

Sistemi Operativi: Concetti Introduttivi

- 1.1** **Principali funzioni di un Sistema Operativo**
- 1.2** **Cenni Storici**
- 1.3** **Classificazione dei Sistemi Operativi**
- 1.4** **Struttura dei Sistemi Operativi**
- 1.5** **Processi e gestione della memoria**

Sistema Operativo:

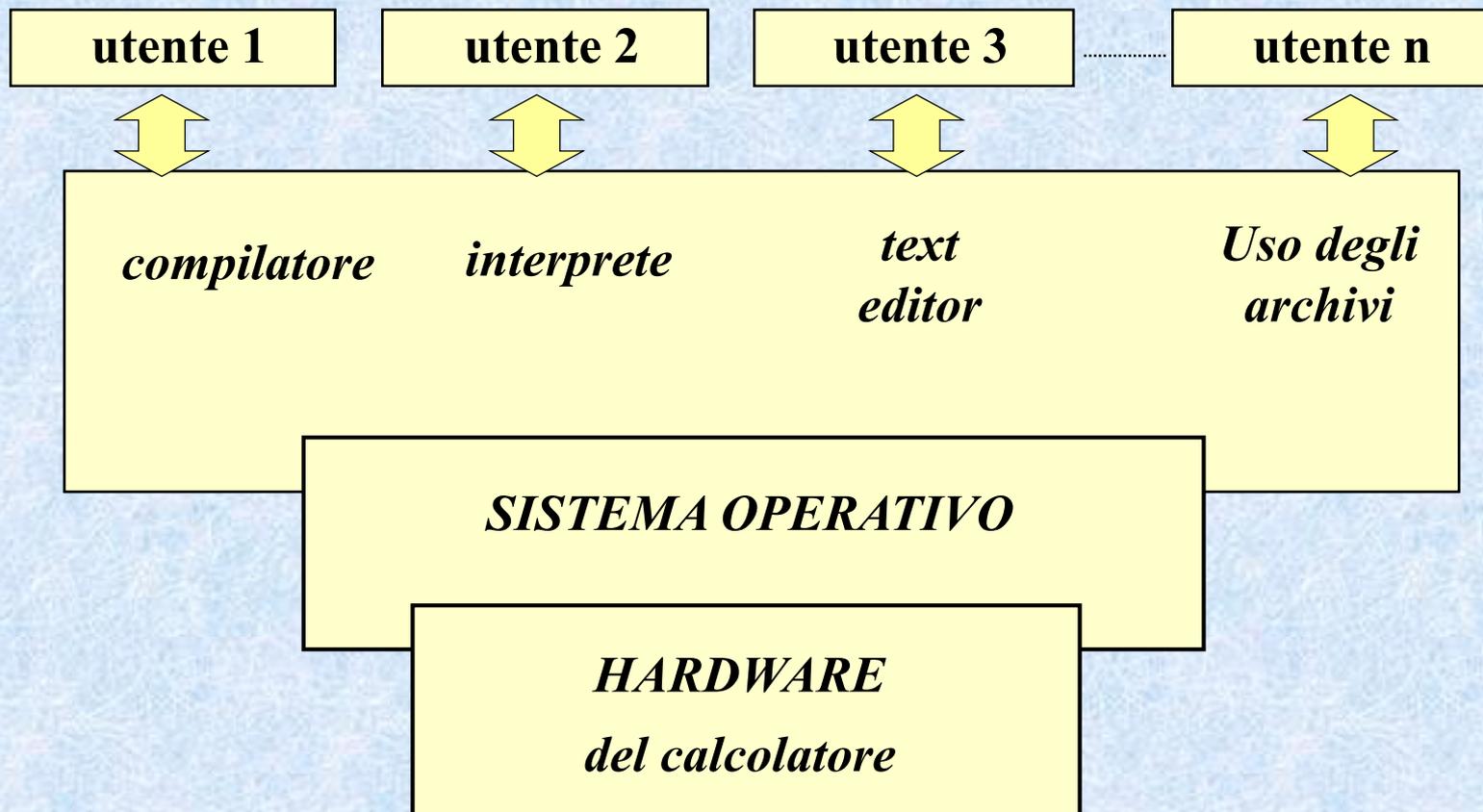
Componente software del sistema di elaborazione

Intermediario tra la macchina fisica (hardware) e i programmi applicativi

Obiettivi:

- **facilitare l'uso del sistema di elaborazione**
- **garantire l'efficienza del suo utilizzo**

Funzioni di un Sistema Operativo



Un Sistema Operativo (S.O.) è un insieme di programmi che operano sull'hardware di un calcolatore per:

- **Facilitare la programmazione**
- **Gestire le risorse (hardware e software)**
- **Proteggere le risorse e l'informazione**

Realizza una *macchina virtuale*

Facilitare la programmazione:

- Utilizzo delle risorse fisiche (ad esempio I/O) tramite *chiamate di sistema*
 - Nasconde i dettagli dei dispositivi fisici
- Realizzazione di risorse logiche (ad esempio archivi) e loro utilizzo tramite *chiamate di sistema*
- Indipendenza del software applicativo dall'hardware (\implies *portabilità*)

Gestire le risorse:

