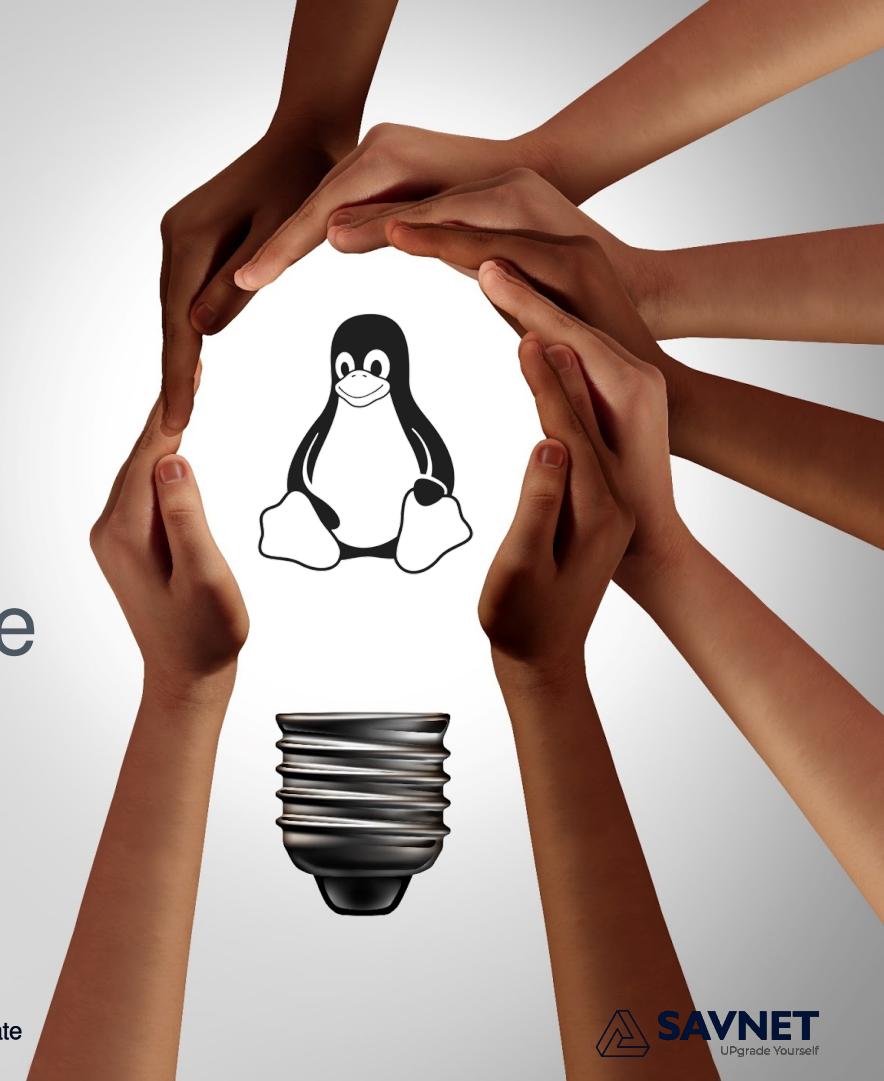


Linux

Administrare și Securitate



Ce vom învăța în acest curs?

- Elementele hardware relevante pentru Linux
- Cum vedem în Linux aceste componente
- Interacțiunea hardware-software
- Dispozitive embedded, interfețe de comunicație, IoT



Capitolul 4: Înțelegerea Hardware-ului



Elementele hardware relevante pentru Linux



Identificarea Hardware-ului și Vizualizarea Informațiilor despre CPU

Plăci de Bază (Motherboards)

Placa de bază este placa hardware principală din computer care conectează unitatea centrală de procesare (CPU), memoria cu acces aleatoriu (RAM) și alte componente.

Caracteristici:

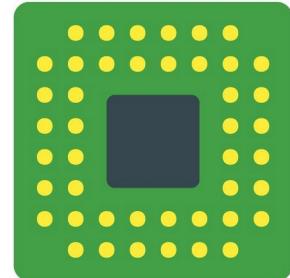
- Hardware-ul comunică prin intermediul plăcii de bază sau prin intermediul magistralelor de sistem (system buses).
- Toate componentele principale sunt conectate la placa de bază
- Magistrala de sistem funcționează ca o "autostradă" pentru comunicarea între componente

Procesoare (Processors)

Unitatea Centrală de Procesare (CPU):

CPU (cunoscut și ca procesor) efectuează deciziile și calculele pentru sistemul de operare. Este conectat la alte componente hardware prin intermediul plăcii de bază.

