minim

Procesare Batch

Operațiuni simultane pe mii de fișiere într-o singură comandă

Debugging Superior

Output detaliat, log-uri complete, funcționează în recovery mode

Productivitate Maximă

Autocompletare, history, combinarea comenzilor prin pipe-uri

Comenzi de Bază și Orientare în CLI

Scurtături Utile:

TAB → autocompletare comenzi și paths
CTRL + R → search în history comenzi
CTRL + L → curăță terminalul (clear)
↑ / ↓ → navighează prin comenzi anterioare

Orientare în Sistem:

pwd → afișează directorul curent (Present Working Directory)
whoami → afișează utilizatorul curent
Exemplu: whoami → user
hostname → afișează numele sistemului
Exemplu: hostname → server01

Navigare între Directoare:

cd /path → schimbă directorul
Exemplu: cd /var/log
cd ~ sau cd → mergi la HOME directory
Exemplu: cd ~ → /home/user
cd .. → urcă un nivel (director părinte)
cd - → revino la directorul anterior
cd . → rămâi în directorul curent (folosit în scripts)



Obținerea de Ajutor

Filosofia "Self-Service" în administrarea Linux

Realitatea în producție (exemple):

- Servere izolate în DMZ - fără acces internet
- Console mode (KVM/iLO/IPMI) - doar terminal text
- Incident în timpul nopții - soluție rapidă necesară
- Documentația locală - întotdeauna disponibilă

Avantaje documentație built-in:

- Viteză - `man grep` - instant vs Google search - 1-2 minute (găsire, citire, adaptare)
- Acuratețe: Documentația pentru VERSIUNEA instalată
- Acces offline: Funcționează în medii securizate
- Profesionalism: Cerut în certificări RHCSA/LFCS și interviuri

Cazuri de utilizare - Documentație locală

Troubleshooting în Data Center:

Server nu răspunde SSH → console access

Verificare sintaxă tcpdump pentru packet capture

Soluție: `man tcpdump`

Medii securizate (Telecom/Banking):

Policy: NO internet pe servere core

Config iptables pentru traffic shaping

Soluție: `man iptables-extensions + /usr/share/doc/`

Audit de securitate:

Verificare parametri stat, auditd

Sistem izolat - nu poate accesa resurse externe

Soluție: documentație locală completă

Protecție împotriva erorilor:

Comenzi distructive: `rm -rf`, `dd`, `mkfs`

Parametri greșiți → downtime

Side-effects neașteptate



Utilizarea paginilor manual

Man Pages

Documentație creată pentru UNIX, moștenită în Linux
Fiecare comandă are același template
Instalate local împreună cu tool-ul

`man ls`
`man grep`
`man ssh`

Căutare în pagină: `/` apoi pattern, `n` pentru next, `N` pentru previous
leșire: `q`

Man page-ul corespunde EXACT versiunii instalate - Apache 2.2 vs 2.4 au sintaxă diferită
Majoritatea adminilor fac copy/paste direct din secțiunea EXAMPLES
Debugg `/error` în man page pentru exit codes

Structura unei Man Page

Secțiuni standard:

NAME - nume + descriere scurtă (one-liner)
SYNOPSIS - sintaxă și exemple (formatul e standardizat)
DESCRIPTION - explicație detaliată funcționalitate
OPTIONS - lista completă parametri disponibili
EXAMPLES - cazuri practice (cea mai folosită secțiune)
FILES - fișiere de configurare relevante
SEE ALSO - comenzi înrudite
BUGS - known issues (când există)

Convenții SYNOPSIS:

[OPTION] - parametru opțional (brackets)
OPTION... - poate fi repetat
option1|option2 - alege unul dintre (pipe)
<required> - parametru obligatoriu

Organizarea Man Pages în Secțiuni

9 secțiuni standard:

- 1 - General Commands: ls, grep, cp
- 2 - System Calls: open, read, write
- 3 - Library Calls: printf, malloc
- 4 - Special Files: /dev/null, /dev/random
- 5 - File Formats: /etc/passwd, /etc/fstab
- 6 - Games
- 7 - Miscellaneous: ascii, regex, environ
- 8 - System Administration: mount, iptables, useradd
- 9 - Kernel Routines

Specificare secțiune:

`man passwd` → implicit secțiunea 1 (comanda passwd)

`man 5 passwd` → explicit secțiunea 5 (formatul fișierului /etc/passwd)

Secțiunea 5 e esențială pentru configurări - `man 5 fstab`, `man 5 ssh_config` arată exact sintaxa

Când troubleshooting servere, secțiunea 8 e prima verificare

Multe nume apar în multiple secțiuni - crontab există ca și comandă (1) și ca format de fișier (5)

info - Alternative la man pages

Organizare ierarhică cu conținutul tabelor
Cross-references între secțiuni (nodes)
Mai detaliat și mai structurat decât man
Specific tool-uri GNU (bash, coreutils, tar, sed)

info grep → documentație grep
info coreutils → TOATE comenzile core (ls, cp, mv, etc.)
info bash → documentație completă bash scripting

Arrow - keys pentru scroll,
Shift+H - pentru help complet,
Q - pentru quit
n - next node
p - previous node
u - up level

Man pages mai popular în practică

Info mai bun pentru învățare în profunzime

Pentru tools complexe (tar, sed, awk) info e mai ușor de urmărit decât man

--help - Quick Reference

ls --help
grep --help

Quick reference cu sintaxă + opțiuni principale

Mai rapid decât `man`

Nu toate comenzile suportă `--help` (unele au `-h`, altele nimic)

whatis - Identificare rapidă scop comandă

Afișează NAME section din [man](#) page - descriere scurtă, o singură linie

```
~$ whatis ls
ls (1)          - list directory contents
~$ whatis cp mv rm
cp (1)          - copy files and directories
mv (1)          - move (rename) files
rm (1)          - remove files or directories
```