- Ripartire l'uso delle risorse (processore, memoria, dispositivi, archivi) tra più programmi, risolvendo i conflitti
- Realizzare *politiche* per l'assegnazione delle risorse

Protezione, sicurezza e tolleranza dei guasti:

- **Protezione del Sistema Operativo contro l'utilizzo errato o malizioso da parte degli utenti (programmi, dati, dispositivi)**
- **Protezione di un utente nei confronti degli altri utenti (interni, esterni)**
- **Garantire la riservatezza dei dati**
- **Identificazione dei guasti e ripristino del corretto funzionamento**

Astrazione della Macchina Virtuale:

- **Interfaccia del S.O verso gli utenti**
(API: Application Programming Interface)
- **Realizzata con *Chiamate di Sistema (primitive)***

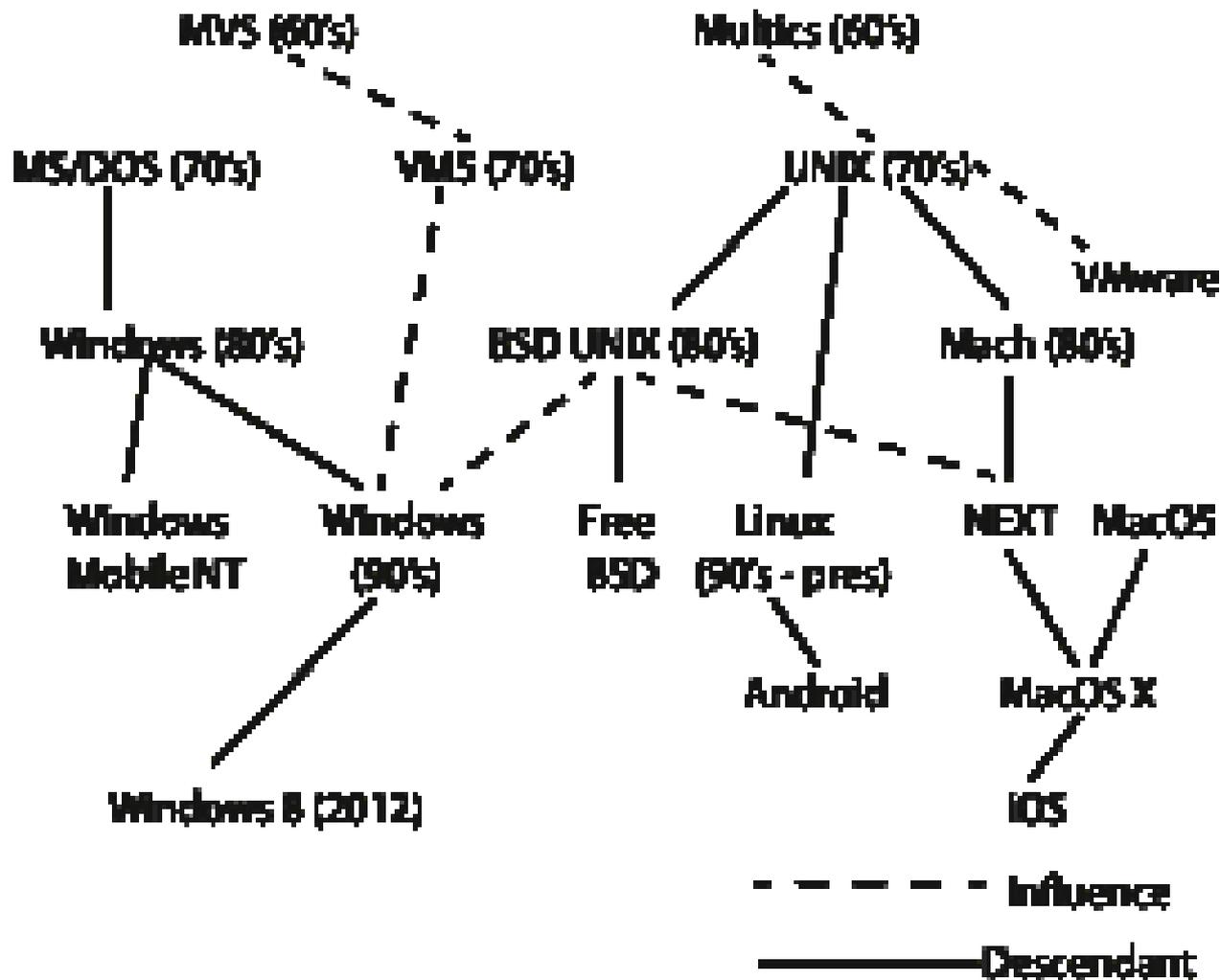
Unica modalità di accesso consentita agli utenti

Cenni Storici

Sistemi Monoprogrammati

- Elaborazione seriale
- Sistemi Batch
- Spooling
- Sistemi Multiprogrammati
- Sistemi Time-sharing

