

# 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**

# Sistema Operativo:

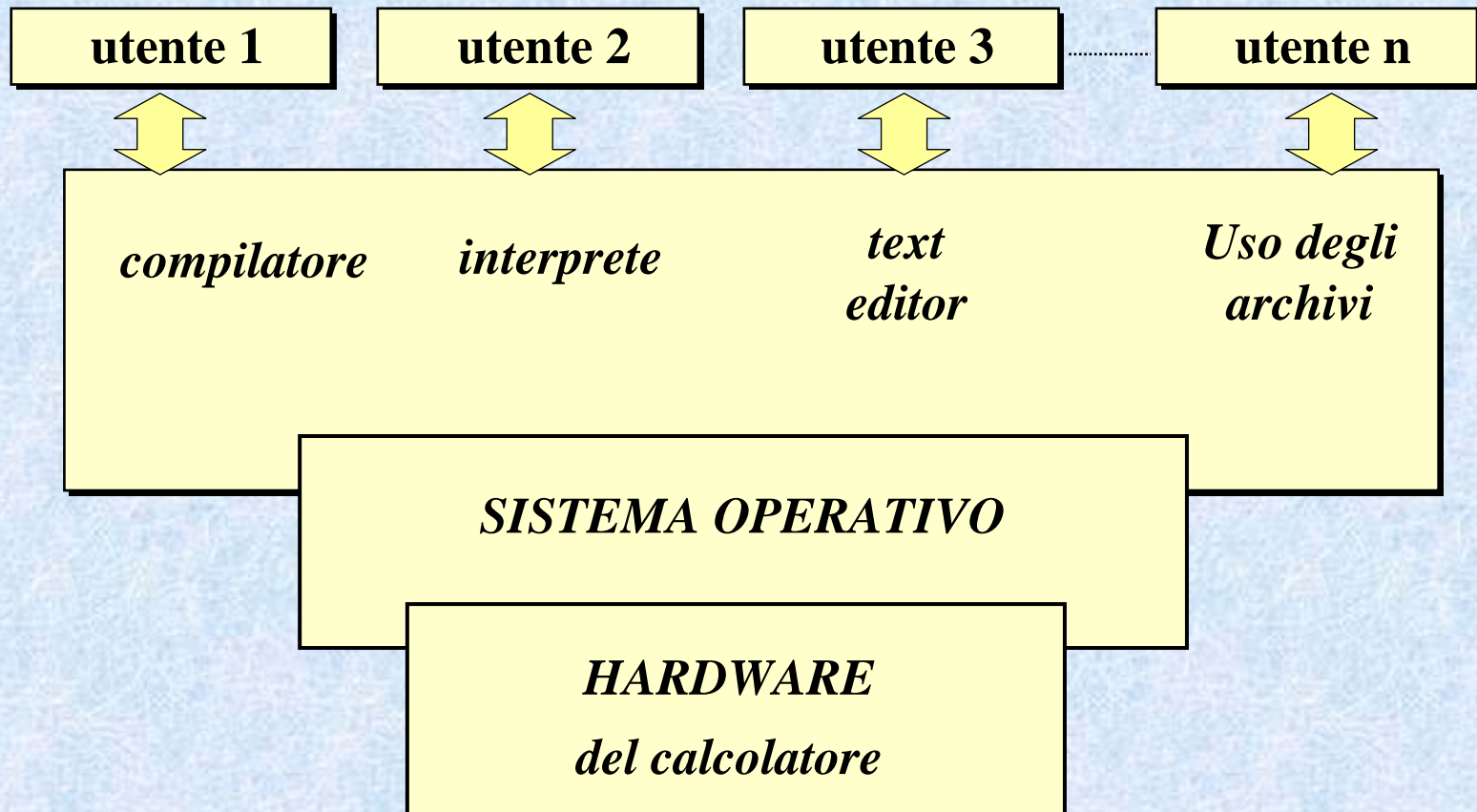
**Componente software del sistema di elaborazione**

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

## Obiettivi:

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

# 1.1 Funzioni di un Sistema Operativo



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

- **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*
- Realizzazione di risorse logiche (ad esempio archivi) e loro utilizzo tramite *chiamate di sistema*
- Indipendenza del software applicativo dall'hardware (==> *portabilità*)



## Gestire le risorse:

- Nascondere i dettagli dei dispositivi fisici (chiamate di sistema)
- Ripartire l'uso delle risorse (processore, memoria, dispositivi, archivi) tra più utenti, 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 le 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)**

