

```
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
```

```
#define MAXPAROLA 30
#define MAXRIGA 80
```

```
int main(int argc, char *argv[])
{
    int freq[MAXPAROLA]; /* vettore di contatori
delle frequenze delle lunghezze delle parole */
    char riga[MAXRIGA];
    int i, inizio, lunghezza;
    FILE *f;
```

```
for(i=0; i<MAXPAROLA; i++)
    freq[i]=0;
```

```
if(argc != 2)
```

```
{
    fprintf(stderr, "ERRORE, serve un parametro con il nome del file\n");
    exit(1);
}
```

```
f = fopen(argv[1], "r");
if(f==NULL)
```

```
{
    fprintf(stderr, "ERRORE, impossibile aprire il file %s\n", argv[1]);
    exit(1);
}
```

```
while( fgets( riga, MAXRIGA, f ) != NULL )
```



# Processi

## Processi: aspetti teorici

Stefano Quer

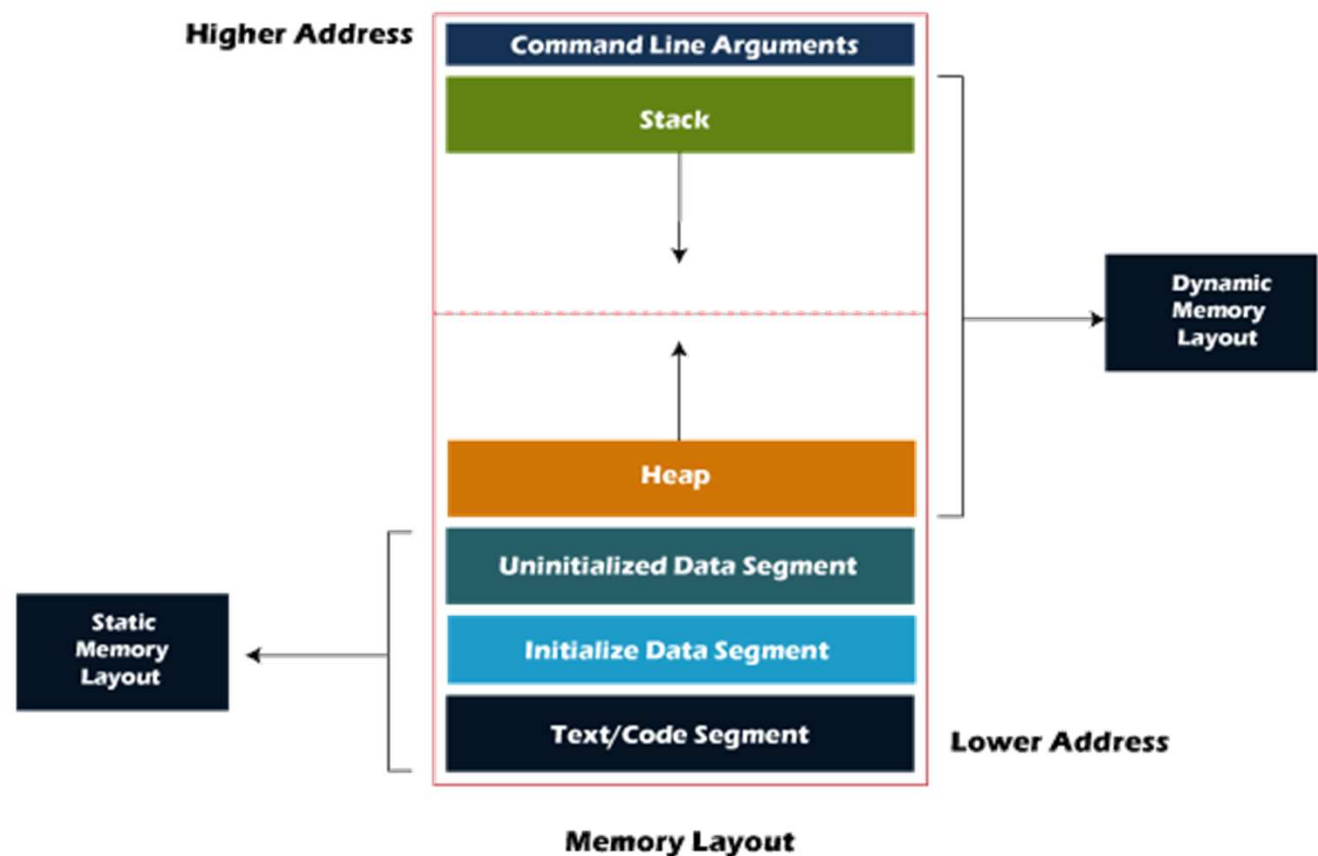
Dipartimento di Automatica e Informatica