Storia dei sistemi operativi

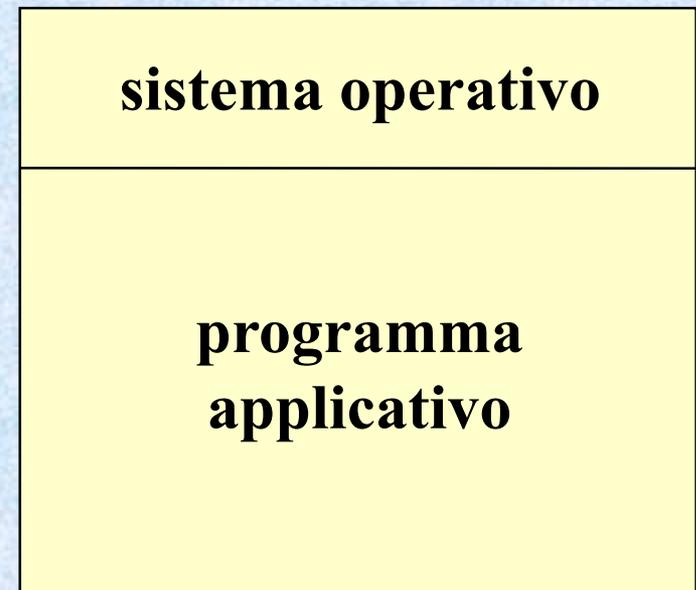


Prestazioni dei sistemi nel tempo

	1981	1996	2011	Factor
MIPS	1	300	10000	10K
MIPS/\$	\$100K	\$30	\$0.50	200K
DRAM	128KB	128MB	10GB	100K
Disk	10MB	4GB	1TB	100K
Home Internet net	9.6 Kbps	256 Kbps	5 Mbps	500
LAN network	3 Mbps (shared)	10 Mbps	1 Gbps	300
Users per machine	100	1	<< 1	100+

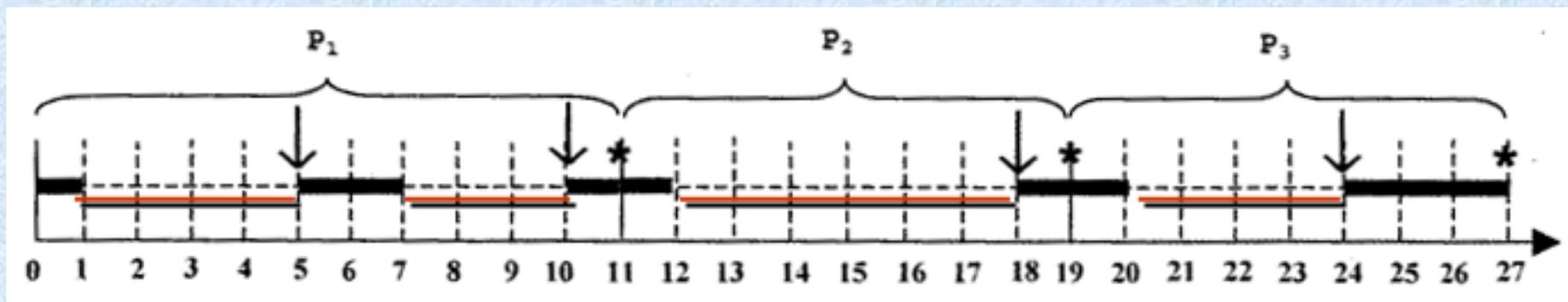
Sistemi monoprogrammati

Oltre al S.O. (*monitor, BIOS*), in memoria centrale risiede (al più) un programma applicativo.



- *Uso inefficiente del processore*

Esecuzione sequenziale in un sistema monoprogrammato



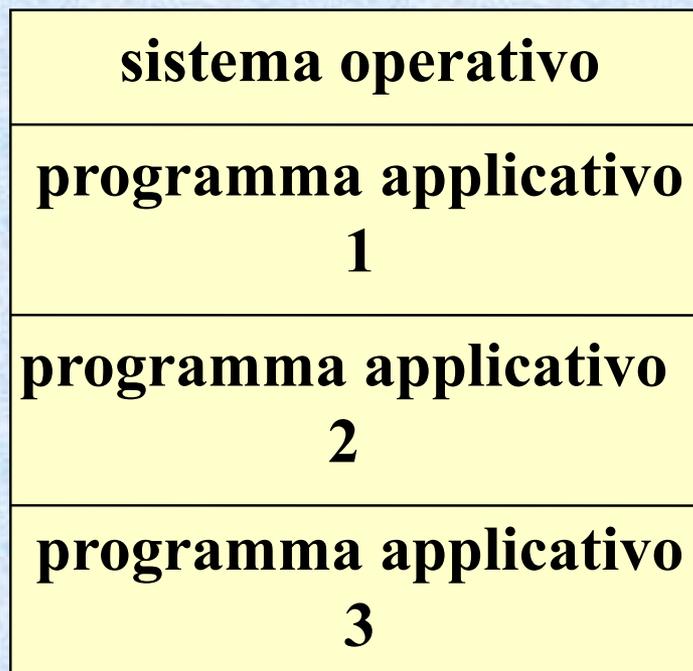
Sistemi *batch* multiprogrammati

Più programmi caricati
contemporaneamente in memoria
(*sistemi multiutente*)