- Multiprocessor - sistem cu mai mult de un CPU
- Multi-core - mai mult de un procesor pe un singur chip



Procesoare - Tipuri

Arhitecturi x86 (CISC) - Cele mai răspândite pe desktop și server

- x86 (32 bit):
 - Inventat în 1978
 - Limitat la 4GB RAM
- x86_64 (64 bit):
 - Suportă mai mult de 4GB RAM
 - Arhitectură modernă pentru sisteme actuale

Arhitecturi ARM (RISC) - Domină piața de mobile și embedded

ARMv7 (32-bit) – dispozitive mai vechi

ARMv8 / ARMv9 (64-bit) – telefoane moderne, servere ARM

Producători: Apple, Qualcomm, Samsung, Broadcom

Memoria RAM (Random Access Memory)

Ce este RAM?

Memoria cu acces aleatoriu (RAM) este locul unde sunt stocate datele programelor și ale sistemului.

Memorie virtuală (Virtual Memory):

- Swap space - tip de sistem de fișiere sau fisier folosit când RAM-ul disponibil este redus
- Permite sistemului să funcționeze când RAM-ul fizic este insuficient

Vizualizare utilizare memorie:

Comanda `free` poate afișa utilizarea memoriei



```
~$ free -m
              total        used        free      shared    buffers    cached
Mem:       128920       56000      72919          0       1181     12110
-/+ buffers/cache:    42709      86211
Swap:      131050           3      131047
```

Magistrale (Buses)

Ce sunt Buses?

Conexiune de mare vitează care permite comunicarea între computere sau componente interne

Dispozitive periferice:

Dispozitive neatașate direct la placă de bază

Două tipuri comune:

- Peripheral Component Interconnect (PCI):

- Comandă vizualizare:

[lspci](#)

- Universal Serial Bus (USB):

- Comandă vizualizare:

[lsusb](#)

Hard Disk-uri (Hard Drives)

Dispozitive de stocare:

Numite și dispozitive disk (disk devices)

Pot fi conectate direct la placa de bază, la PCI sau USB

Partiționare:

Disk-urile sunt împărțite în partiții - diviziuni logice care iau spațiul mare de stocare și îl împart în bucăți mai mici.

Două tipuri de partiționare:

- Master Boot Record (MBR) - tehnologie mai veche
- GUID Partitioning Table (GPT) - tehnologie mai nouă

Disk-uri SSD (Solid State Disks)

Tip de hard disk fără piese mobile sau discuri rotative, este efectiv RAM sau memorie

Avantaje:

- Consum mai mic de energie
- Mai puțină căldură și vibrație
- Timp mai scurt pentru boot-area sistemului și încărcarea programelor

Dezavantaje:

- Capacitate mai mică datorită costului mai mare
- Fără posibilitate de upgrade dacă este lipit pe placa de bază

Gestionarea Dispozitivelor

Cum gestionează distribuțiile Linux dispozitivele?

Sunt dispozitivele compatibile?

Drive (Drivers):

Dispozitivele hardware au nevoie de software, numit drivere, care le permite să comunice cu sistemul de operare (OS).

- Linux include drivere pentru majoritatea hardware-ului comun
- Unele dispozitive necesită drivere proprietare

Plăci rețea - Ethernet (LAN – Local Area Network)

Cele mai comune, utilizate pentru rețele cablate.

Standard: IEEE 802.3

Viteze: 10/100 Mbps, 1 Gbps, 10 Gbps, 25/40/100 Gbps

Conectori: RJ-45 (cupru) sau SFP+/QSFP+ (fibră optică)

Utilizare: PC-uri, servere, switch-uri

```
~$ ip link | grep enp
2: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel stat
e UP mode DEFAULT group default qlen 1000
```

Plăci rețea - Wireless (Wi-Fi)

Pentru conexiuni fără fir.

Standard: IEEE 802.11 a/b/g/n/ac/ax (Wi-Fi 4/5/6/6E)

Frecvențe: 2.4 GHz, 5 GHz, 6 GHz

Pot fi interne (PCIe/M.2) sau externe (USB)

```
~$ ip link | grep wlp
3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
mode DORMANT group default qlen 1000
```



Cum vedem in Linux aceste componente hardware?