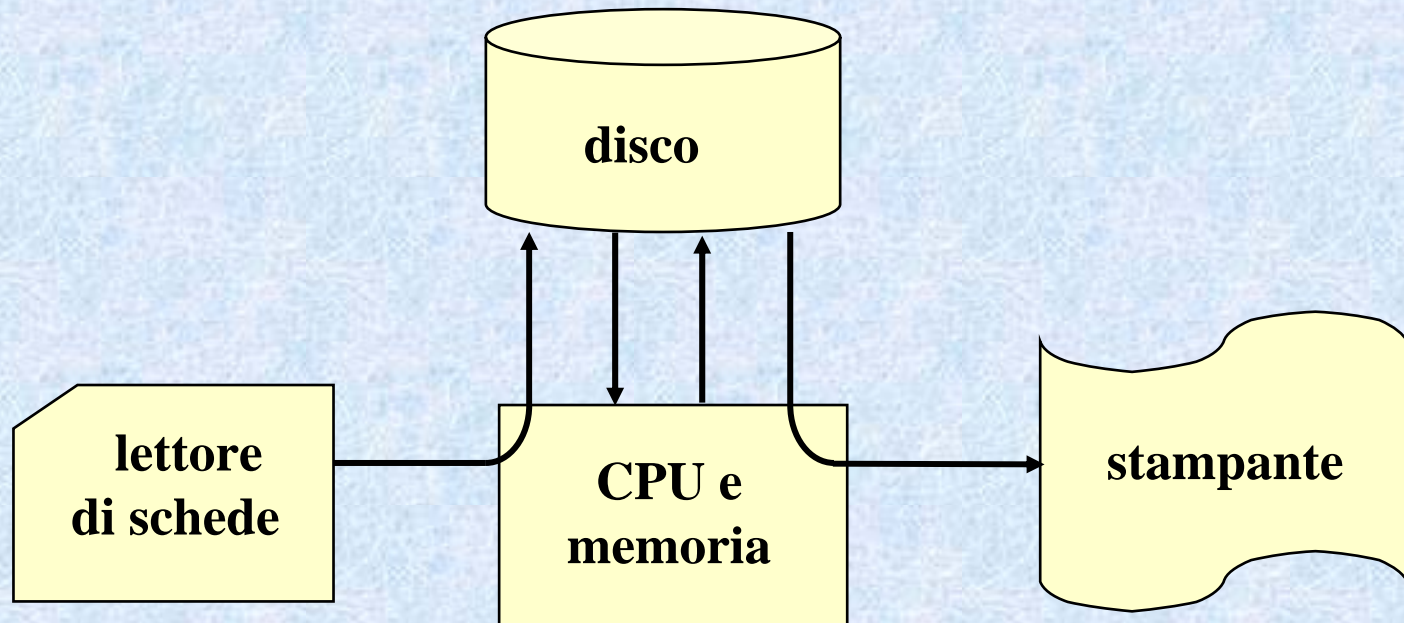
**Unica modalità di accesso consentita agli utenti**



## 1.2 Cenni Storici

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

# Spooling (simultaneous peripheral operation on-line)



***(Primi sistemi batch)***

## Sistemi monoprogrammati

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

**sistema operativo**

**programma  
applicativo**

- *Uso inefficiente del processore*

## Confronto tra esecuzione sequenziale ed esecuzione multi-tasking

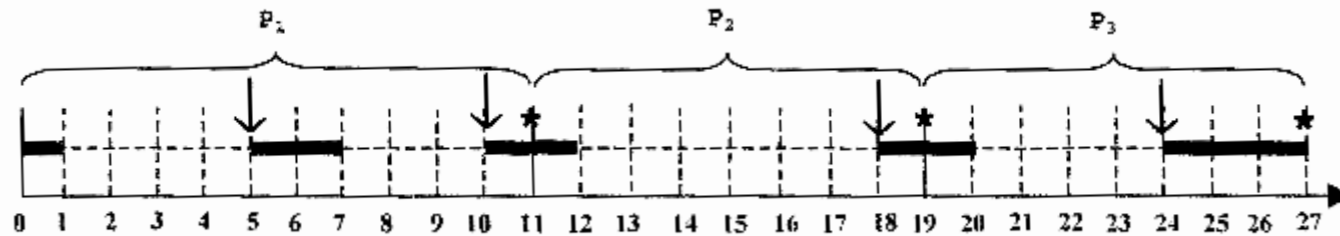


Figura 1.4 Esecuzione sequenziale.

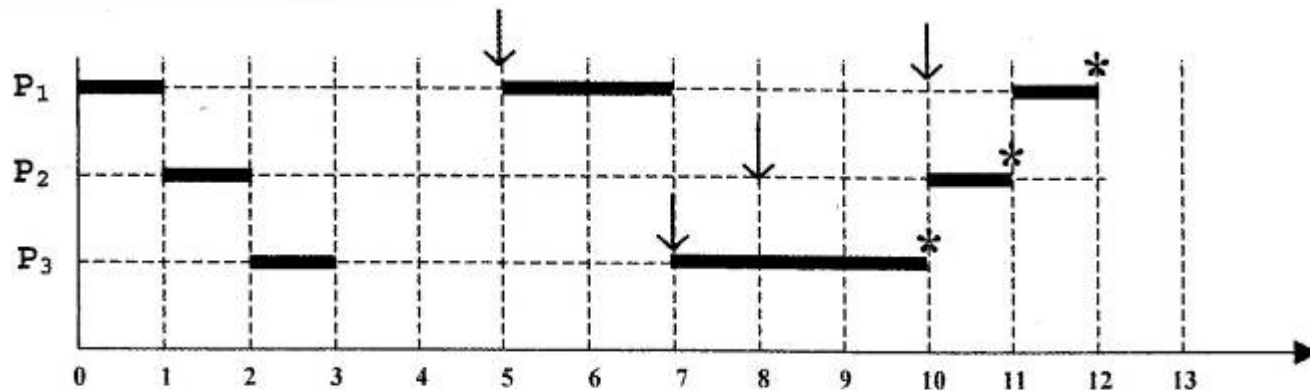


Figura 1.5 Esecuzione in multi-tasking.