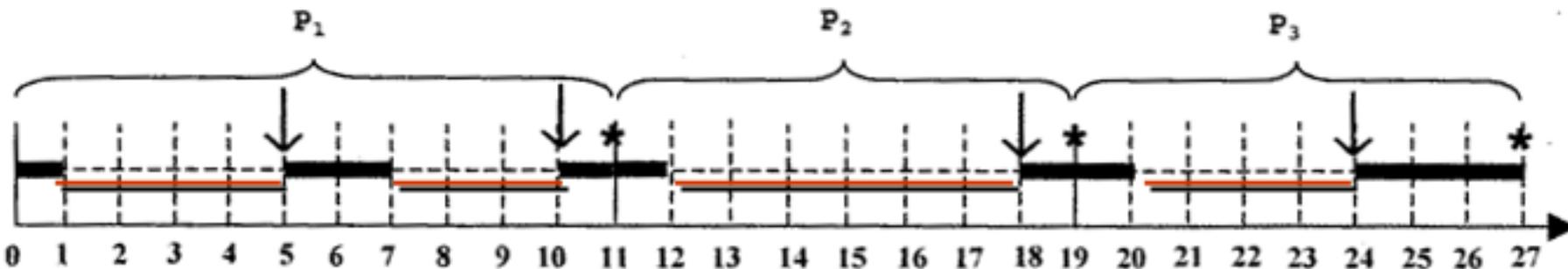
- *ottimizzazione dello spool*

- *ottimizzazione dell'Uso delle
risorse (processore, dispositivi)*

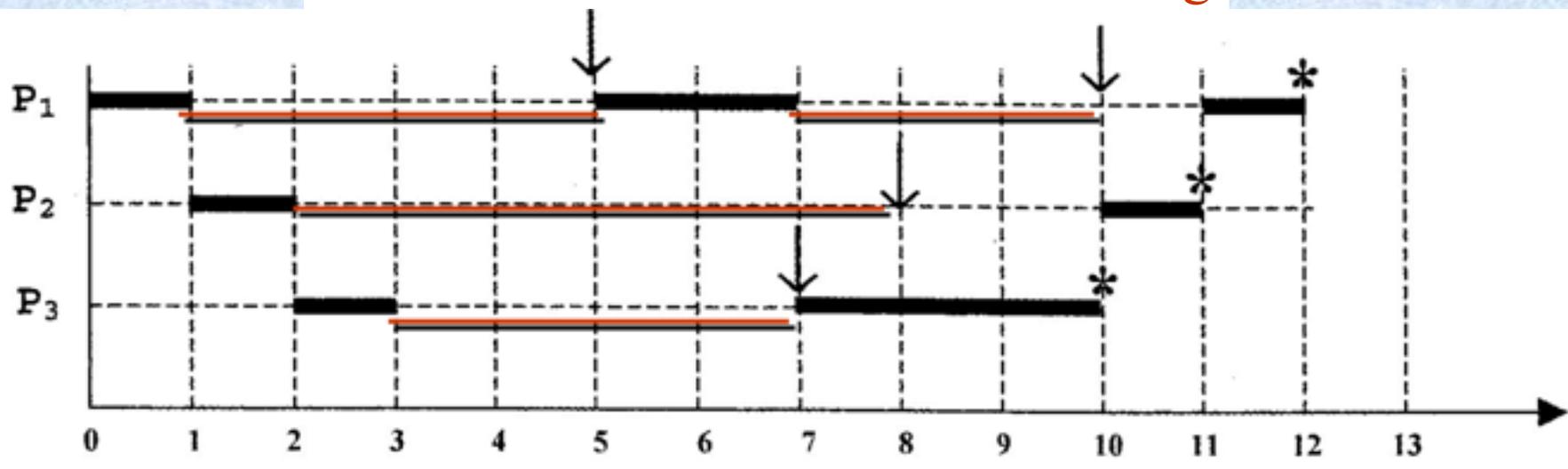
>> *anche a scapito del tempo di risposta*



Esecuzione sequenziale in un sistema monoprogrammato



Esecuzione in un sistema multitasking



Sistemi a partizione di tempo (time-sharing)

Estensione della multiprogrammazione

- *ottimizzazione del tempo di completamento*
- *ingresso/uscita interattiva*
- **Ad ogni programma il S.O. assegna ciclicamente un intervallo (quantum) di tempo della CPU, fino al suo completamento**
- **Al termine dell'intervallo (o durante, se il programma inizia un'operazione di I/O) la CPU viene assegnata ad un altro programma (Round-Robin)**

1.3 Classificazioni dei Sistemi Operativi

Organizzazione interna:

- **monoprogrammati**
- **multiprogrammati**
- **a divisione di tempo**

Visibilità utente:

- **Batch**
- **Interattivi**
- **Transazionali**
- **In tempo reale**
- **PC, PDA**
- **Dedicati**
- **Distribuiti**

Struttura dei Sistemi Operativi

Componenti del S.O.