Filozofia Linux - Totul este tratat ca un fișier

Conceptul fundamental în Linux: "Everything is a file"

Hardware-ul NU este o excepție!

- Hard disk-ul? → /dev/sda
- Mouse-ul? → /dev/input/mouse0
- Placa de rețea? → /dev/eth0
- Procesorul? → /proc/cpuinfo
- Memoria? → /proc/meminfo

Linux tratează hardware-ul ca pe niște fișiere pe care le poți citi, scrie sau interoga.

Avantaj: Poți investiga tot sistemul cu comenzi simple.

Directoarele Magice

Unde "locuiește" hardware-ul în Linux: **/dev (devices)**

- Fișiere speciale pentru dispozitive
- /dev/sda, /dev/ttyUSB0, /dev/video0

/proc (process information)

- Info despre procese și sistem
- /proc/cpuinfo, /proc/meminfo

/sys (sysfs)

- Structura hardware-ului, drivere, module kernel
- /sys/class/net/eth0

```
cat /proc/cpuinfo  
# Citești info CPU ca pe un fișier text!
```

```
~$ cat /proc/cpuinfo  
processor      : 0  
vendor_id     : GenuineIntel  
cpu family    : 6  
model         : 142  
model name    : Intel(R) Core(TM) i5-7300U CPU @ 2.60GHz  
stepping       : 9  
microcode     : 0xf6  
cpu MHz        : 3499.985  
cache size    : 3072 KB  
physical id   : 0  
siblings       : 4  
core id        : 0  
cpu cores     : 2  
apicid         : 0  
initial apicid: 0  
fpu            : yes  
fpu_exception  : yes  
cpuid level   : 22  
wp             : yes
```

lshw - Inventarul Complet

lshw - List Hardware

Scanează și afișează tot hardware-ul detectat de sistem

Comenzi utile:

```
sudo lshw          # Tot  
sudo lshw -short    # Format compact  
sudo lshw -class disk # Doar storage  
sudo lshw -class network # Doar rețea
```

Output arată: CPU, RAM, disk-uri, plăci de rețea, USB

Exemplu IoT:

Verifici ce interfețe WiFi/Bluetooth sunt disponibile pe Raspberry Pi înainte să configurezi gateway-ul

lsblk - Hartă Stocare

lsblk - List Block devices

Afișează toate dispozitivele de stocare și structura lor

Arată în format arbore:

```
sda          # Disk fizic
├─sda1       # Partiția 1
└─sda2       # Partiția 2
mmcblk0      # SD card (embedded)
├─mmcblk0p1
└─mmcblk0p2
```

Relevanță embedded:

În Raspberry Pi sau device-uri IoT, vezi imediat SD card-ul (mmcblk0), USB stick-uri pentru logging, spațiu disponibil pentru date senzori

lspci - Magistrala PCI

Lspci - listează tot ce e conectat la magistrala PCI: plăci video, rețea, sound, controlere

```
lspci          # Lista simplă  
lspci -v       # Detalii  
lspci -k       # Cu driverele folosite
```