# Sistemi multiprogrammati

Gestione contemporanea di  
più programmi nella  
memoria principale (sistema  
multiutente)

sistema operativo
programma applicativo 1
programma applicativo 2
programma applicativo 3

*Sistema Batch multiprogrammato*

*Ottimizzazione dell'uso delle risorse; tempo di risposta*

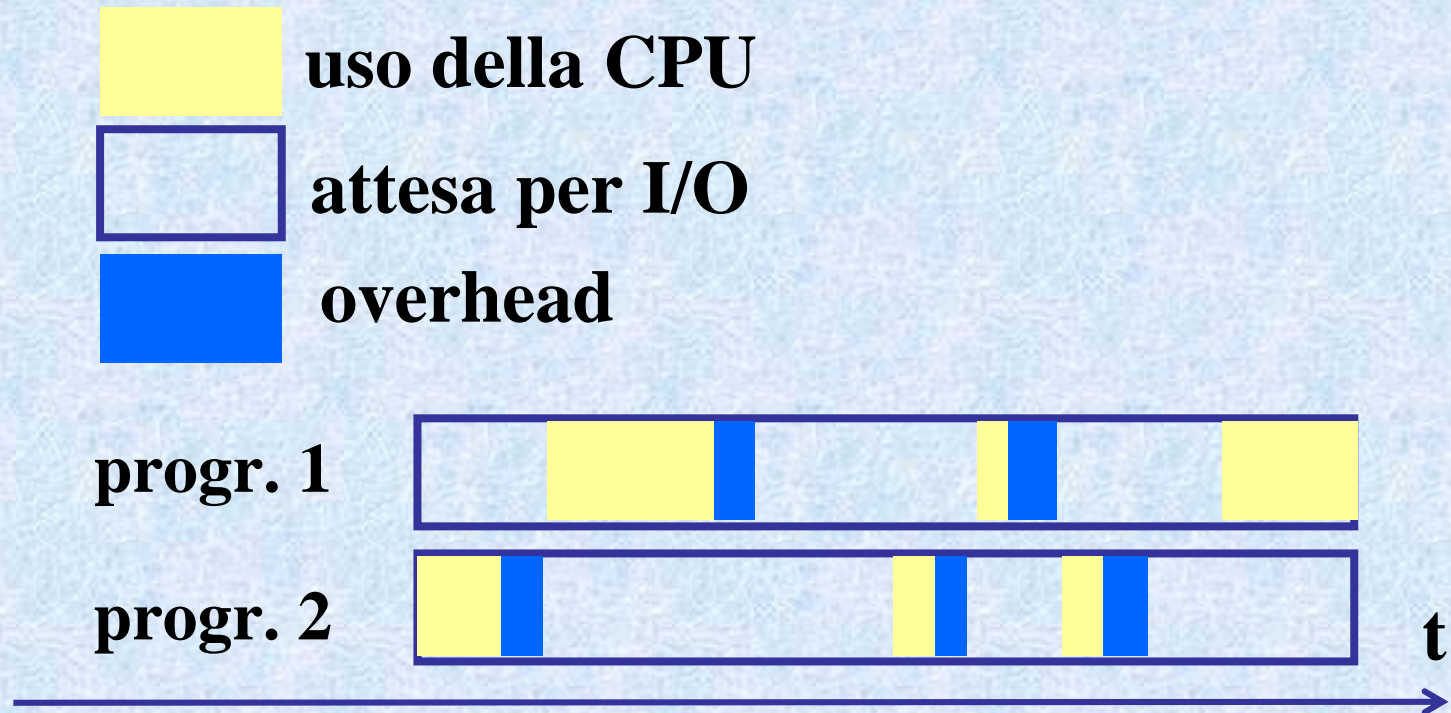
# Sistemi a partizione di tempo (time-sharing)

## Estensione della multiprogrammazione

*(ottimizzazione del tempo di risposta)*

- 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)

# Overhead



## 1.3 Classificazioni dei Sistemi Operativi

### Organizzazione interna:

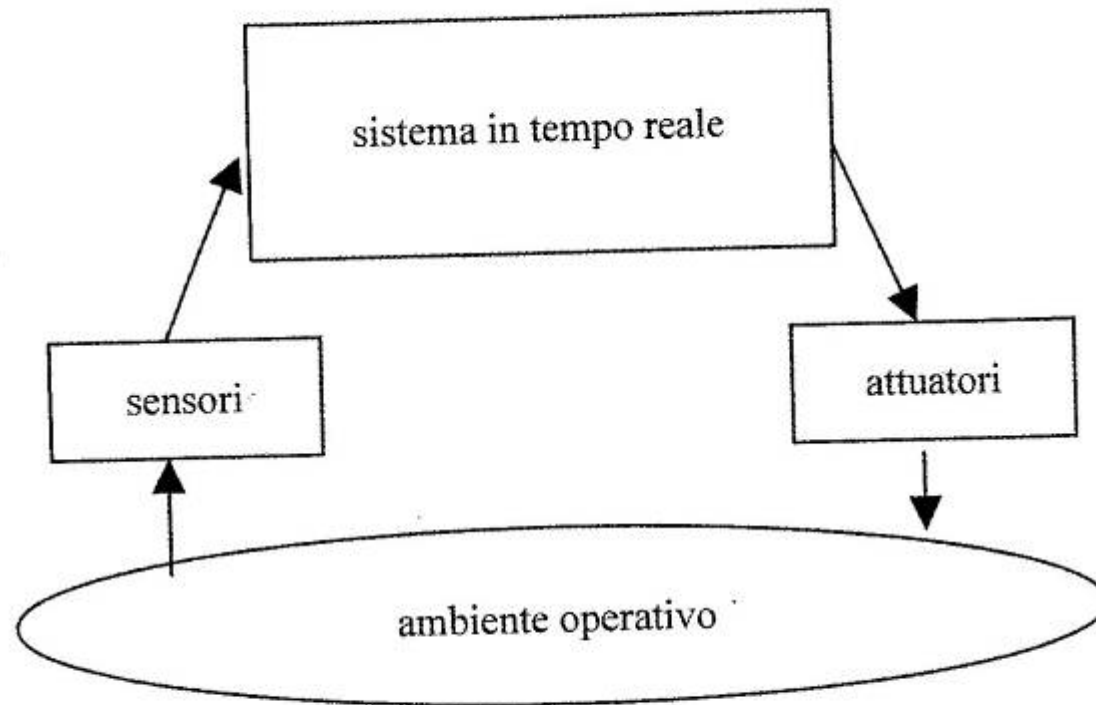
- **monoprogrammati**
- **multiprogrammati**
- **a divisione di tempo**