- **Gestore del processore**
==> concetto di processo, coordinamento dei processi
- **Gestori della memoria principale e secondaria**
- **Gestore dei dispositivi di I/O**
- **Gestore degli archivi (*file system*)**
- **Sistema di protezione e sicurezza**
- **Gestione della comunicazione tra sistemi remoti**
- **Interfaccia con i programmi applicativi**

Organizzazione di un Sistema Operativo

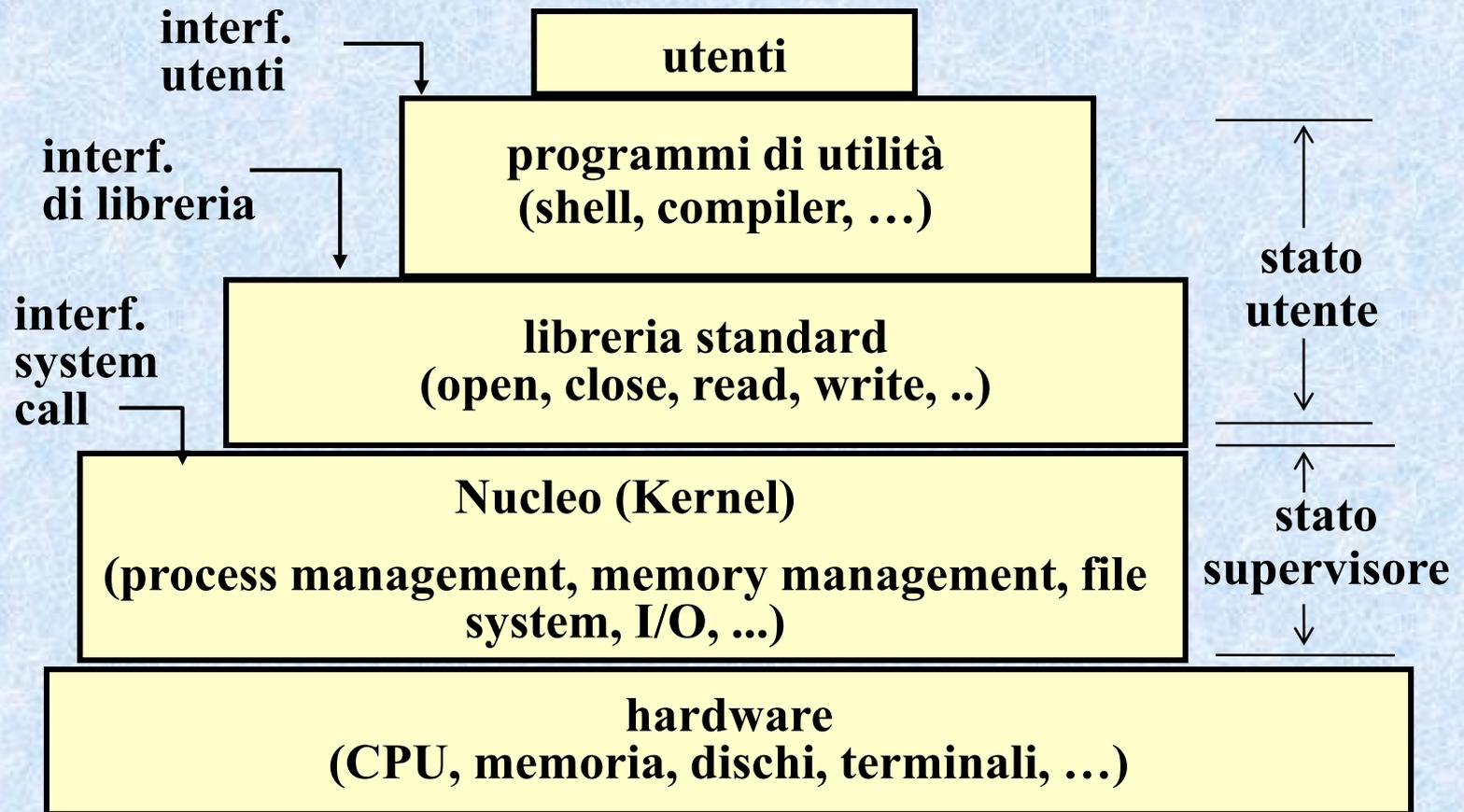
- Come sono organizzate le varie componenti di un S.O.?
- Quali sono le modalità di interazione tra esse?
 - Sistemi monolitici
 - Sistemi modulari*
 - Sistemi a livelli*
 - Sistemi a microkernel
 - Sistemi *client-server*

Sistema Monolitico

Il sistema operativo è costituito da un unico programma contenente un insieme di procedure, che realizzano le varie componenti.

Le applicazioni richiedono i servizi del S.O attivando le corrispondenti componenti con chiamate di sistema, che instaurano lo “stato supervisore” (o “modo kernel”) del processore