- Identificare placă de rețea exactă, verificare ce driver rulează, debugging hardware necunoscut

```
~$ lspci  
00:00.0 Host bridge: Intel Corporation Xeon E3-1200 v6/7th Gen Core Processor Host Bridge/DRAM Registers (rev 02)  
00:02.0 VGA compatible controller: Intel Corporation HD Graphics 620 (rev 02)  
00:04.0 Signal processing controller: Intel Corporation Xeon E3-1200 v5/E3-1500 v5/6th Gen Core Processor Thermal Subsystem (rev 02)  
00:14.0 USB controller: Intel Corporation Sunrise Point-LP USB 3.0 xHCI Controller (rev 21)  
00:14.2 Signal processing controller: Intel Corporation Sunrise Point-LP Thermal subsystem (rev 21)  
00:15.0 Signal processing controller: Intel Corporation Sunrise Point-LP Serial IO I2C Controller #0 (rev 21)  
00:15.1 Signal processing controller: Intel Corporation Sunrise Point-LP Serial IO I2C Controller #1 (rev 21)  
00:16.0 Communication controller: Intel Corporation Sunrise Point-LP CSME HECI #1 (rev 21)  
00:17.0 RAID bus controller: Intel Corporation 82801 Mobile SATA Controller [RAID mode] (rev 21)  
00:1c.0 PCI bridge: Intel Corporation Sunrise Point-LP PCI Express Root Port #1 (rev f1)  
00:1c.2 PCI bridge: Intel Corporation Sunrise Point-LP PCI Express Root Port #3 (rev f1)  
00:1f.0 ISA bridge: Intel Corporation Sunrise Point LPC Controller/eSPI Controller (rev 21)  
00:1f.2 Memory controller: Intel Corporation Sunrise Point-LP PMC (rev 21)  
00:1f.3 Audio device: Intel Corporation Sunrise Point-LP HD Audio (rev 21)  
00:1f.4 SMBus: Intel Corporation Sunrise Point-LP SMBus (rev 21)  
00:1f.6 Ethernet controller: Intel Corporation Ethernet Connection (4) I219-LM (rev 21)  
01:00.0 Unassigned class [ff00]: Realtek Semiconductor Co., Ltd. RTS525A PCI Express Card Reader (rev 01)  
02:00.0 Network controller: Intel Corporation Wireless 8265 / 8275 (rev 78)
```

dmidecode - Secretele din BIOS

dmidecode - DMI table decoder

Citește informații low-level din BIOS/UEFI despre hardware

Comenzi targetate:

```
sudo dmidecode -t processor    # Info CPU (frecvență, cache)  
sudo dmidecode -t memory       # RAM (tip, viteză, sloturi)  
sudo dmidecode -t system       # Producător, model, serial
```

De ce e util?

- Află capacitatea MAX de RAM
- Verifică ce tip de memorie să cumperi
- Serial numbers pentru inventar echipamente server

Note: Necesară sudo, nu funcționează în VM-uri

lsusb - Dispozitive USB

lsusb - List USB devices

Listează toate dispozitivele USB conectate

Output:

```
Bus 001 Device 003: ID 046d:c52b Logitech, Inc. Mouse  
Bus 001 Device 004: ID 0bda:8179 Realtek WiFi adapter
```

Comenzi utile:

```
lsusb -v          # Verbose (detalii complete)  
lsusb -t          # Arată ca arbore (hub-uri)
```

Critical pentru IoT:

- Senzori pe USB/Serial (ttyUSB0), adaptoare WiFi/4G, Arduino/ESP32 conectate



Interacțiunea hardware-software

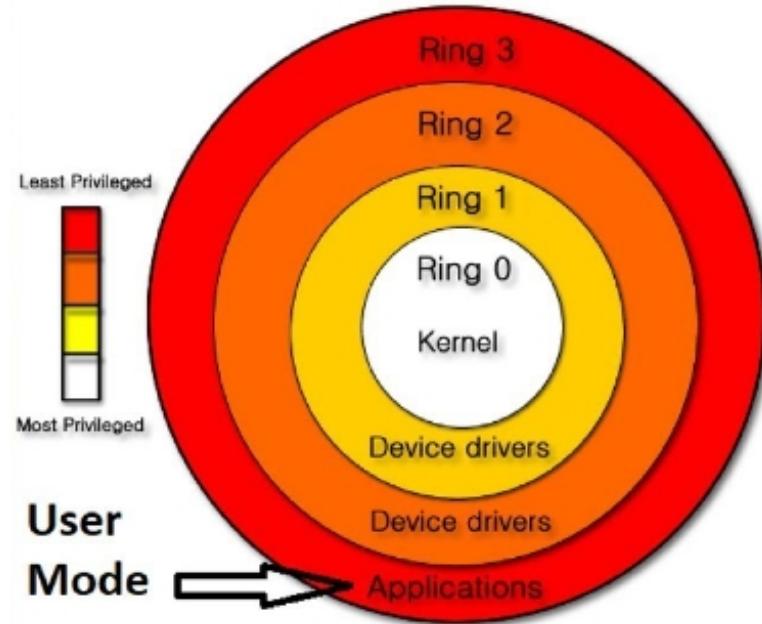
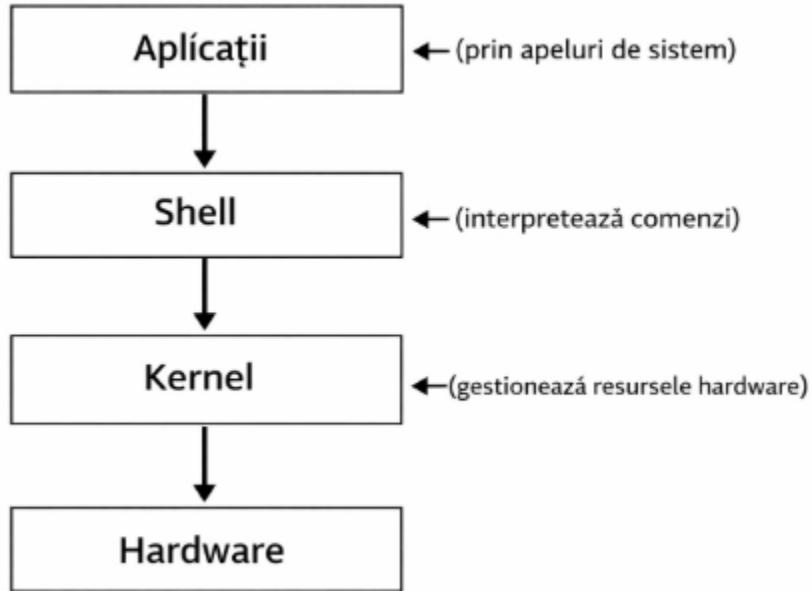