## **Visibilità utente:**

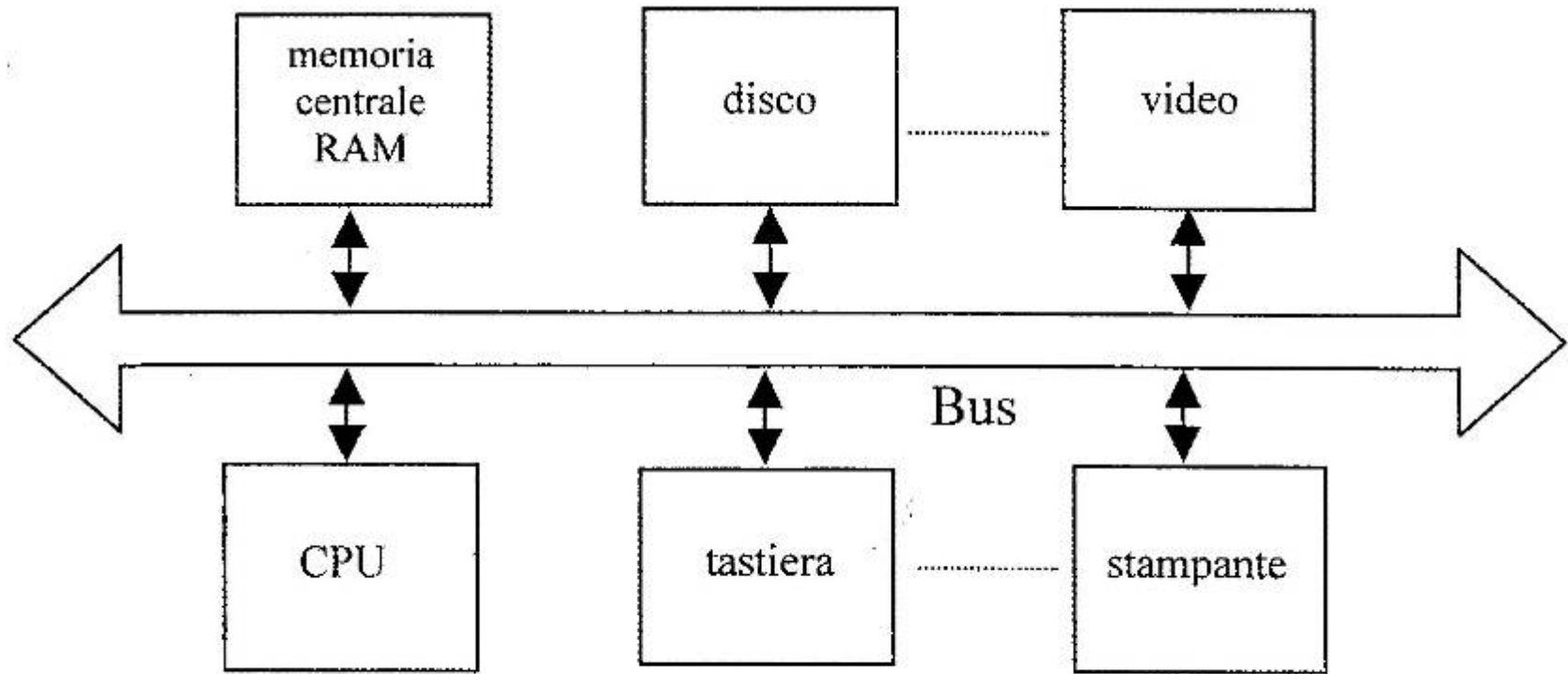
- **Batch**
- **Interattivi**
- **Transazionali**
- **In tempo reale**
- **PC, PDA**
- **Dedicati**
- **Distribuiti**

# Sistema Operativo in tempo reale



**Figura 1.6** Sistema in tempo reale.

## 1.3 Richiami di architettura dei sistemi



**Figura 1.7** Visione semplificata dell'architettura di un sistema a processore singolo.

## 1.3 Richiami di architettura dei sistemi

### Il processore

- Registri generali
- Registri di stato e controllo
  - Program Counter (PC o IP)
  - Stack Pointer (SP)
  - Program Status (PS)

Ciclo di estrazione-esecuzione dell'istruzione (fetch-execute)



## 1.3 Richiami di architettura dei sistemi: il processore

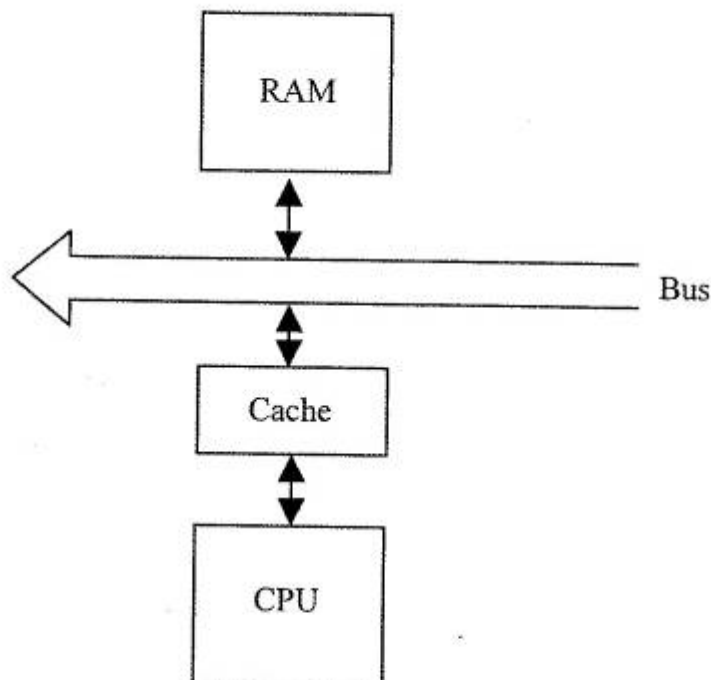
### Registro PS (*Program Status Word*)