Politecnico di Torino

# Programmi e processi

## ❖ Programma

- Entità passiva
- Sequenza di linee di codice



# Programmi e processi

## ❖ Processo

### ➤ Programma in esecuzione

### ➤ Entità attiva

- Codice sorgente e program counter
- Area dati (variabili globali)
- Stack (parametri e variabili locali) e stack pointer
- Heap (variabili dinamiche allocate durante l'esecuzione del processo)

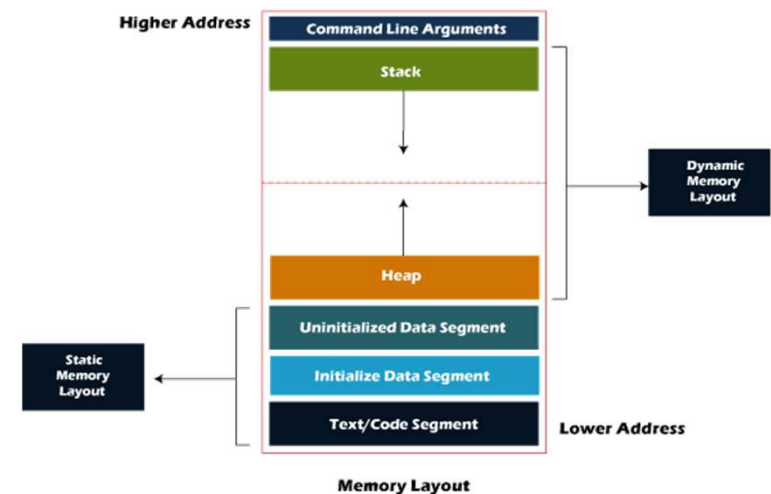


Tabella dei processi

## Process Control Block (PCB)

- ❖ Il SO tiene traccia di ogni processo associando a esso un insieme di dati
- ❖ Tali dati includono
  - **Stato del processo**
    - New, Ready, Running, Waiting, Terminated
  - **Program counter**
    - Indirizzo della successiva istruzione da eseguire

pointer	process state
process number	
program counter	
registers	
memory limits	
list of open files	
⋮	

## Process Control Block (PCB)

- **Registri della CPU**
  - In numero e tipo dipendente dall'hardware
- **Informazioni utili per lo scheduling della CPU**
  - Priorità, puntatori alle code di scheduling, etc.
- **Informazioni utili per la gestione della memoria**
  - Registro base, registro limite, tabelle pagine o segmenti, etc.

pointer	process state
process number	
program counter	
registers	
memory limits	
list of open files	
⋮	

## Process Control Block (PCB)

- **Tabella dei segnali**
  - Signal Handlers
- **Informazioni amministrative varie**
  - Tempo di utilizzo CPU, limiti, etc.
- **Informazioni sullo stato delle operazioni di I/O**
  - Lista dispositivi di I/O, lista dei file aperti, etc.

pointer	process state
process number	
program counter	
registers	
memory limits	
list of open files	
⋮	

