

# Processi, Thread, IPC & Scheduling

---

## Sistemi Operativi (MZ)



[ComplexIntelligentSystems@gmail.com](mailto:ComplexIntelligentSystems@gmail.com)

<http://cis-lab.dmi.unict.it/>

**Mario Pavone**

[mario.pavone@unict.it](mailto:mario.pavone@unict.it)

<https://www.dmi.unict.it/mpavone/>

<https://www.dmi.unict.it/mpavone/so.html>

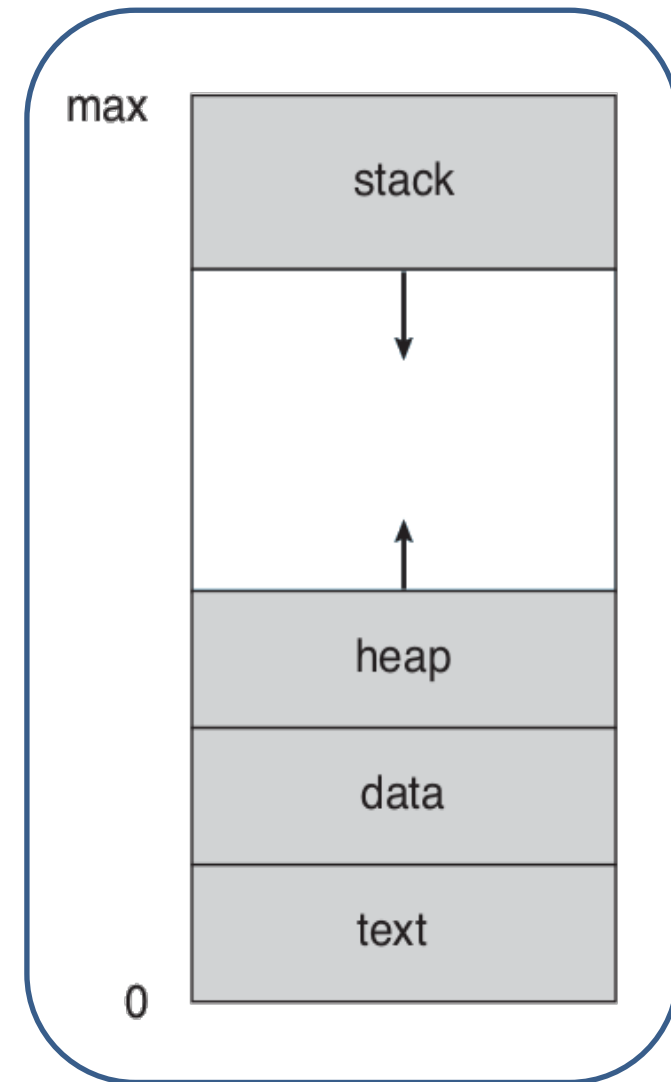
**C.d.L. in Informatica**

**(laurea triennale)**

AA 2025-2026

# Processo

- **Definizione:** una istanza di esecuzione di un programma.
- **Programma vs. Processo**
- Ad ogni processo è associato il suo **spazio degli indirizzi:**
  - codice eseguibile;
  - dati del programma;
  - stack;
  - copia dei **registri** della CPU;
  - **file aperti**;
  - **allarmi** pendenti;
  - processi imparentati.
- Tutte le informazioni relative al processo devono essere salvate (es. file aperti)
- **Tabella dei processi** con un **Process Control Block** (PCB) per ogni processo.