- *Condition Code*
- Stato del processore
  - stato utente (*user mode*)
  - stato supervisore (*kernel mode*)
- Abilitazione delle interruzioni (*interrupt enable bit*)

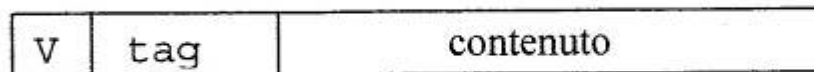
# La memoria

- **Memoria ROM**
- **Memoria RAM**
  - **Cache**
  - **Unità di gestione della memoria (MMU)**
    - rilocalazione, protezione**
    - caricamento dinamico (memoria virtuale)**
    - condivisione dei programmi: rientranza**
- **Memoria di massa**
  - **dispositivi esterni (dischi ...)**

# Memoria Cache



**Figura 1.8** La memoria cache.



**Figura 1.9** Elemento della cache.

## I dispositivi periferici

- **Dispositivi di ingresso/uscita**
- **Dispositivi di memoria di massa**

**Controllore del dispositivo**

**Asincronia rispetto al processore**

**Meccanismi di interazione**

- **Sistema di interruzione**
- **Sistema di accesso diretto alla memoria (DMA)**



## Sistema di interruzione

- Riconoscimento del segnale di interruzione
- Gestione dell'interruzione
  - salvataggio di PC, SP e PS
  - funzione di servizio (*interrupt handler*)
  - ripristino e ritorno (*IRET*)
- Abilitazione e disabilitazione delle interruzioni (*STI, CLI*)
- Politiche

# Meccanismo di interruzione (1)

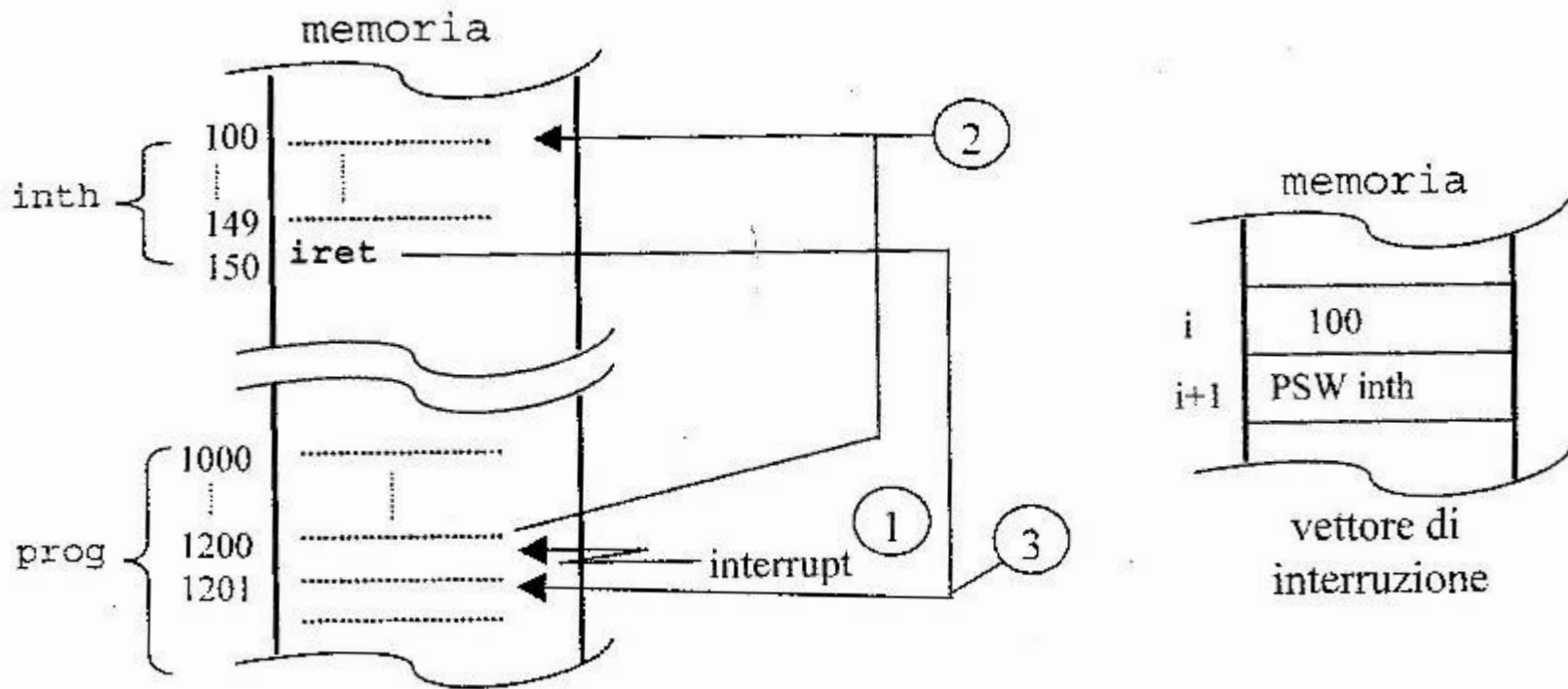


Figura 1.13 Meccanismo d'interruzione.