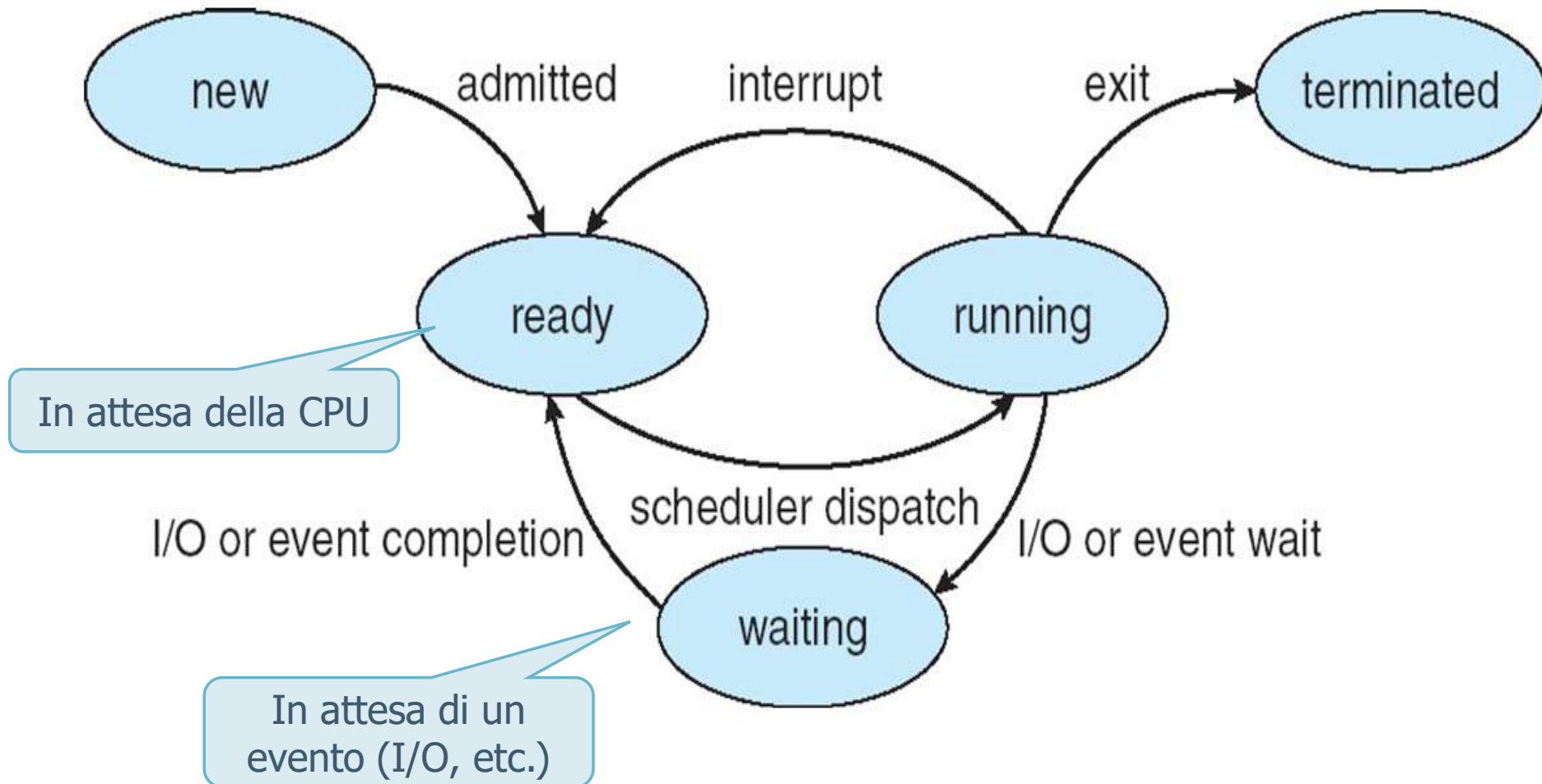
## Stato di un processo

- ❖ Durante la sua esecuzione un processo cambia di stato
  - **New**: il processo viene creato e sottomesso al SO
  - **Running**: in esecuzione
  - **Ready**: Logicamente pronto ad essere eseguito, in attesa della risorsa processore
  - **Waiting**: in attesa della disponibilità di risorse da parte del sistema oppure di un qualche evento
  - **Terminated**: Il processo termina e rilascia le risorse utilizzate