# Processo

- **Definizione:** una istanza di esecuzione di un programma

- **Programmazione**

- Ad ogni processo vengono associati **degli indirizzi**

- codice

- dati

- stack

- copia dei file aperti

- **file aperti**

- **allarmi**

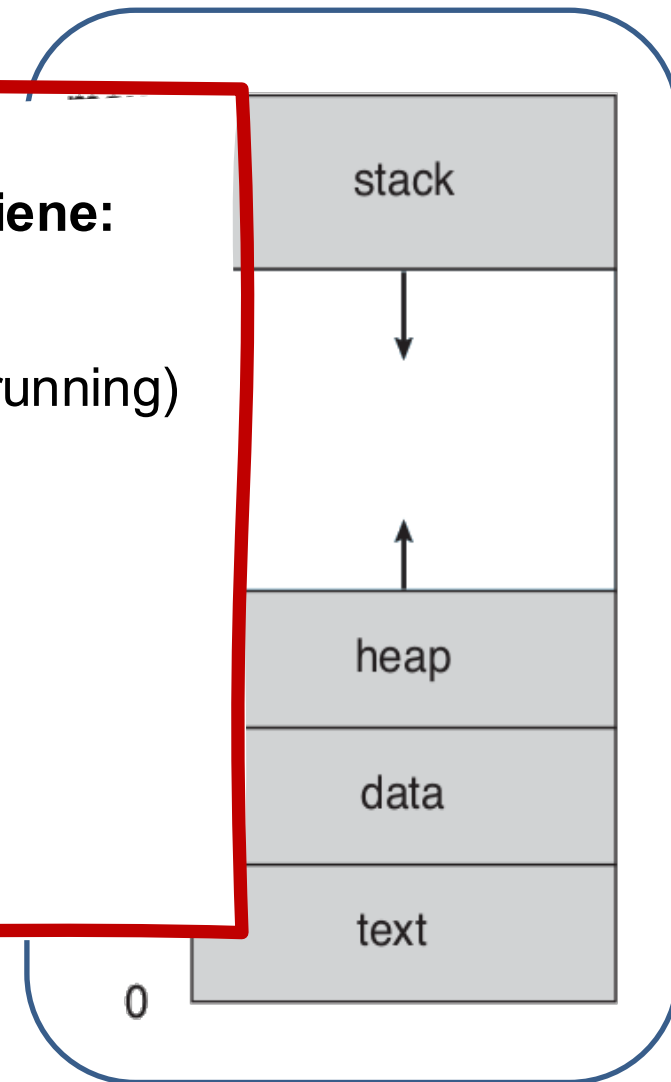
- processi

## Il PCB (Process Control Block) contiene:

- PID (Process ID)
- Stato del Processo (ready, blocked, running)
- Registri della CPU salvati
- Informazioni di scheduling
- Permessi e credenziali
- Puntatore allo spazio degli indirizzi
- Gestione risorse (file aperti, etc.)

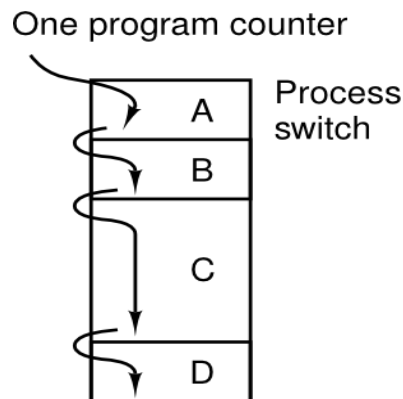
- Tutte le informazioni relative al processo devono essere salvate (es. file aperti)

- **Tabella dei processi** con un **Process Control Block** (PCB) per ogni processo.

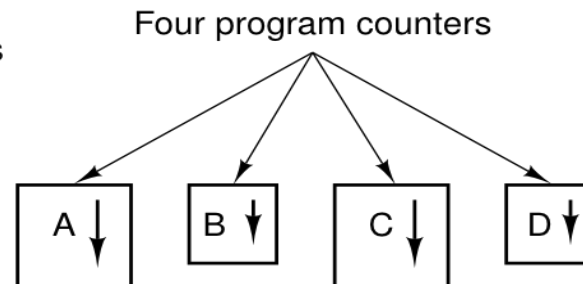


# Modello dei processi

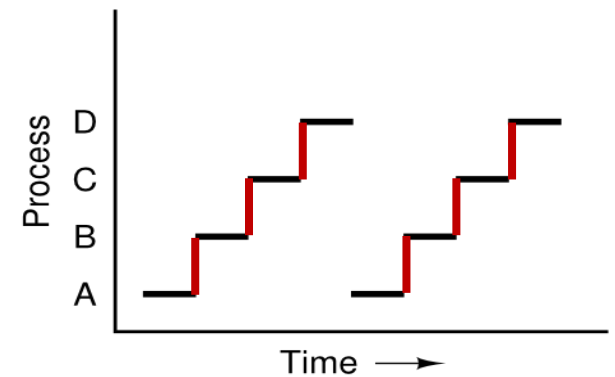
- **Multiprogrammazione e pseudo-parallelismo.**
- È più semplice ragionare pensando a **processi sequenziali** con una **CPU virtuale** dedicata.



