

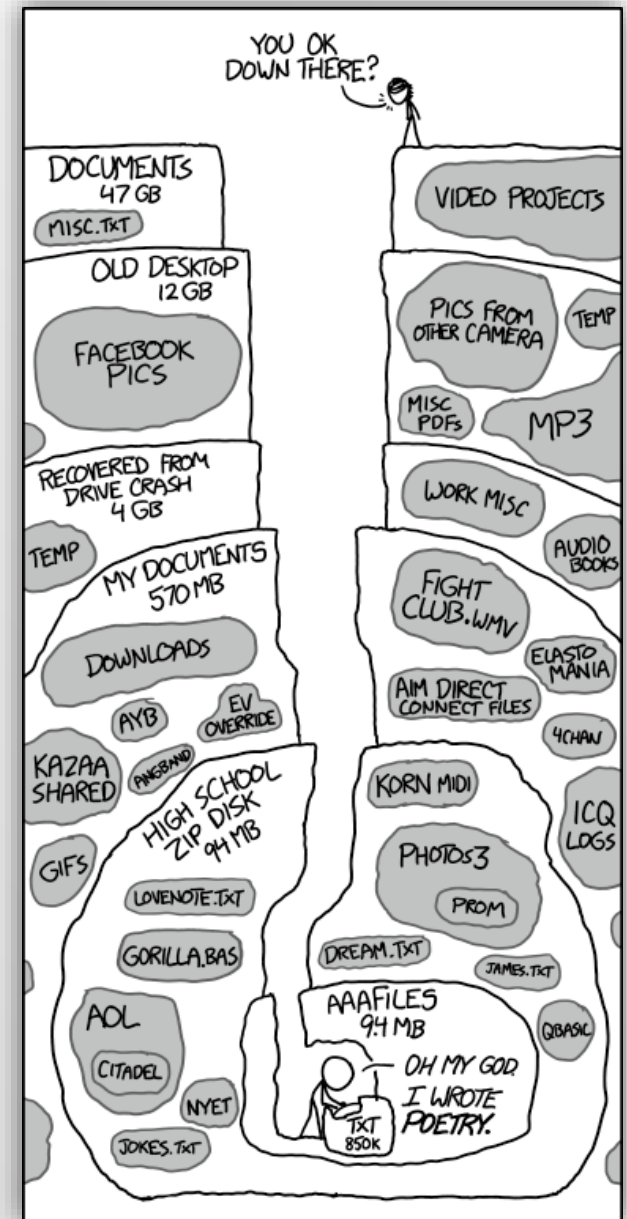
Cursul #2

Procese



Recapitulare

When looking for an old file...



Multitasking:

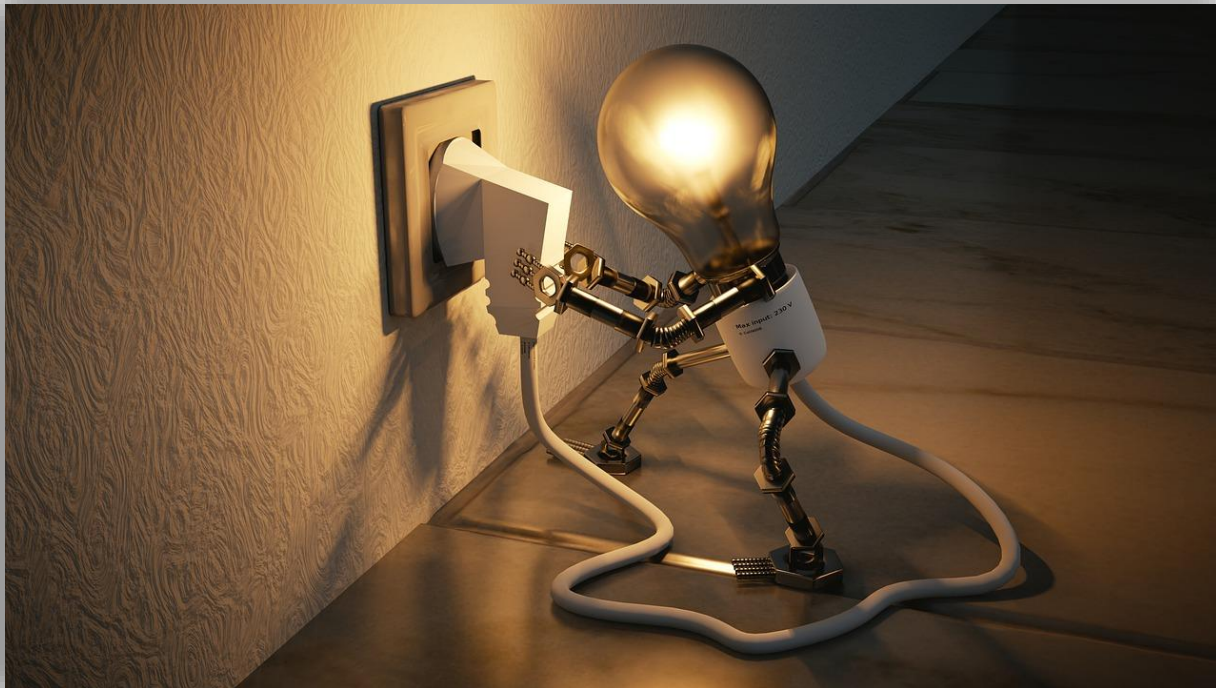
A polite way of telling someone
you haven't heard a word they said

Suport de curs

- Utilizarea Sistemelor de Operare
 - Capitolul 4 - Procese

Ce este un proces ?