## Diagramma a stati

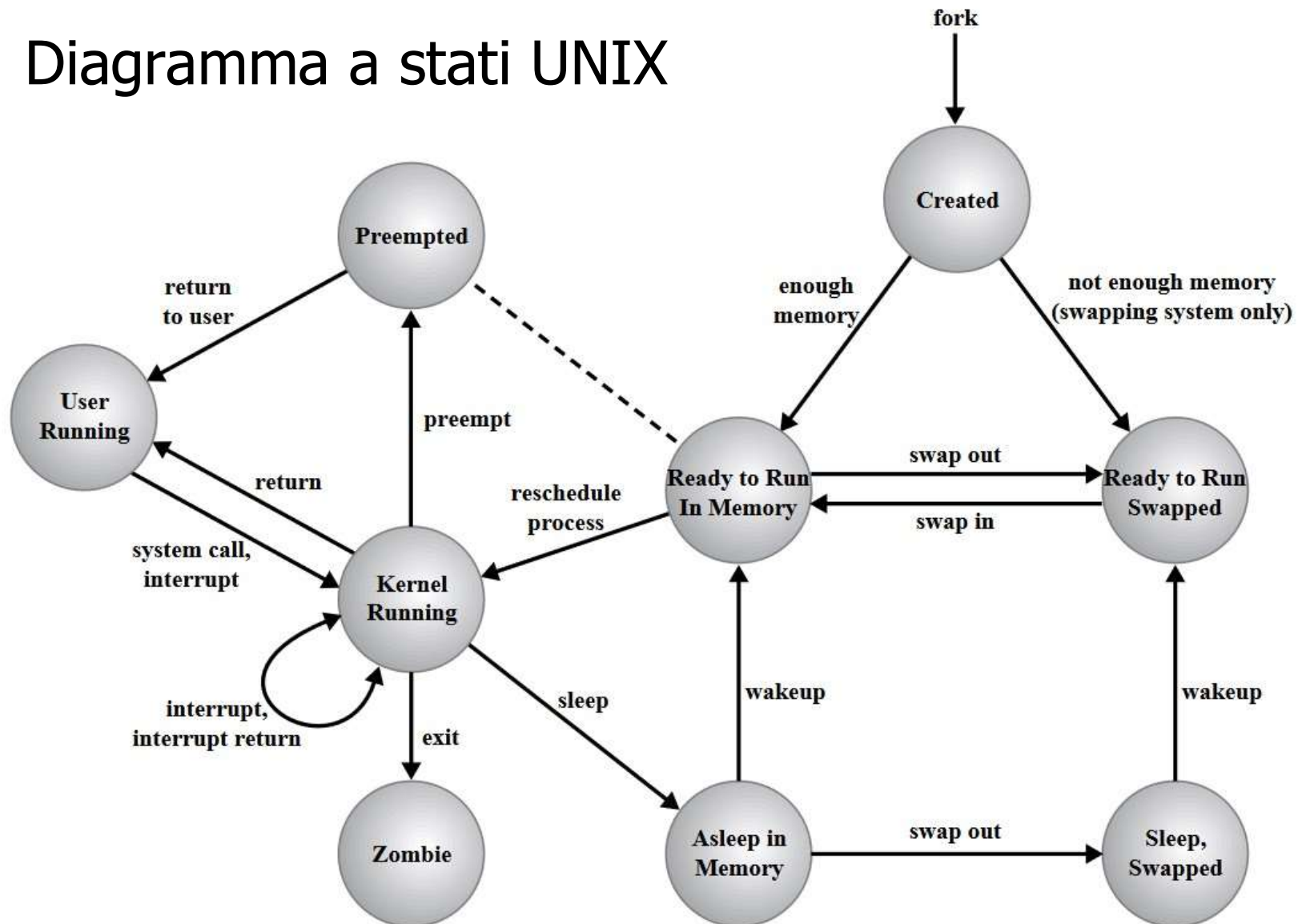
- ❖ L'evoluzione temporale di un processo è descritta da un diagramma a stati