Kernel



Generated by AI

Interacțiuni în sistem de operare



Procesele Kernel

O funcție cheie a kernel-ului Linux este gestionarea proceselor.

- Kernel-ul acceptă comenzi și gestionează procesele care execută acele comenzi.
- Kernel-ul oferă comenzilor acces la dispozitive precum memoria, discurile, interfețele de rețea, tastaturile, mouse-urile, monitoarele și multe altele.
- Kernel-ul oferă, de asemenea, acces la informații despre procesele active printr-un sistem de fișiere pseudo, vizibil în directorul /proc.
- Alte sisteme de fișiere pseudo includ /dev și /sys, care oferă informații despre dispozitivele hardware.

Sistemele de fișiere pseudo sunt cele care par a fi fișiere reale pe disc, dar există doar în memorie.

Directorul /proc

Directorul /proc nu conține doar informații despre procese (după cum sugerează numele "proc")
Oferă și informații despre hardware-ul sistemului și configurația curentă a kernel-ului.

Output-ul afișează o varietate de directoare numerotate și denumite:

```
student@localhost:~$ ls /proc
1          cpuinfo      irq        modules      sys
128         crypto       kallsyms   mounts      sysrq-trigger
17          devices      kcore      mtrr        sysvipc
21          diskstats   key-users   net        thread-self
```

Directorul /proc

Unele dintre comenzi care citesc din `/proc` includ: `top`, `free`, `mount`, `umount`

Există, de asemenea, fișiere regulate importante în directorul /proc, cum ar fi:

- `/proc/cmdline` - Conține informații transmise kernel-ului în timpul boot-ului
- `/proc/meminfo` - Conține informații despre utilizarea memoriei de către kernel
- `/proc/modules` - Conține lista modulelor încărcate în kernel

Ierarhia Proceselor

Când kernel-ul finalizează încărcarea în timpul boot-ului, acesta pornește procesul **init** sau mai modern **systemd** și îi atribuie un **PID** de 1.

Acest proces pornește apoi alte proceze de sistem și atribuie un PID în ordine secvențială.

Când un proces pornește un alt proces, primul proces este numit **proces părinte**.

Al doilea proces este numit **proces copil**.

Ierarhia Proceselor

Procesele pot fi mapate într-un "arbore" care poate fi vizualizat cu comanda **pstree**

```
~$ pstree
systemd--NetworkManager---2*[{NetworkManager}]
|   accounts-daemon---2*[{accounts-daemon}]
|   acpid
|   bluetoothd
|   colord---2*[{colord}]
|   cron---cron---sh---.monitordrive.s---.monitordrive.s---.syncdrive.sh---rclone---7*[{rclone}]
|                           |   \_inotifywait
|   dbus-daemon
|   gdm3---gdm-session-wor---gdm-wayland-ses---gnome-session-b---2*[{gnome-session-b}]
|                           |   \_2*[{gdm-wayland-ses}]
|                           \_2*[{gdm-session-wor}]
|                               \_2*[{gdm3}]
|   geoclue---2*[{geoclue}]
|   gnome-keyring-d---3*[{gnome-keyring-d}]
|   inetd
|   irqbalance---{irqbalance}
|   2*[{kerneloops}]
```