- Proces = acțiunea utilizatorului tradusă în logica mașinii



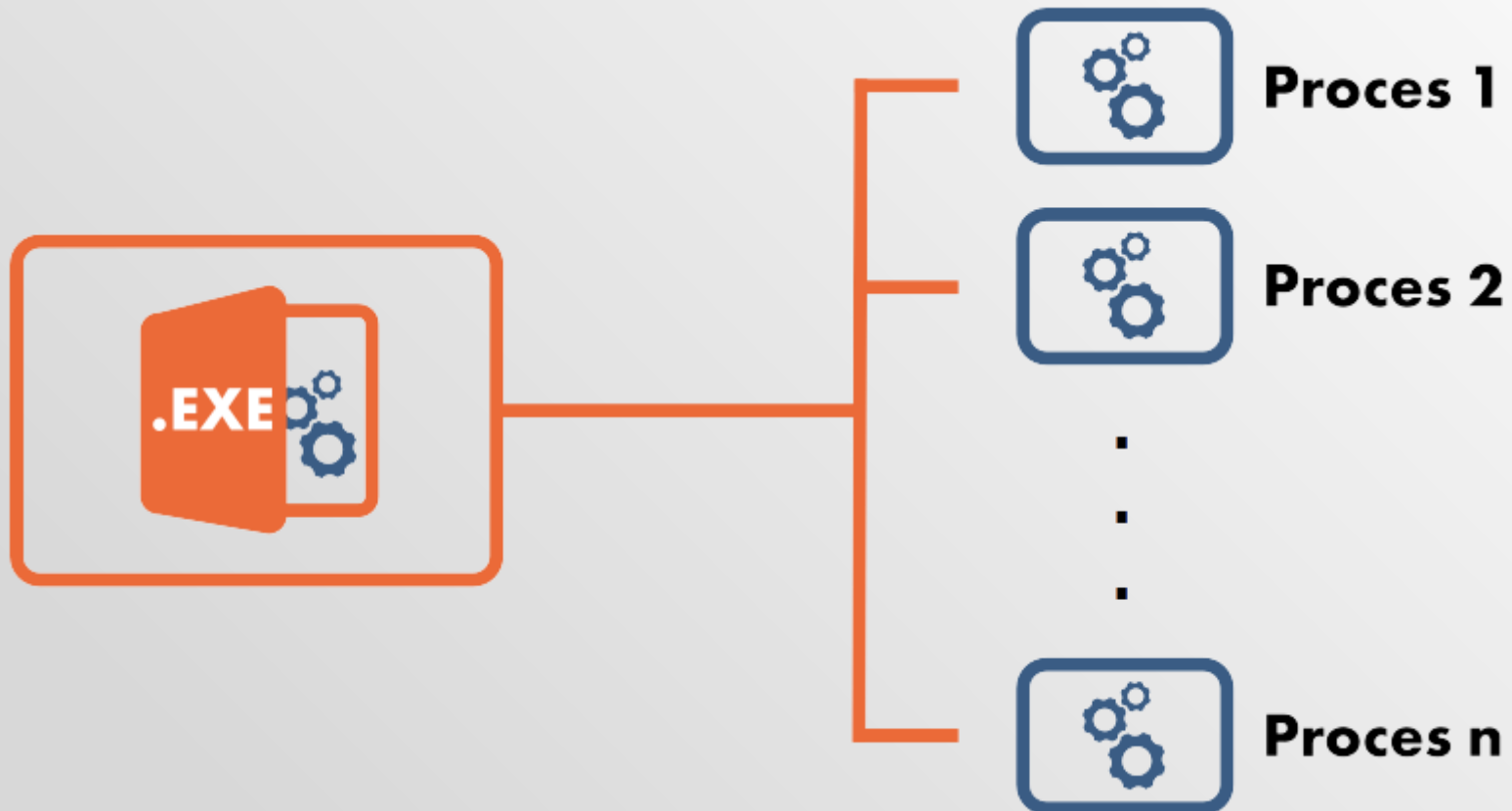
Ce este un proces ?

- Utilizatorul comandă o **acțiune** care declanșează **procese**
- Procesele sunt aplicații ale SO ce pun în practică solicitarea utilizatorului:
 - Procesele sunt legate de utilizatorul care le-a inițiat
 - Procesele folosesc **resurse** ale sistemului (procesor, memorie, spațiu pe disc etc)
- Exemplu: **init**, **bash**, **firefox**

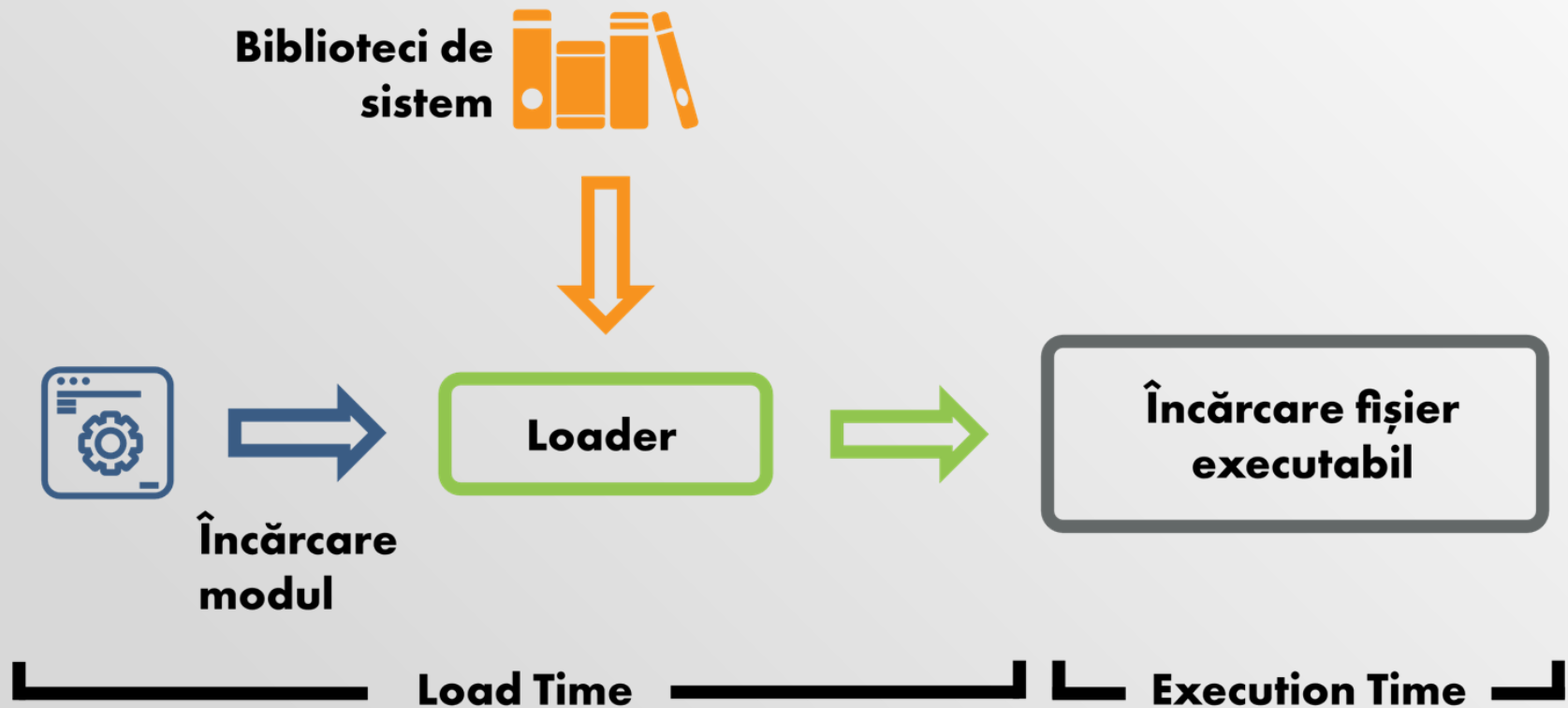
De ce ?