(a)



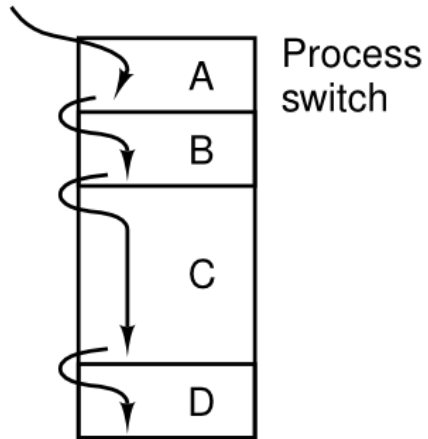
(b)



(c)

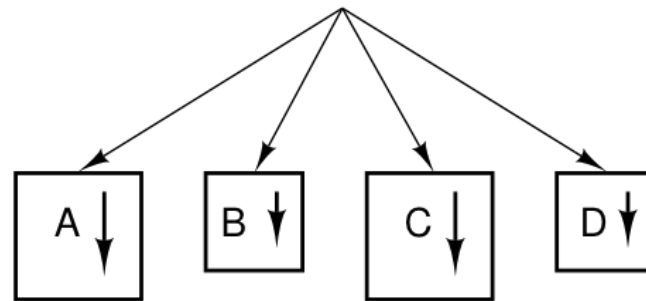
# Modello dei processi

One program counter

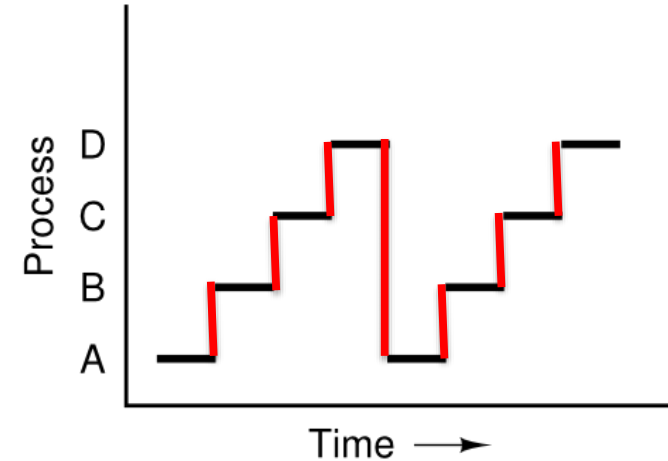


(a)

Four program counters



(b)



(c)

- Esiste un solo PC fisico:
  - **PC logico** viene caricato nel **PC fisico**
  - STOP => PC fisico viene memorizzato nel PC logico