Vizualizarea Instantaneelor Proceselor

O altă modalitate de a vizualiza procesele este cu comanda **ps**

În mod implicit, **ps** va afișa doar procesele în curs de execuție cu opțiunea aux se afisează toate. Comanda **ps** poate fi, de asemenea, folosită cu comenziile **head** și **grep** pentru a filtra procesele afișate:

```
~$ ps aux | head -10
USER        PID %CPU %MEM      VSZ      RSS TTY      STAT START   TIME COMMAND
root          1  0.0  0.0 168056 12932 ?        Ss  08:48  0:02 /sbin/init
root          2  0.0  0.0      0      0 ?        S  08:48  0:00 [kthreadd]
root          3  0.0  0.0      0      0 ?        S  08:48  0:00 [pool_workqueue_release]
root          4  0.0  0.0      0      0 ?        I< 08:48  0:00 [kworker/R-rcu_g]
root          5  0.0  0.0      0      0 ?        I< 08:48  0:00 [kworker/R-rcu_p]
root          6  0.0  0.0      0      0 ?        I< 08:48  0:00 [kworker/R-slub_]
root          7  0.0  0.0      0      0 ?        I< 08:48  0:00 [kworker/R-netns]
root          9  0.0  0.0      0      0 ?        I< 08:48  0:00 [kworker/0:0H-events_highpri]
root         12  0.0  0.0      0      0 ?        I< 08:48  0:00 [kworker/R-mm_pe]
```

Vizualizarea Proceselor În Timp Real

Comanda **top** are o interfață dinamică, bazată pe ecran, care va actualiza periodic output-ul proceselor în execuție.

top - 11:34:27 up 2:46, 1 user, load average: 0,64, 0,97, 1,08												
Tasks: 268 total, 1 running, 267 sleeping, 0 stopped, 0 zombie												
%Cpu(s): 2,1 us, 1,1 sy, 0,0 ni, 96,0 id, 0,7 wa, 0,0 hi, 0,1 si, 0,0 st												
MiB Mem : 15874,5 total, 7412,7 free, 3752,9 used, 4709,0 buff/cache												
MiB Swap: 2048,0 total, 2048,0 free, 0,0 used. 11023,3 avail Mem												
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND	
8176	cristi	20	0	562064	59184	45032	S	2,7	0,4	0:03.14	gnome-terminal-	
1927	cristi	9	-11	2029208	39300	28548	S	2,3	0,2	3:23.79	pulseaudio	
3996	cristi	20	0	3015236	376368	203900	S	2,3	2,3	3:58.91	spotify	
2153	cristi	20	0	5583660	336232	185572	S	2,0	2,1	6:28.06	gnome-shell	
4082	cristi	20	0	1409,2g	390680	143612	S	1,3	2,4	10:38.99	spotify	
4023	cristi	20	0	1717524	153316	98980	S	1,0	0,9	1:23.16	spotify	
11933	cristi	20	0	1263184	41600	27008	S	0,7	0,3	0:00.26	rclone	
54	root	-51	0	0	0	0	S	0,3	0,0	0:18.74	irq/9-acpi	
161	root	-51	0	0	0	0	S	0,3	0,0	0:21.10	irq/51-DLL07A8:01	
2732	cristi	20	0	49,0g	505744	309968	S	0,3	3,1	2:47.84	chrome	
2789	cristi	20	0	48,4g	149620	111628	S	0,3	0,9	0:27.94	chrome	
© Sa	2023 crist	20	Confidential	1815812	897296	183860	Securitate	0,3	5,5	5:30.52	soffice.bin	
6119	cristi	20	0	1393,8g	287944	142516	S	0,3	1,8	2:03.36	chrome	

Vizualizarea Memoriei

Pentru a vizualiza un instantaneu al memoriei utilizate în acel moment, folosiți comanda **free**:

	total	used	free	shared	buff/cache	available
Mem:	16255516	3903396	7492792	816316	4859328	11199180
Swap:	2097148	0	2097148			

Output-ul de mai sus explicat:

Mem: - statisticile pentru memoria fizică din sistem

buff/cache: - memoria fizică minus memoria folosită de kernel

Swap: - memoria virtuală

Fișierele Jurnal (Log Files)

Procesele care rulează pe un sistem produc output care descrie ce face procesul.