- **Concept fundamental** în sistemele de operare prin care sunt definite **acțiunile**
- Procesul poate fi numit aplicație, task, job sau, nu foarte corect, „program”
 - **Programul** este, tehnic, executabilul din care se naște procesul
 - Procesul ia naștere dintr-un executabil după ce se încarcă executabilul în memorie

Procese în execuție



Lansarea unui proces



La ce bun ?

- Utilizator
 - **Depanarea** sistemului prin examinarea proceselor
 - Procesele pot cauza probleme prin **consumul excesiv** de resurse
 - Identificarea proceselor malițioase
- Dezvoltator
 - Iluzia de **multitasking**
 - Proiectarea unor aplicații **eficiente**

Ce poate face un proces ?

- Ce poate face o rachetă de tenis ?



Ce poate face un proces ?

- Orice proces aparține unui utilizator
- Depinde ce privilegii și permisiuni are pe resurse
 - Un proces poate crea un fișier, sau nu
 - Un proces poate trimite pachete pe rețea, sau nu
 - Privilegiile unui proces sunt date de utilizator

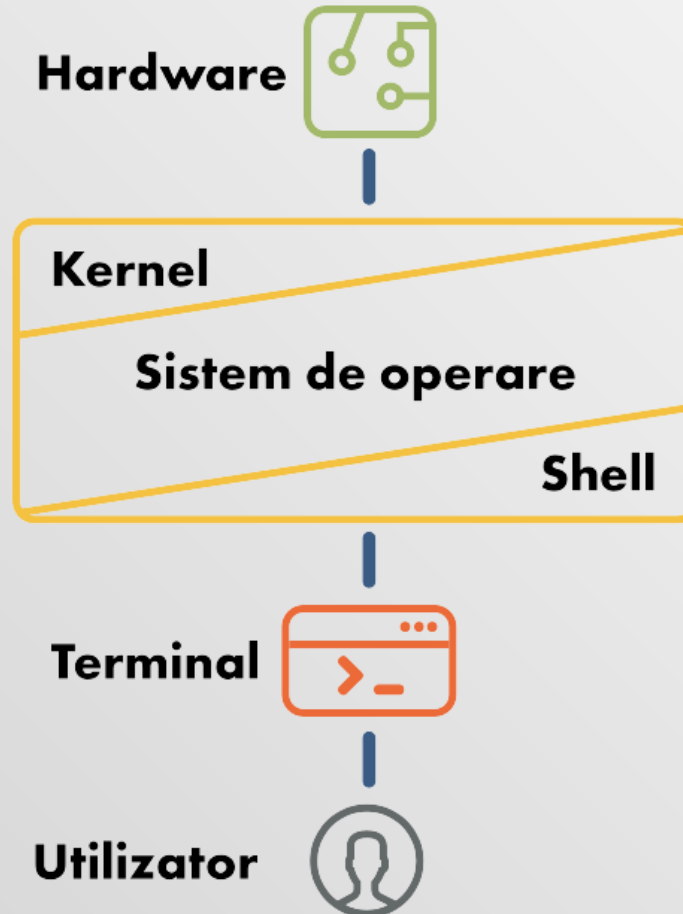
Cum creează un utilizator procese ?

- Deschide aplicații
- Dublu click în GUI
- Rulează o comandă în linia de comandă
- Alt + F2 (Linux) sau Buton Windows + R (Windows)
- La autentificare se pornesc procesele de startup
 - Interfața grafică pentru GUI
 - Shell-ul pentru CLI

Tipuri de procese

- **Interactive**
 - Declanșate de un utilizator prin I/O
- **Neinteractive sau daemon**