Il sistema UNIX



Struttura del sistema UNIX

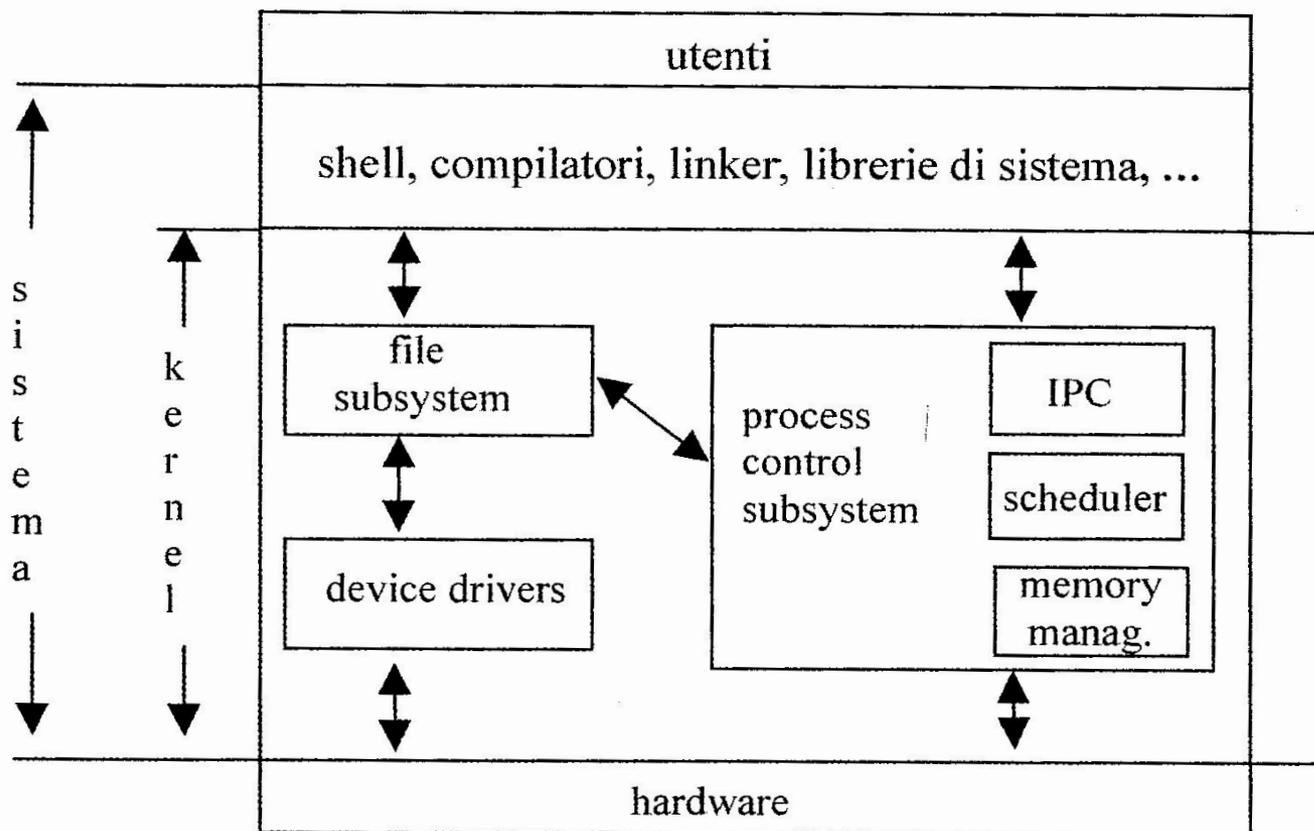


Figura 1.18 Struttura del sistema Unix.

Libreria standard di UNIX

Ogni applicazione C può richiedere l'esecuzione di una system call attraverso una chiamata alla specifica funzione C di libreria che la rappresenta:

Esempio: funzione di libreria per la lettura da file

```
count = read (file, buffer, nbyte);
```