# Diagramma a stati: Sistemi UNIX

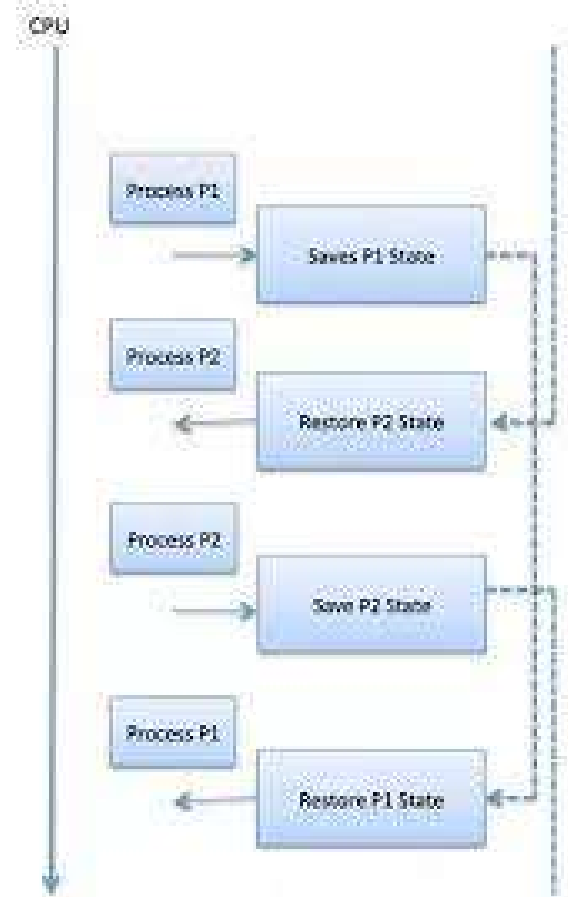
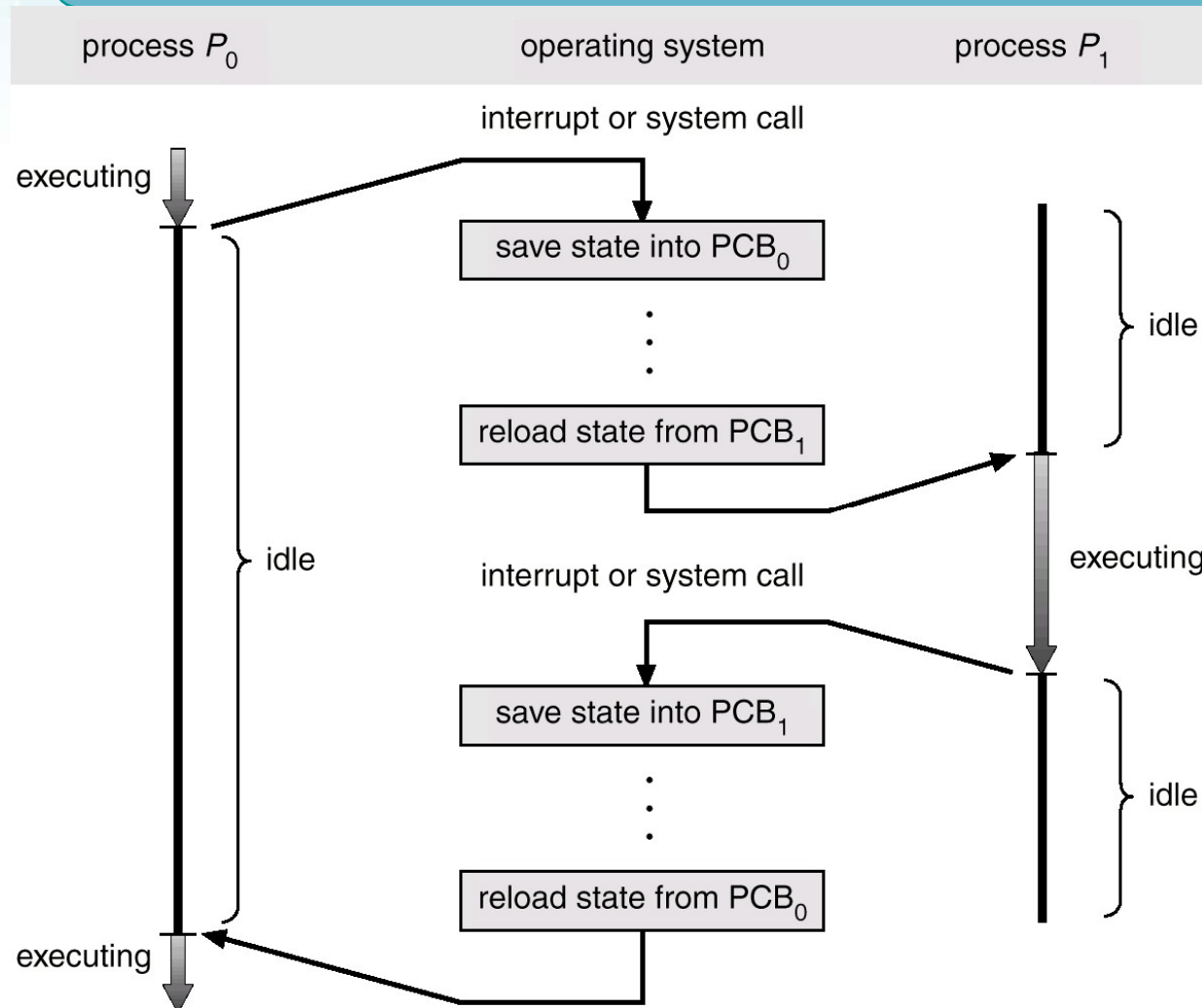
## ❖ Diagramma a stati UNIX



## Context switching

- ❖ Quando la CPU viene assegnata a un altro processo, il kernel deve
  - Salvare lo stato del processo running
  - Caricare un nuovo processo ripristinandone lo stato salvato precedentemente
- ❖ Il tempo dedicato al **context switching** è overhead, cioè lavoro non utile direttamente ad alcun processo
- ❖ Il tempo che un sistema usa per il context switching dipende da svariati fattori
  - Hardware a disposizione, SO, numero di processi, politica di scheduling, etc.

# Context switching



## Scheduler della CPU

- ❖ Le operazioni di **context switching** vengono controllate dalla **scheduler** della CPU
  - L'obiettivo dello scheduler è quello di
    - Massimizzare l'utilizzo della CPU (da parte dei processi)
    - Tentare di soddisfare le varie richieste temporali a livello di sistema (interrupt hardware e software)
  - In ultima analisi, lo scheduler ha il compito di
    - Determinare quando il processo corrente deve terminare la sua esecuzione
    - Selezionare tra i processi disponibili il successivo processo da eseguire

Continua nella  
sezione u09s01