Interacțiune terminal



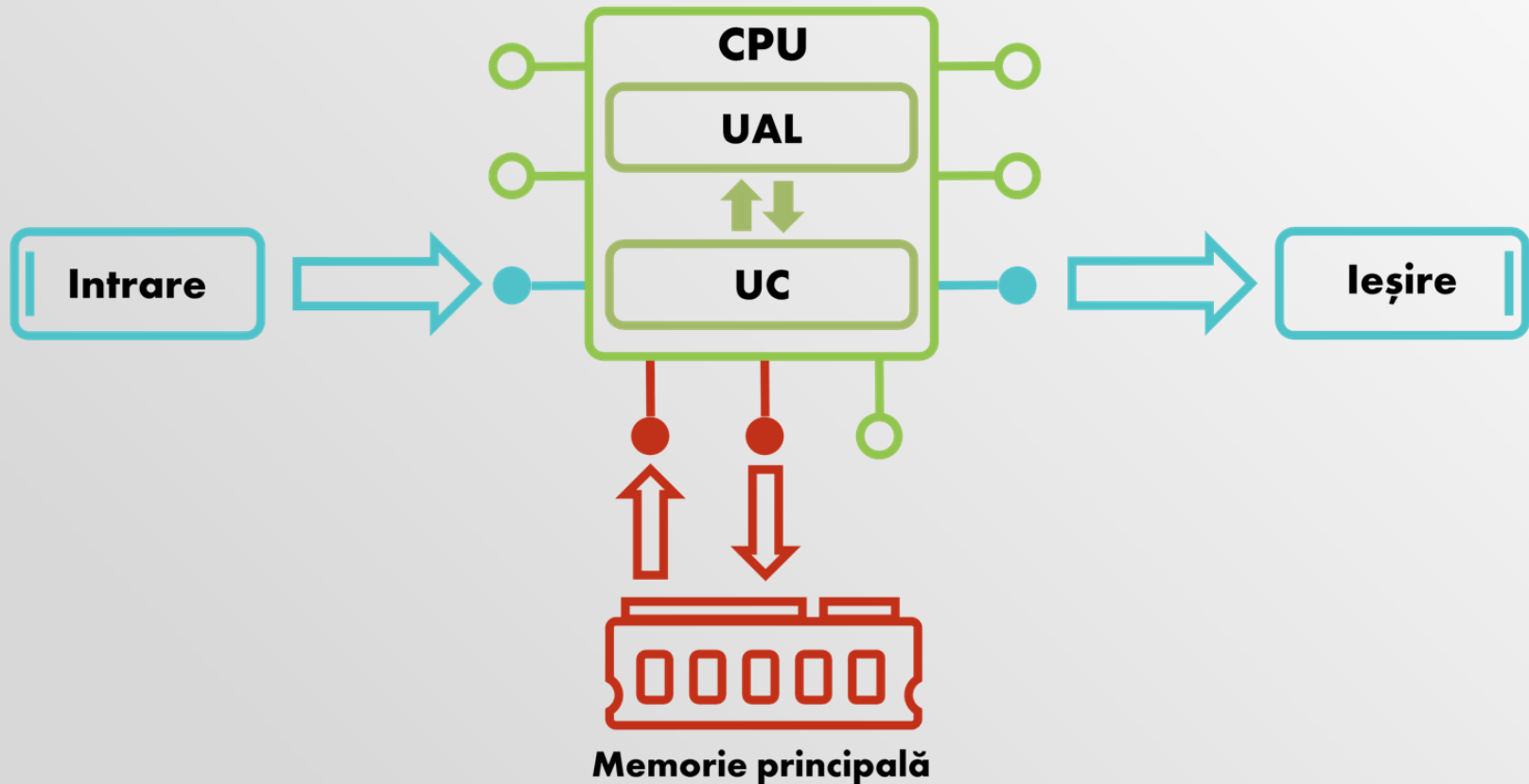
Aplicații vs. procese

- Unele aplicații pot fi compuse din mai multe procese:
 - Navigatoare web (Chrome, Firefox)
 - Server web (servirea cererilor din rețea)
 - Terminal (proces shell)

Resursele proceselor

- Un **identificator** la nivelul sistemului
 - PID = process identifier
- **Memoria:** stochează instrucțiunile executabilului (programului) și datele prelucrate
- **Procesorul:** execută instrucțiunile
- **Spațiu pe disc,** dacă lucrează cu fișiere
- **Dispozitive de I/O:** placă de rețea, USB devices

Folosirea resurselor



Resursele proceselor

- Procesele folosesc resursele SO
- **Aplicații multiproces**
 - Se realizează o partajare a proceselor
- **Integritatea sistemului**
 - Un proces nu poate corupe alt proces
- Procese care accesează **o resursă comună**
 - Accesarea ei ordonată

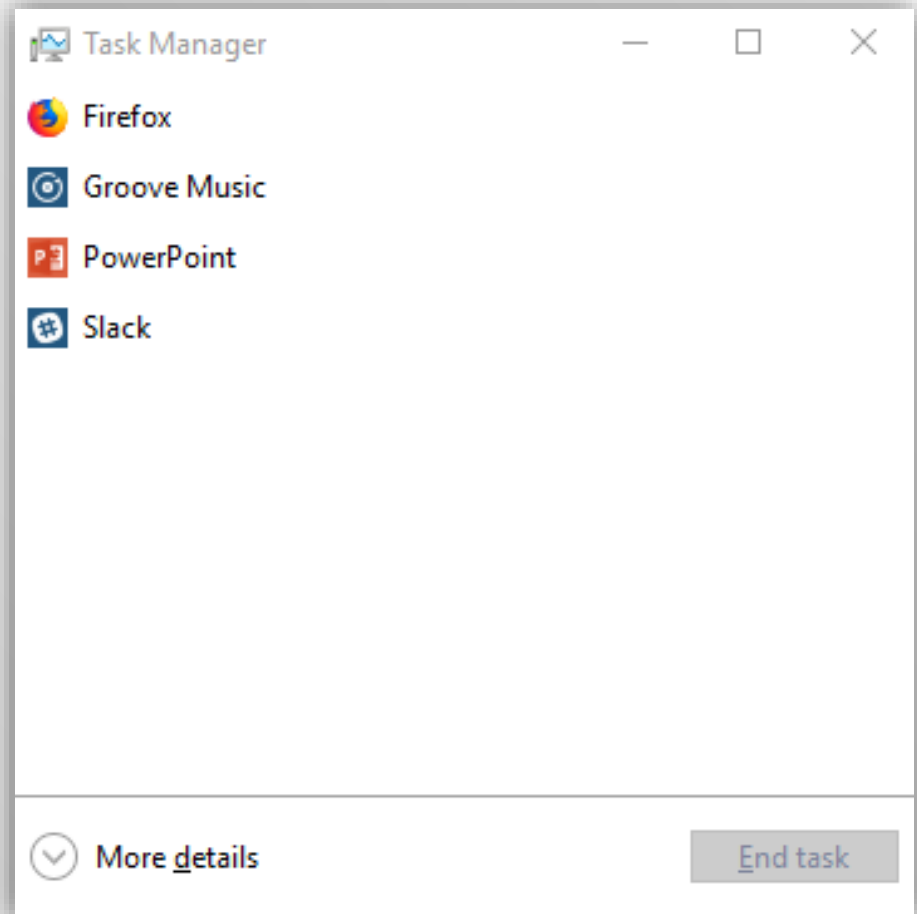
Resursele proceselor

- SO este responsabil de **alocarea echilibrată a resurselor**
 - Un proces să nu consume prea multe resurse de memorie, procesor, etc.