# Meccanismo di interruzione (2)

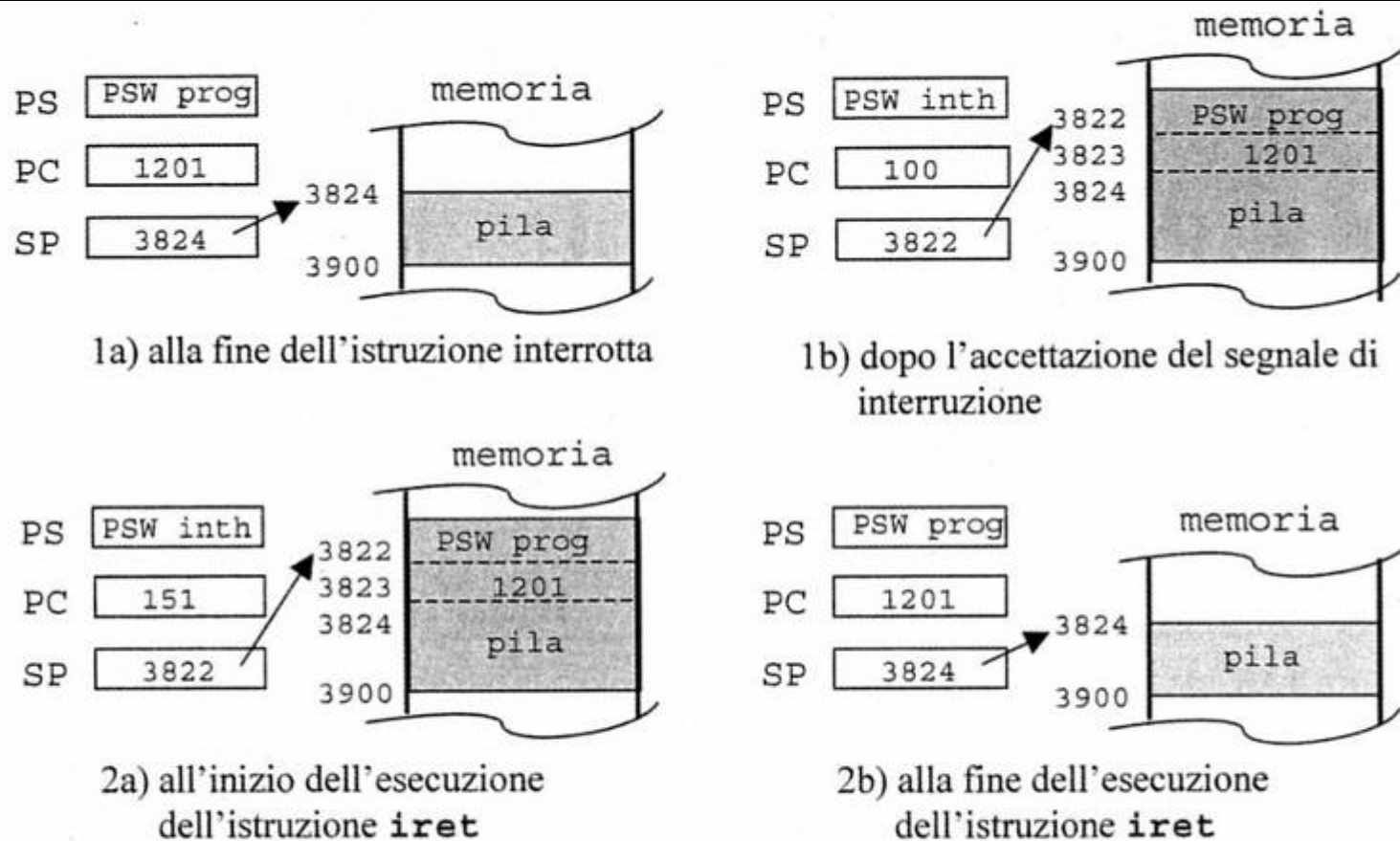


Figura 1.14 Funzionamento dell'istruzione iret.

## Classificazione delle interruzioni

- Interruzioni esterne
- Interruzioni interne (*eccezioni*)
- Interruzioni software (*Supervisor Call - SVC/INT*)



## Sistema di accesso diretto alla memoria

- Canale di DMA
  - registri del canale: puntatore, contatore
- Sottrazione di cicli (*cycle stealing*)
- Relazione con il sistema di interruzione