- O parte din output merge la terminal, însă alt output nu este văzut în terminal și este scris în fișiere ca mesaje jurnal (sau date jurnal).
- Unele procese înregistrează date în mod implicit, în timp ce altele folosesc un daemon (un proces care rulează în fundal, fără interacțiune directă cu utilizatorul, și care oferă servicii sistemului sau altor aplicații)
- Exemple de daemon-uri includ: syslogd, klogd, rsyslogd, journald
- Fișierele jurnal sunt plasate în directorul /var/log.

Fișierele Jurnal (Log Files)

Pentru a vizualiza fișierele jurnal:

- Folosiți comanda `cat` sau `less`
- Folosiți comanda `journalctl`
- Fișierele jurnal sunt *rotated*, ceea ce înseamnă că fișierele jurnal mai vechi sunt redenumite și înlocuite cu fișiere jurnal mai noi.
- Majoritatea fișierelor jurnal conțin text, care poate fi vizualizat în siguranță cu multe instrumente. Alte fișiere, cum ar fi `/var/log/btmp` și `/var/log/wtmp`, conțin date binare.
- Folosiți comanda `file` pentru a vizualiza fișierele jurnal binare.

Mesajele Kernel

Mesajele kernel pot fi găsite în următoarele fișiere:

/var/log/dmesg - conține mesajele kernel produse în timpul pornirii sistemului

/var/log/messages - va conține mesajele kernel produse pe măsură ce sistemul rulează

Pentru a vizualiza mesajele generate de kernel, folosiți comanda **dmesg**. Pentru a filtra output-ul, folosiți un pipe cu comanda **less** sau **grep**:

```
~$ dmesg | grep -i usb
usbcore: registered new interface driver usbfs
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
```



Drive și Module Kernel

Drivere și Module Kernel

Driver - cod care traduce între hardware și sistem

Lanțul de comunicare: Hardware → Driver → Kernel → Aplicație

În Linux, majoritatea driverelor sunt kernel modules:

- Module - bucăți de cod încărcate/descărcate dinamic
- Nu trebuie să recompilezi tot kernel-ul

```
lsmod                      # Module încărcate acum  
modinfo iwlwifi            # Info despre un modul (WiFi)  
lspci -k                    # Arată ce driver folosește hardware-ul
```

Exemplu real:

```
lsmod | grep bluetooth    # Verifică dacă Bluetooth e activ
```



Dispozitive embedded, interfețe de comunicatie, IoT

Relevantă Telecomunicații și IoT

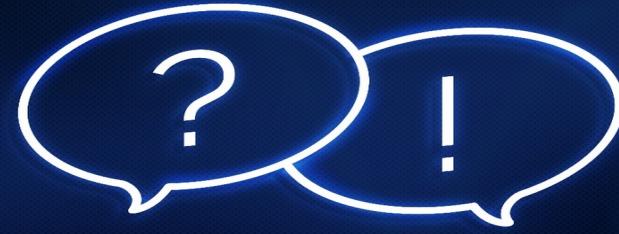
Dispozitive embedded (Raspberry Pi, BeagleBone, ESP32):

Interfețe critice:

- Serial ports (RS-232, RS-485) → senzori industriali → `ls /dev/ttyUSB*`
- I2C, SPI → senzori temperatura, presiune → `ls /dev/i2c-*`
- GPIO → control LED-uri, relee → `/sys/class/gpio/`
- Network → WiFi, 4G/5G, LoRa → `lshw -class network`
- Storage → SD card pentru logging → `lsblk` (vezi `mmcblk0`)

In scenarii reale:

- Gateway IoT nu vede senzorii? → `lsusb`, `dmesg`
- WiFi nu merge? → `lspci -k`, `lsmod | grep wifi`
- SD card plin? → `lsblk`, `df -h`



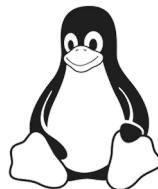
Q&A



Activități pentru acasă



Ce urmează?



Configurarea Rețelei



SAVNET
UPgrade Yourself