Monitorizarea proceselor

- ps, top, htop
- Task Manager (Windows)



Task Manager

File Options View

Processes Performance App history Startup Users Details Services

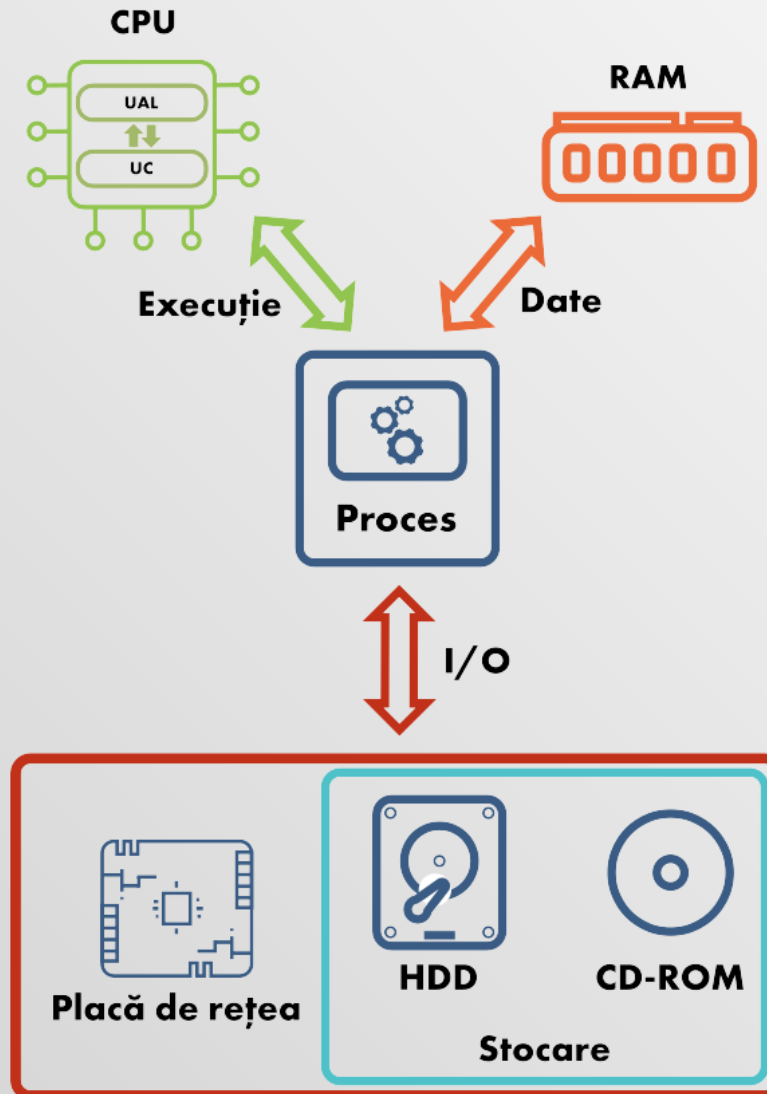
Name	Status	73% CPU	78% Memory	13% Disk	0% Network	3% GPU	GPU engine
Apps (5)							
> Firefox (4)		0.1%	770.9 MB	0 MB/s	0 Mbps	0%	
> Groove Music (2)		1.0%	177.8 MB	8.7 MB/s	0 Mbps	0%	
> PowerPoint		0.6%	117.7 MB	1.0 MB/s	0 Mbps	0.5%	GPU 0 - 3D
> Slack (11)		0.1%	625.8 MB	0 MB/s	0 Mbps	0%	GPU 0 - 3D
> Task Manager		0.1%	34.1 MB	0 MB/s	0 Mbps	0%	
Background processes (118)							
> Adobe Genuine Software Integri...		0%	0.1 MB	0 MB/s	0 Mbps	0%	
> Adobe Genuine Software Servic...		0%	0.2 MB	0 MB/s	0 Mbps	0%	
Adobe IPC Broker (32 bit)		0%	0.8 MB	0 MB/s	0 Mbps	0%	
Adobe Update Service (32 bit)		0%	0.2 MB	0 MB/s	0 Mbps	0%	
Alps Pointing-device Driver		0%	1.2 MB	0 MB/s	0 Mbps	0%	
Alps Pointing-device Driver		0%	0.4 MB	0 MB/s	0 Mbps	0%	

