

# Comunicarea interprocese (IPC)

SO: Curs 4

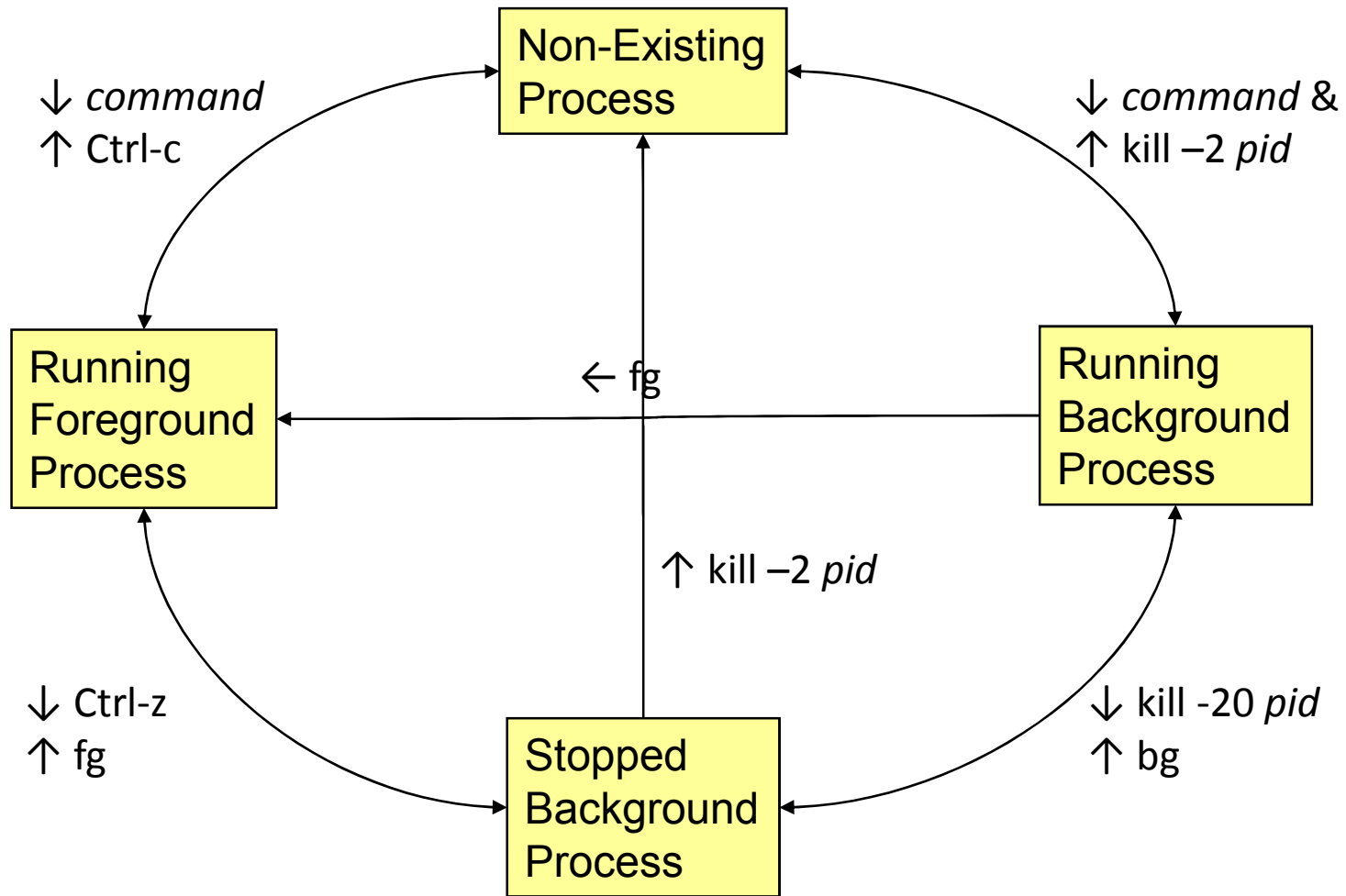
# Recapitulare procese

- Un proces = o unitate de execuție
- Fiecare proces are:
  - Propriu spațiu virtual de memorie
  - Tabelă de descriptori
- Stari proces:
  - READY
  - RUNNING
  - WAITING
- Multi-tasking: SO planifică procesele pe procesoarele existente