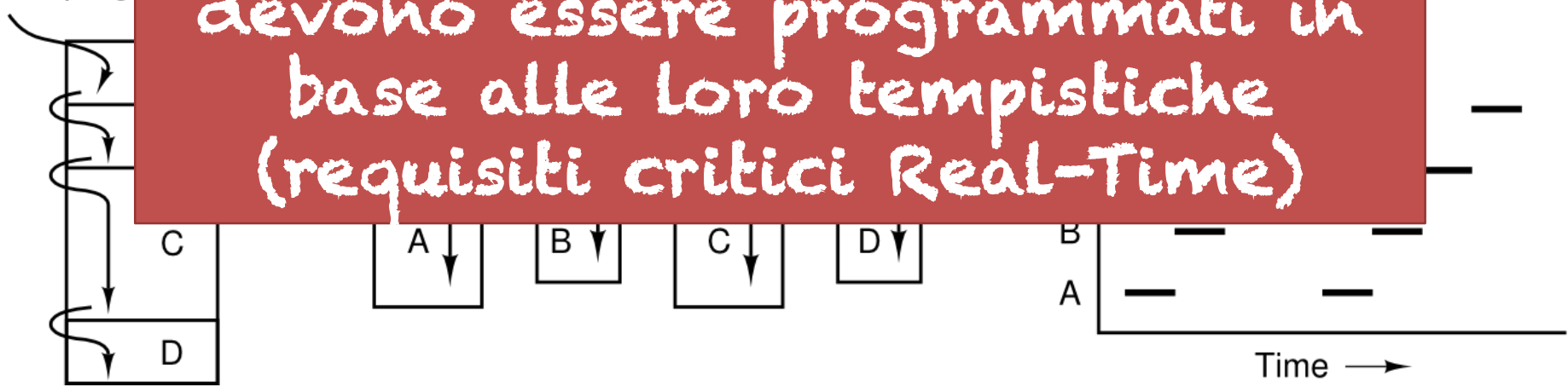
# Modello dei

Multiprogrammazione e pseudo-parallelismo.

pi **NOTA: due processi che eseguono lo stesso programma sono comunque distinti**

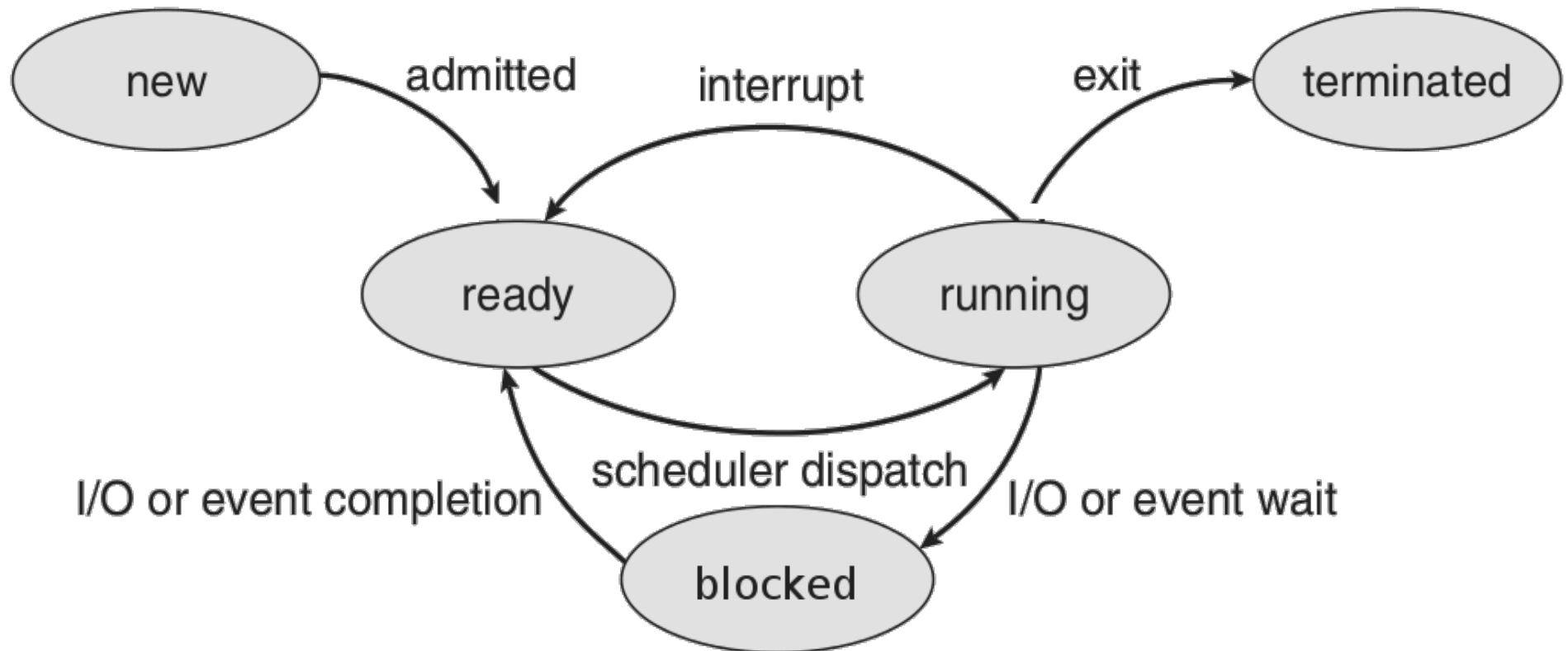
**ATTENZIONE: i processi non devono essere programmati in base alle loro tempistiche (requisiti critici Real-Time)**

One prog



# Stato di un processo

- 3 stati principali (+ 2 aggiuntivi);
- transizioni.