< >

^ Fewer details

End task

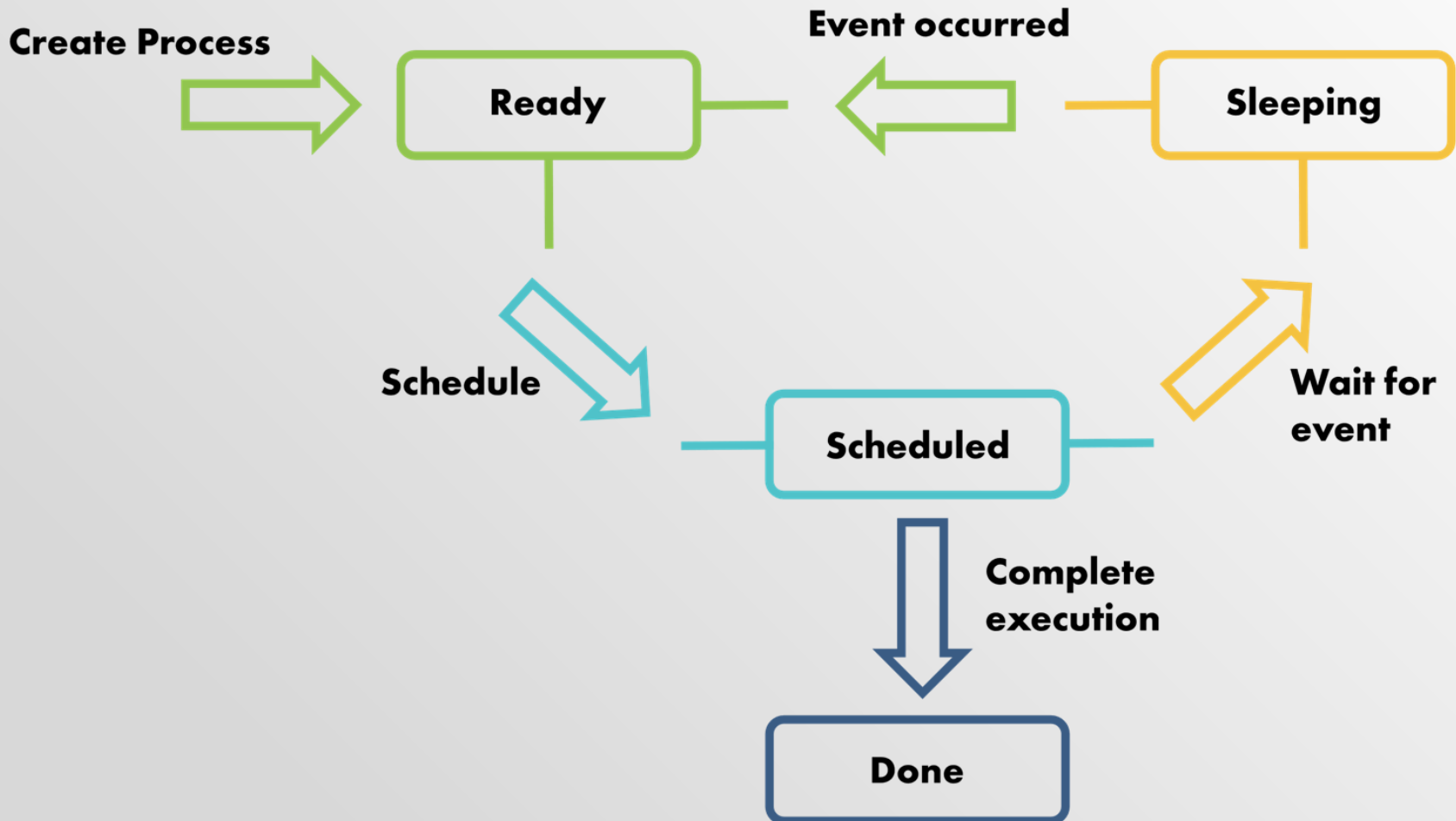
Procese peste CPU



Instrucțiuni și attribute

- **UID:** identifică utilizatorul care deține procesul
- **PID:** identificatorul unic al procesului
 - Nu se poate schimba
- **Instrucțiuni** / cod:
 - Memorare la crearea unui proces
 - Se execută la rularea unei comenzi
- **Attributele** identifică un proces și decid resursele utilizate

Stările proceselor



Utilitare procese

- Afișarea informațiilor despre procesele active din sistem: **ps**, **pgrep**, **pidof**, **pstree**, **lsof**
- Monitorizarea sistemului: **top**, **htop**
- Investigarea amănunțită a proceselor: **ltrace**, **strace**, **ptrace**
- **uptime** - afișează de cât timp funcționează sistemul
- **pidof** – afișează PID-ul procesului cu numele specificat
- Utilitarele ce lucrează cu procese își iau informațiile din `/proc`, din `procfs`

Utilitarul ps

- Utilitarul principal pentru afișarea informațiilor despre procese
- Starea proceselor:
 - Scheduled
 - Sleeping
- **top**
- Prioritatea proceselor – parametrul **nice**

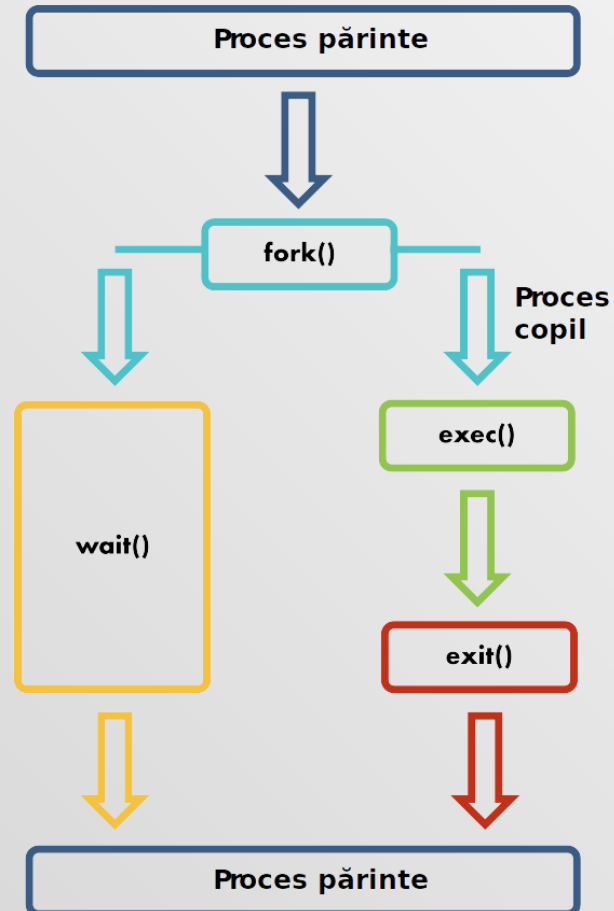
Parametrii comenzii **ps**

- Afişare detaliată: **ps aux**
- **a** = arată procesele tuturor utilizatorilor
- **u** = afişează deţinătorul procesului
- **x** = afişează şi procesele ce nu aparţin terminalului
- Vizualizarea ierarhiei de procese: **ps -H**

Ierarhizarea proceselor

- Crearea unui proces printr-un:
 - Fișier executabil (loading)
 - Proces părinte (unul singur)
- Procesul **init** – rădăcina ierarhiei de procese
- **Shell:** proces care creează alt proces (Windows Explorer, Bash)
- Afișarea ierarhiei de procese: comanda **ps tree**

Ierarhia de procese



Procesul init

- Rădăcina ierarhiei de procese
- Primul proces al sistemului
 - PID 1
- Pornește serviciile sistemului & procesele de bază
 - Shell-uri
- Adoptă procesele orfane