Numărul în paranteză reprezintă secțiunea [man](#) page (1 = user commands)

Verificare funcționalitate comandă cunoscută

apropos - Găsește comanda potrivită

Detectare utilitare necunoscute în instalație

Căutare keyword în NAME section din toate man pages

Mapare cerință funcțională → utilitare disponibile în sistem

`apropos copy` → găsește cp, rsync, cpio, dd

`apropos "network traffic"` → query multi-word (ghilimele obligatorii)

`man -k keyword` → alias pentru `apropos`

`apropos -s 8 firewall` → filtrare pe secțiune specifică (admin tools)

`apropos -a network monitor` → operație AND (ambele keywords obligatorii)

Database mandb: refresh nightly (cron), manual rebuild via `sudo mandb`

Error "`nothing appropriate`" → database lipsă sau neactualizată

Discovery workflow: `apropos network | grep monitor` → identificare tool-uri disponibile



Locuri comune pentru documentație

Documentație adițională în sistem

De ce există documentație locală?

- Fiecare pachet instalat vine cu documentația proprie
- Developers includ README-uri, FAQ-uri, exemple de configurare
- Documentație specifică distribuției folosite (Debian \neq RedHat)
- Disponibilă offline, fără dependență de internet

Locații standard:

- `/usr/share/doc/[package]/`
- `/usr/doc/` (locație alternativă pe unele distro)

Ce putem găsi în acele locații:

- README – instrucțiuni de setare specifice distribuției
- examples/ - templates funcționale, testate
- changelog - breaking changes între versiuni
- FAQ - probleme comune
- INSTALL - proceduri instalare/configurare
- COPYING / LICENSE - informații licență



Detectarea Versiunii

Detectarea versiunii - OS, Kernel, Distribuție

Verificare compatibilitate pachete/software

Troubleshooting: bug-uri specifice versiuni kernel

Documentare infrastructură: ce versiuni rulează

```
~$ uname -a
Linux Titi 6.8.0-90-generic #91~22.04.1-Ubu
ntu SMP PREEMPT_DYNAMIC Thu Nov 20 15:20:45
UTC 2 x86_64 x86_64 x86_64 GNU/Linux
```

Versiunea kernel:

`uname -r` → versiune kernel (5.15.0-91-generic)

`uname -a` → info completă sistem (kernel, hostname, arhitectură)

Versiunea distribuție:

`cat /etc/os-release` → info distribuție (NAME, VERSION, ID)

`lsb_release -a` → LSB info (dacă e instalat)

Fișiere sistem pentru identificare:

`/proc/version` → kernel version și compiler info

`/etc/debian_version` → specific Debian/Ubuntu

`/etc/redhat-release` → specific RHEL/CentOS/Fedora

Arhitectură sistem:

`uname -m` sau `arch` → arhitectură (x86_64, aarch64, i686)

`dpkg --print-architecture` → Debian/Ubuntu



Structura distribuției

Structura distribuției - Familii și identificare

Familii principale Linux:

Debian-based: Ubuntu, Debian, Linux Mint, Kali

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

Arch-based: Arch Linux, Manjaro

SUSE-based: openSUSE, SUSE Enterprise

Comenzi identificare distribuție:

`cat /etc/os-release` → ID="ubuntu" sau ID="centos" sau ID="debian"

`hostnamectl` → system info complet (OS, kernel, architecture)

`cat /etc/*-release` → toate fișierele de release disponibile

Diferențe cheie între familii:

Package manager: `apt` (Debian) vs `yum/dnf` (RHEL) vs `pacman` (Arch)

Locații config: `/etc/apt/` (Debian) vs `/etc/yum.repos.d/` (RHEL)

Package names: același software, nume diferite (apache2 vs httpd)

Documentație specifică: man pages pot diferi între distribuții

Troubleshooting: soluții specifice fiecărei familii

Community support: forumuri și resurse specifice distribuției



Sistem de fișiere pentru documentare

Conținut și utilizare documentație

Workflow în producție:

1. După instalare nouă

- `ls /usr/share/doc/nginx/`
- citit README pentru setup specific

2. Înainte de upgrade major

- check CHANGELOG pentru breaking changes in config syntax

3. Troubleshooting escalation

- `--help` (2s) → `man nginx` (2m) → `/usr/share/doc/nginx/examples` (5m) → Google (ultimă opțiune)

```
~$ ls /usr/share/doc/nginx  
changelog.Debian.gz  copyright
```

Exemplu practic:

`ls /usr/share/doc/openssh-server/` → README.Debian, examples/, changelog.gz
`ls /usr/share/man/`
`ls /usr/share/info/`



Demo: Workflow

Ce am învățat în acest capitol?

Obținerea de Ajutor

Cum găsim informații despre comenzi necunoscute: `man`, `whatis`, `apropos`, `whereis`, `--help` și `/usr/share/doc/`

Detectarea Versiunii

Identificarea sistemului: ce kernel, ce distribuție, ce arhitectură folosim (`uname`, `/etc/os-release`, `hostnamectl`)

Locuri comune pentru documentație

Documentatii specifice, documente share-uite



Q&A



Activități pentru acasă



Ce urmează?



Gestionarea Fișierelor și Directoarelor

Înțelegerea Hardware-ului Computerului

Introducere in scripting

•