# Creazione dei processi

- **Creazione** di un processo:
  - in fase di inizializzazione del sistema;
    - processi attivi e in background (**daemon**)
  - esecuzione chiamate di sistema di creazione processo
  - richiesta dell'utente
- Creare nuovo processo da un altro processo:
  - **sdoppiamento del padre**: **fork** e **exec** (UNIX);
  - **nuovo processo per nuovo programma**: **CreateProcess** (Win32).
- **NOTA**: dopo la creazione del nuovo processo il padre e il figlio hanno il **proprio spazio degli indirizzi**.
  - una modifica nello spazio di indirizzo di uno non è visibile all'altro



# terminazione dei processi

- **uscita normale:** *exit* (UNIX), *ExitProcess* (Win32);
- **uscita su errore causato dal processo;**
- **errore critico:** alcuni sono gestibili, altri no;
- **terminato da un altro processo:** *kill* (UNIX), *TerminateProcess* (Win32).

# Gerarchia dei Processi



Un processo può crearne un'altro

processo padre e processo figlio

processo figlio può a sua volta crearne altri

**UNIX:** process group

- processo speciale **init**

**In Windows:** NO

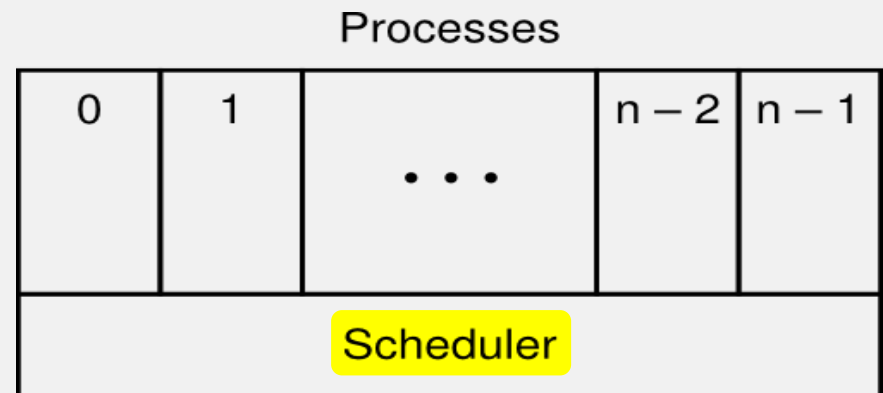
- Tutti i processi sono uguali
- **handle:** token per il processo padre

# Tabella dei processi

|                    |
|--------------------|
| process state      |
| process number     |
| program counter    |
| registers          |
| memory limits      |
| list of open files |
| ...                |

- SO mantiene una **Tabella dei processi**;
- **Process Control Block (PCB)**;
- Contiene tutte le informazioni importanti sullo stato del processo
- **Scheduler**

| Process management        | Memory management             | File management   |
|---------------------------|-------------------------------|-------------------|
| Registers                 | Pointer to text segment info  | Root directory    |
| Program counter           | Pointer to data segment info  | Working directory |
| Program status word       | Pointer to stack segment info | File descriptors  |
| Stack pointer             |                               | User ID           |
| Process state             |                               | Group ID          |
| Priority                  |                               |                   |
| Scheduling parameters     |                               |                   |
| Process ID                |                               |                   |
| Parent process            |                               |                   |
| Process group             |                               |                   |
| Signals                   |                               |                   |
| Time when process started |                               |                   |
| CPU time used             |                               |                   |
| Children's CPU time       |                               |                   |
| Time of next alarm        |                               |                   |



- **NOTA:** i campi della tabella dipendono dal SO