fg & bg

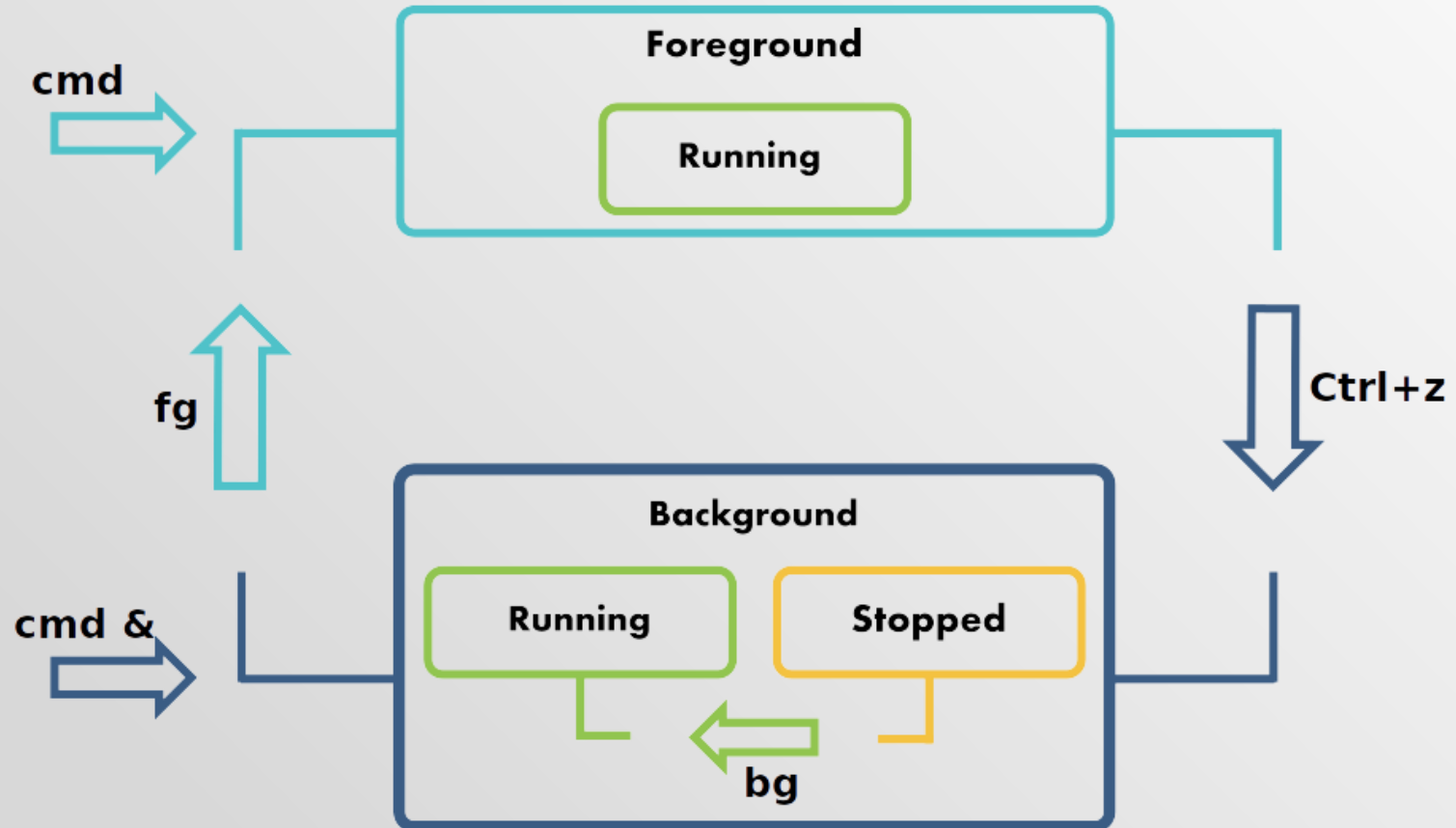
- **Procesul foreground**

- Controlează intrarea standard
- Poate exista doar unul singur

- **Procese background**

- Nu mai are acces la intrarea utilizatorului
- Pot exista mai multe procese de acest fel
- Două stări: running, suspended
- Operatorul **&** trimite procesul în background
- Ctrl + Z
- Comanda job – starea procesului