read = nome della funzione di libreria

file = identificatore del file

buffer = area di memoria per i dati

Nbyte = numero di byte da leggere

- *la lettura avviene a partire dal valore corrente del puntatore di lettura*

Processi

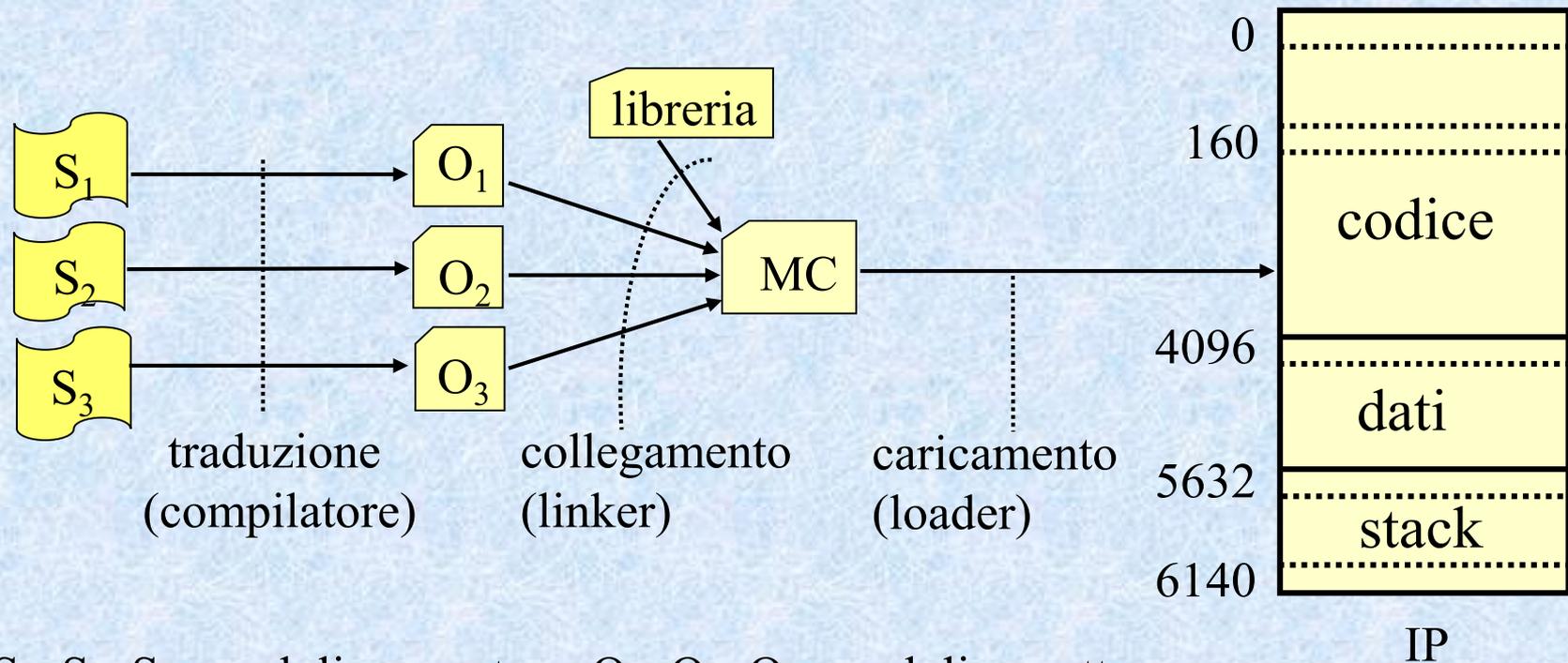
- Informalmente, il termine processo viene utilizzato per indicare un programma in esecuzione.
- *“Processo: attività controllata da un programma che si svolge su un processore”*
- **È l’unità di esecuzione** all’interno del S.O.
 - processi sequenziali
 - un S.O. multiprogrammato consente l’esecuzione concorrente di più processi

Programma e processo

- **Programma:** sequenza di operazioni da eseguire (*flusso*)
- **Processo:** attività di esecuzione, generalmente discontinua

==> A uno stesso programma possono corrispondere più processi, che ciascuno rappresenta l'esecuzione dello stesso codice in tempi diversi e/o con dati diversi

Preparazione di un programma per l'esecuzione

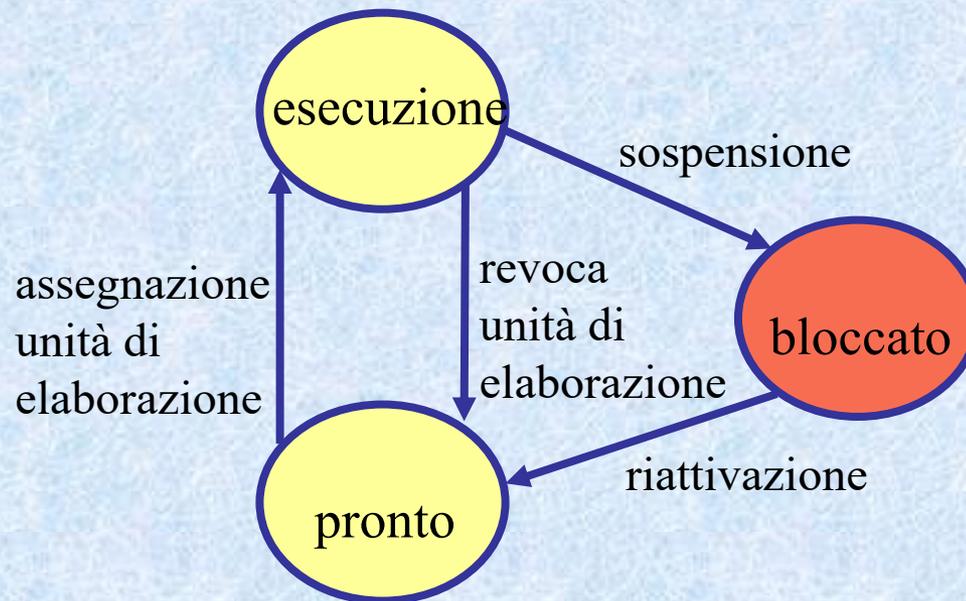


S_1, S_2, S_3 : moduli sorgente; O_1, O_2, O_3 : moduli oggetto;

MC: modulo di caricamento (file eseguibile);