# Canale di DMA

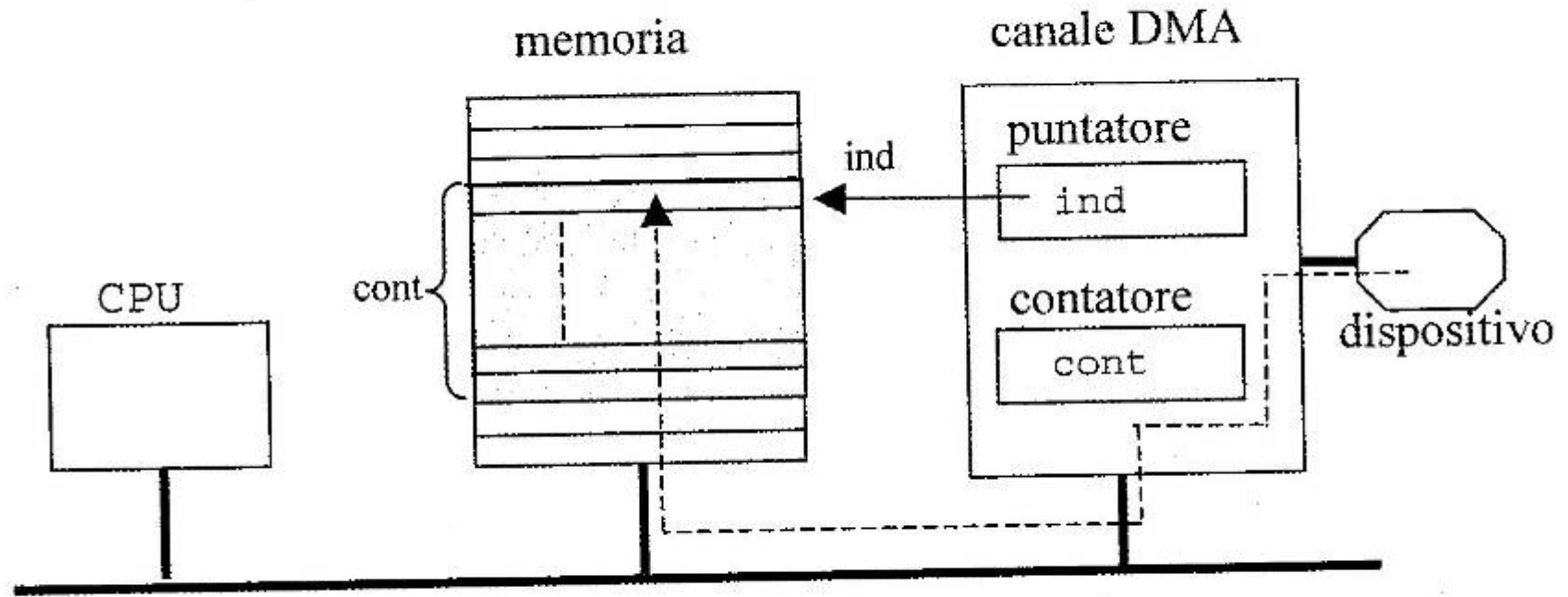


Figura 1.15 Canale di DMA.

## Meccanismi di protezione

- Protezione della memoria
  - registri base e limite
  - altre tecniche (MMU)
  - eccezione di indirizzamento

meccanismi combinati con quelli di protezione
- Protezione del sistema
  - doppio stato del processore (*utente/supervisore*)
  - istruzioni privilegiate

# Modalità di funzionamento di un sistema di calcolo

## **modo utente** (user mode)

- Usato per la normale esecuzione dei programmi
- Istruzioni privilegiate non eseguibili
- Non è possibile accedere liberamente a tutte le risorse del sistema (memoria, dispositivi di I/O)



# Modalità di funzionamento di un sistema di calcolo

## **modo kernel** (supervisor mode)

- Usato per lo svolgimento dei servizi richiesti al Sistema Operativo tramite le system call
- Non esiste alcun limite alle operazioni effettuabili (istruzioni privilegiate, accesso alla memoria)

## Transizione da stato utente a stato supervisore

- chiamate di sistema (istruzione INT o SVC)
- interruzioni di altro tipo
  - salvano la PS attuale e caricano SV del modo supervisore

## Transizione da stato supervisore a stato utente

- istruzione IRET
  - ripristina la PS del modo utente precedentemente salvata

## 1.4 Struttura dei Sistemi Operativi

### Componenti del S.O.

- Gestore del processore
- Gestori della memoria principale e secondaria
- Gestore dei dispositivi di I/O
- Gestore degli archivi (*file*)
- Sistema di protezione e sicurezza
- Gestione della comunicazione tra sistemi distribuiti
- **Interprete dei comandi (??)**

# Gestione del processore

- **Concetto di processo**

“attività controllata da un programma, che si svolge sul (su un) processore”