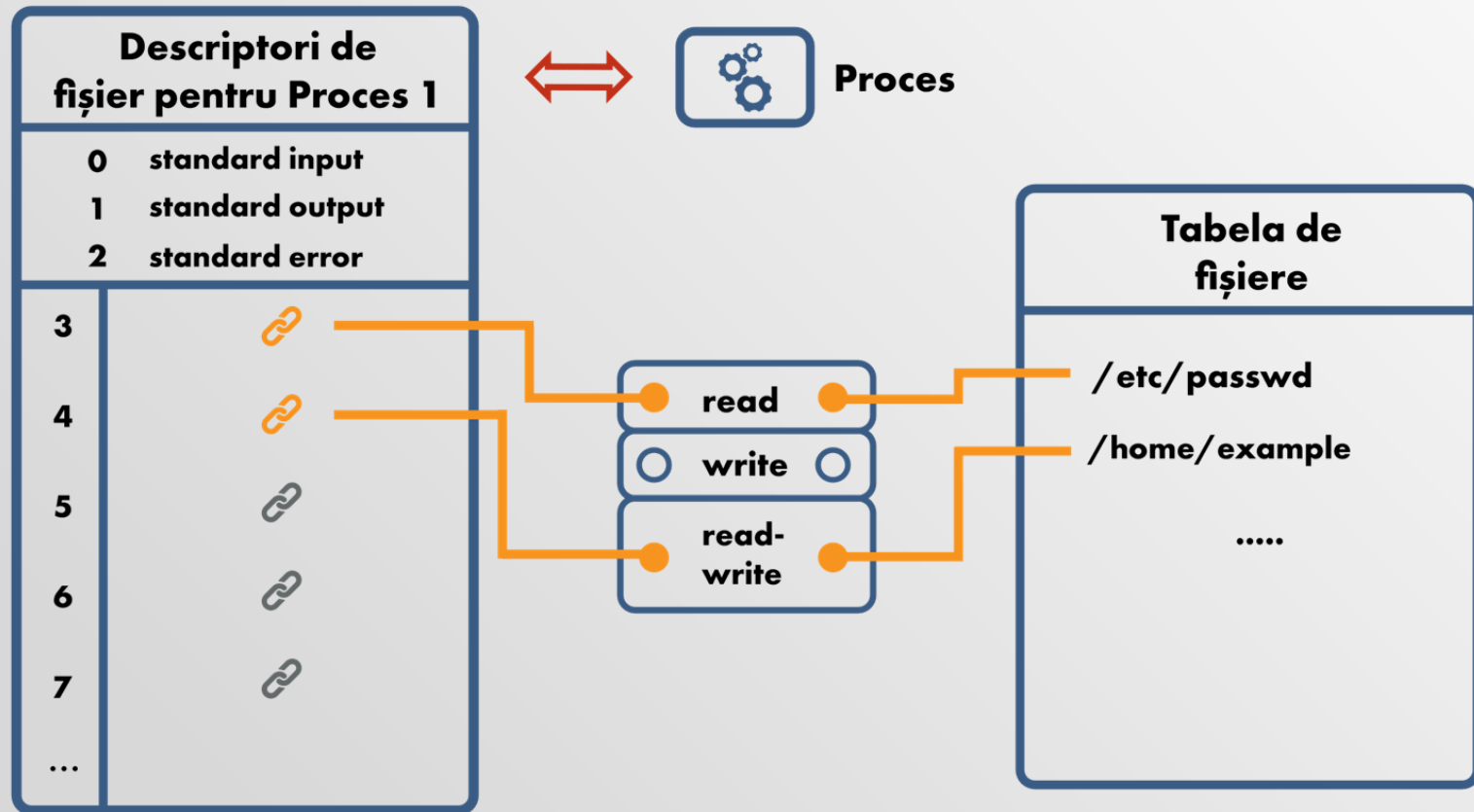
fg & bg



Procese și fișiere

- Accesul către orice fișier se realizează printr-un proces
- Când un proces deschide un fișier primește un **handle** (mâner) către acel fișier
 - O intrare în tabelul descriptorilor de fișiere
 - **Descriptorul de fișiere** identifică fișierul pe care îl rulează un proces
- Un proces poate deschide un număr finit de fișiere (prevenirea atacurilor tip Denial of Service)
- **lshw > ls.out** – redirectarea ieșirii standard

Descriptori de fișiere



Terminarea proceselor

- Un proces este terminat (distrus) atunci când:
 - Ajunge la sfârșitul zonei de execuție
 - Este terminat de utilizator sau distrus de un alt proces
 - Se apelează `exit` sau o funcție similară
 - Efectuează o acțiune nevalidă
 - **Abnormal termination**, generarea unei excepții
- Utilitare: **`kill`**, **`pkill`**, **`killall`**

Ce înseamnă terminare

- La terminarea unui proces:
 - Toate informațiile aferente dispar
 - Resursele (memorie, fișiere) sunt eliberate
- Când un proces moare, se șterge fișierul executabil asociat ?



Semnale

- Interacțiunea între procese
- **SIGSPT** – suspendă un proces
- **SIGTERM** – omoară un proces
- **Ctrl + C** \leftrightarrow **SIGINT** – întrerupe execuția procesului curent
- **Ctrl + ** \leftrightarrow **SIGQUIT** ($>$ **SIGINT**)

Example

- Listă semnale UNIX

```
stud@uso:~$ kill -l  
1) SIGHUP 2) SIGINT 3) SIGQUIT 4) SIGILL  
5) SIGTRAP 6) SIGABRT 7) SIGBUS 8) SIGFPE  
9) SIGKILL 10) SIGUSR1 11) SIGSEGV 12) SIGUSR2  
13) SIGPIPE 14) SIGALRM 15) SIGTERM 17) SIGCHLD  
18) SIGCONT 19) SIGSTOP 20) SIGTSTP 21) SIGTTIN  
22) SIGTTOU 23) SIGURG 24) SIGXCPU 25) SIGXFSZ  
26) SIGVTALRM 27) SIGPROF 28) SIGWINCH 29) SIGIO  
30) SIGPWR 31) SIGSYS
```

Înlănțuirea proceselor

- Redirectare din fișier (< și >)
- Înlănțuirea de comenzi:
 - Înlănțuire necondiționată (;)
 - Sau (||)
 - Și (&&)
- Pipe-uri (|)
 - Sunt folosite la transferul de informație de la o comandă la alta
 - One-linere

Interacțiuni

- Procesele **interactive**:
 - Primesc date de intrare de la utilizator
 - Shell – terminal – interacțiunea cu utilizatorul
- Procese **neinteractive** (batch)
 - Sunt folosite la transferul de informație de la o comandă la alta

Daemoni

- Sunt detașați de terminal
- Oferă un anumit serviciu în sistem
- Au ca părinte procesul **init**
- Exemplu: **atd**
- Daemonizarea unui proces:
 - **nohup**
 - **disown**

De ce „daemoni”?

According to Fernando J. Corbato who worked on Project MAC in 1963 his team is the first to use the term daemon. The use of the term daemon was inspired by **Maxwell's daemon**, in physics and thermodynamics as an imaginary agent which helped to sort molecules.

„We fancifully began to use the word daemon to describe background processes which worked tirelessly to perform system chores.”

<https://stackoverflow.com/questions/33690277/why-are-daemons-called-daemons>

Daemonii ca metaforă

- Daemoni – inaccesibili prin I/O
- Dacă procese = gânduri & emoții, daemoni = procese psihologice sub-conștiente sau inconștiente ?

Example

- **nohup** – redirectează rezultatul procesului daemonizat într-un fișier (nohup.out)

```
stud@uso:~# kill -SIGINT 400
```

- **disown** – oprește procesul și continuă execuția instrucțiunilor: `disown JobID`

Operating Systems Concepts

- editia a 9-a, 2013
- editia a 10-a, 2020
- Silberschatz, Galvin, Gagne
- Una dintre cele mai renumite cărți de programare în Unix
- Acopera foarte bine subiectele de gestiunea proceselor si gestiunea memoriei

Donald Knuth

- The Art of Computer Programming
- analiza algoritmilor
- autorul TEX, METAFONT
- umor profesional





- International Business Machines Corporation
- fondată, ca nume, în 1924
- una dintre cele mai mari companii din lume
- unul dintre cele mai recunoscute brand-uri
- hardware (servere), software, servicii
- Blue Gene supercomputers

DTrace

- dynamic tracing
- analiza dinamica (in timp real) a aplicatiilor si sistemului de operare
- aparut in 2005, Sun Microsystems
- ruleaza pe Solaris, Mac OS X si *BSD
- cand a fost apelata o functie, cu ce argumente
- „scriptat” cu ajutorul unui limbaj de programare
- a primit numeroase premii de inovatie
- modul de kernel portat pe Linux
- echivalent pe Linux este utilitarul SystemTap (stap)

Cuvinte cheie

- Resurse
- Instrucțiuni
- Attribute
- ps
- top
- Shell
- init
- fg
- fg
- Descriptor de fișiere
- kill
- killall
- Semnale
- Interacțiuni
- Daemoni