# Comunicare interproces

- Semnale
- Pipe
- Memorie partajată
- Sockets

# Controlul proceselor în UNIX



# Implementarea controlului proceselor

## Ce se întâmplă cand

- **Tastăm Ctrl-c?**
  - Tastatura generează întrerupere hardware
  - Întreruperea este tratată de SO
  - SO trimite un semnal 2/SIGINT
- **Tastăm Ctrl-z?**
  - Tastatura generează întrerupere hardware
  - Întreruperea este tratată de SO
  - SO trimite un semnal 20/SIGSTOP
- **Tastăm comanda “kill *-sig pid*”?**
  - SO trimite un semnal *sig* procesului cu id-ul *pid*
- **Tastăm “fg” sau “bg”?**
  - SO transmite semnalul 18/SIGCONT (și alte lucruri!)

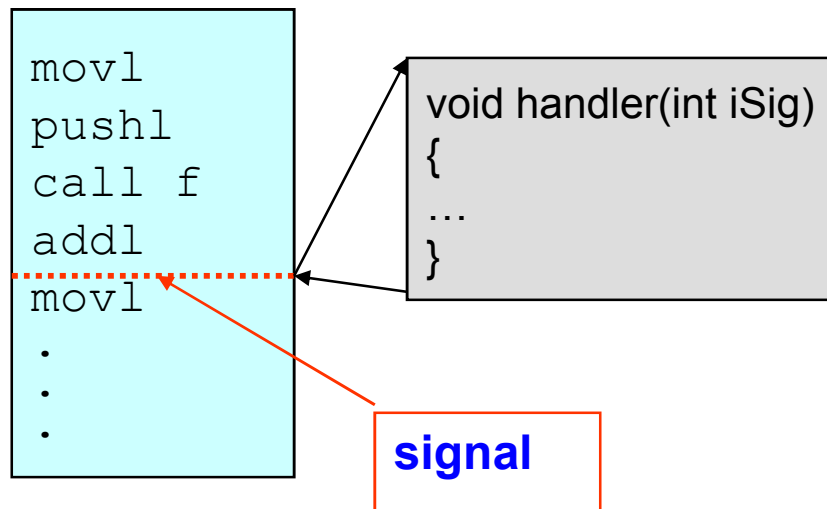
# Semnale: definiție

Semnalele permit SO sa comunice cu procesul

**Semnal:** o notificare transmisă unui proces

- SO descoperă un eveniment (e.g. întrerupere)
- SO oprește procesul (unde îl nimereste)
- **Handlerul de semnal** se execută până la capăt
- Procesul continuă de unde a rămas

Process



# Handler de semnal

- Implicit:
  - De cele mai multe ori “omooară procesul”
- Probleme semnale:
  - Asincrone cu execuția procesului: race conditions
  - Apelurile blocante pot fi intrerupte (read/write): EINTR
  - Nu se pot apela decât funcții reentrante în handler-urile de semnal

# Pipe

- Comunicație unidirecțională
- Pipe: **anonime și cu nume**
- Pipe **anonime**: apelul pipe()
  - Întoarce doi descriptori: unul folosit pentru scriere, altul pentru citire
  - Comunicare între procese înrudite (părinte/copil)
- Pipe **cu nume**: comunicare între orice procese
  - mkfifo / mknod
  - open / ...



# Folosire pipe

- În linia de comanda |
- Anonime: comunicare dupa fork()
- Cu nume: comunicare pe un sistem fizic

# Memorie partajată

- Fiecare proces are propria memorie
- O zonă de memorie poate fi partajată între procese
  - shmget / shmat
  - mmap
- Comunicare fără nici un fel de overhead
  - Fără trecere în nucleu
  - Acces la memorie
  - Necesită sincronizare

# Sockets

- Cea mai răspândită formă de IPC
- Unix sockets: similar cu named pipes
- BSD sockets
  - TCP / UDP
  - Cross-machine
  - Sincronizare explicită (apeluri blocante)
- API: reliable byte stream
  - socket / connect / accept / send / recv