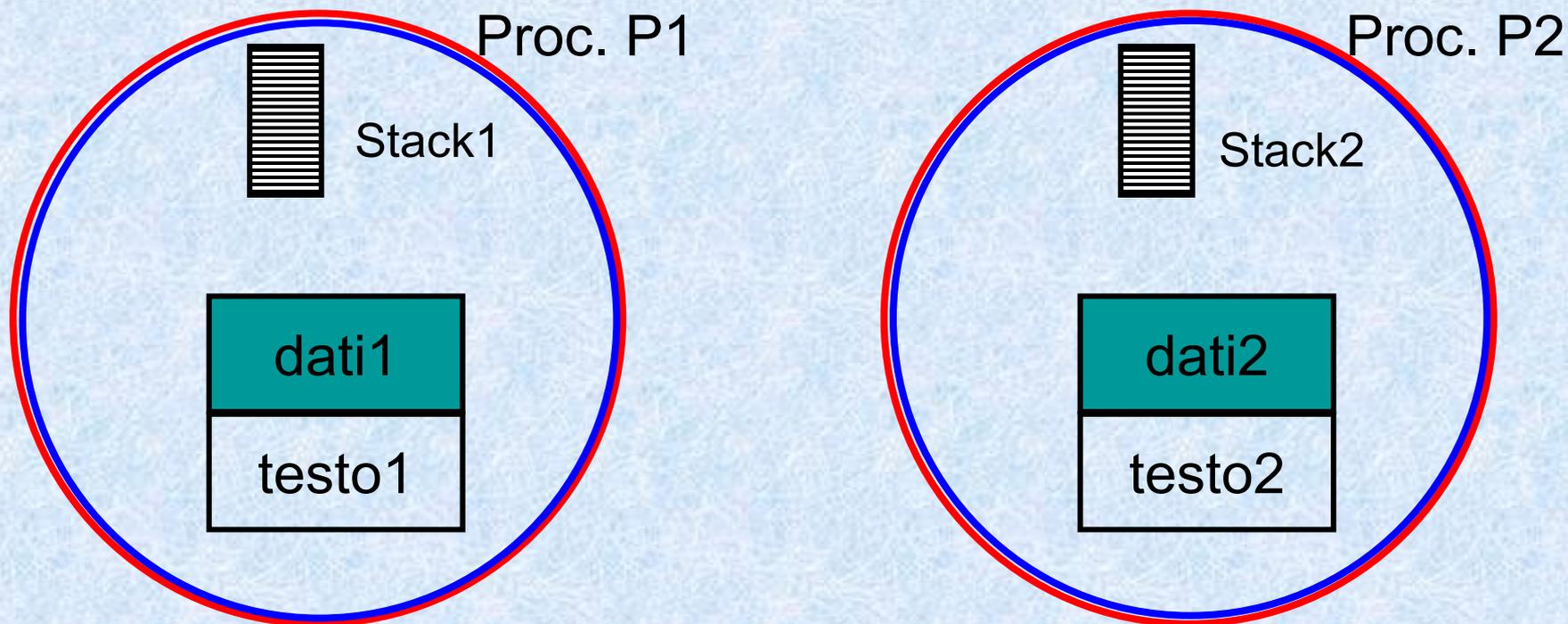
IP: immagine del processo

Stati di un processo



Modello ad ambiente locale

Spazi di indirizzamento (logici) e aree di memoria (fisiche) separati

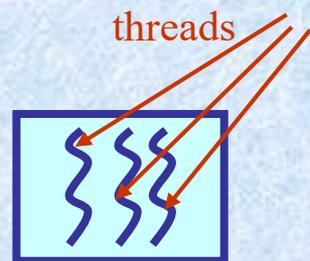


Thread

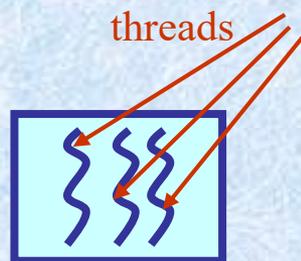
Un thread realizza un flusso di esecuzione all'interno di un processo

- *i thread di uno stesso processo condividono lo (o parte dello) spazio di indirizzamento del processo e le altre risorse*

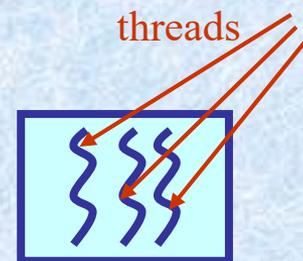
Multithreading: il S.O. realizza flussi di esecuzione multipli all'interno dei singoli processi



Processo P1



Processo P2



Processo P3

Multithreading

Esempio: editore di testi con multithreading

Processo: possiede le risorse

- Memoria
- Archivio (memoria secondaria)
- Dispositivi di I/O (tastiera, video, mouse)

Threads: corrispondono ai flussi di esecuzione

1. Gestione del mouse
2. Immissione da tastiera
3. Gestione della struttura dati del testo
4. Visualizzazione sul terminale
5. Controllo ortografico
6. Salvataggio periodico nell'archivio

Processi e Thread

Processo (*pesante*):

- Risorse di memoria
 - codice, dati
- altre risorse
 - Archivi aperti, ...
- Thread1, Thread2,

Thread (*processo leggero*):

- Condivide (*in tutto o in parte*) codice e dati del processo
- Possiede stato, contesto (PC, registri), stack, descrittore

Thread

Proprietà dei processi:

- Spazi di indirizzamento (generalmente) separati e reciprocamente protetti
- Interazioni che implicano chiamate di sistema e cambi di contesto
- Generazione e terminazione: operazioni complesse che implicano anche l'assegnazione e il rilascio di risorse

Proprietà dei thread:

- Condividono lo spazio di indirizzamento del processo cui appartengono
- Interazioni con modalità semplificate (attraverso dati comuni)
- Generazione e terminazione non implicano assegnazione e rilascio di risorse