- stati del processo

- **Processore virtuale**

- realizzato mediante la commutazione del processore fisico



# 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

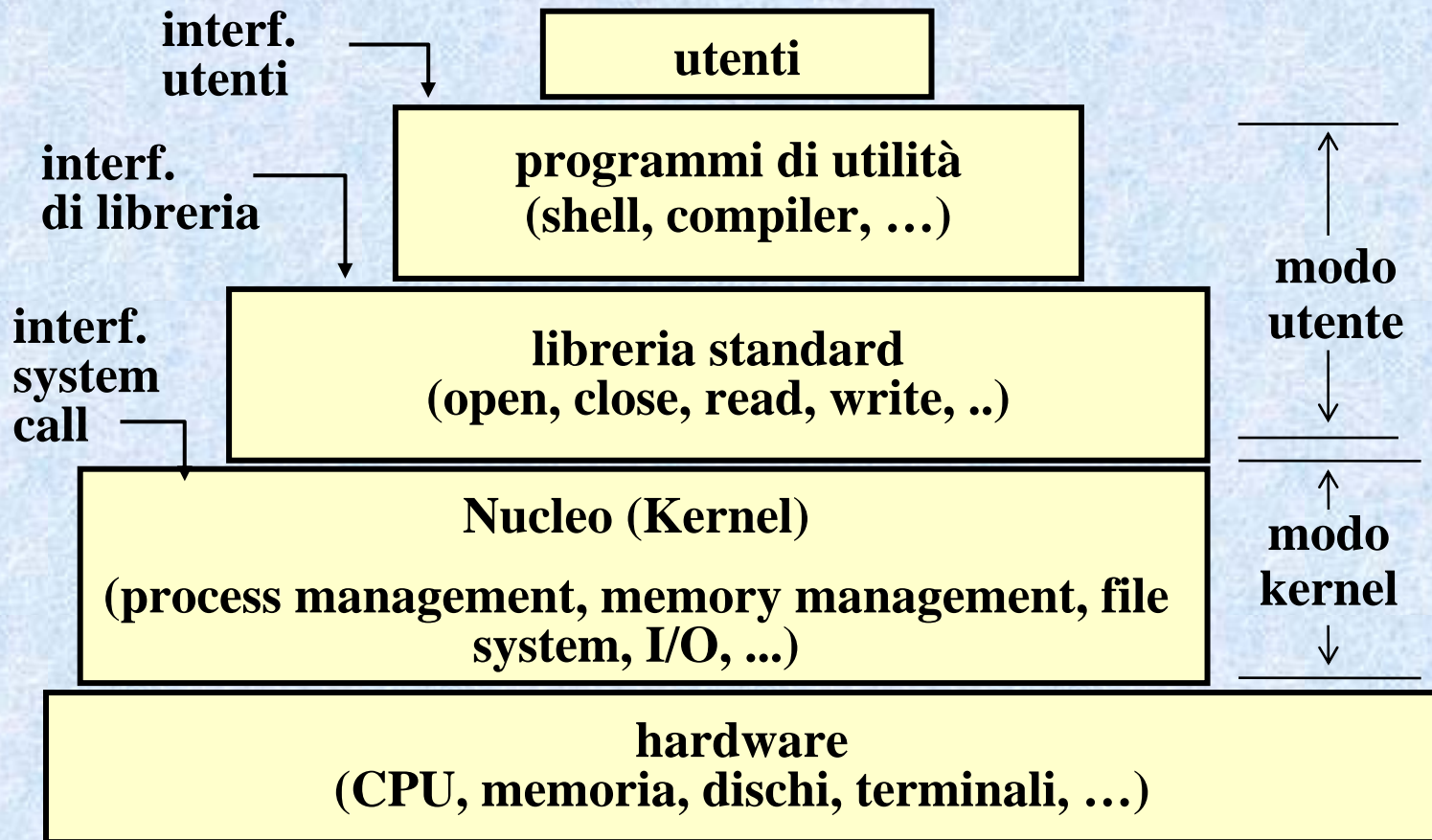
Modello cliente-servitore

# 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 il modo kernel del processore**

# Il sistema UNIX



# Struttura del sistema UNIX

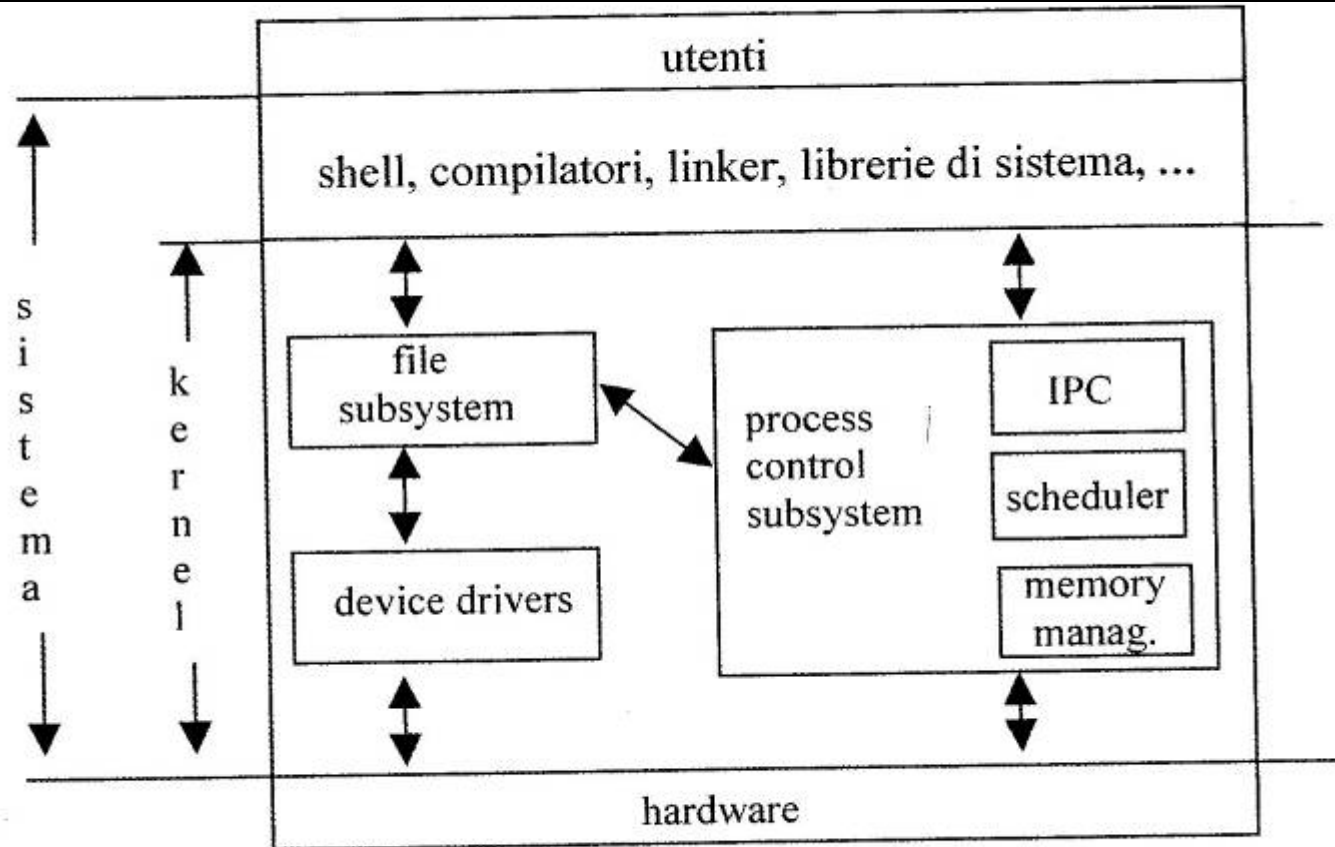


Figura 1.18 Struttura del sistema Unix.



## 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: *read* System Call

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

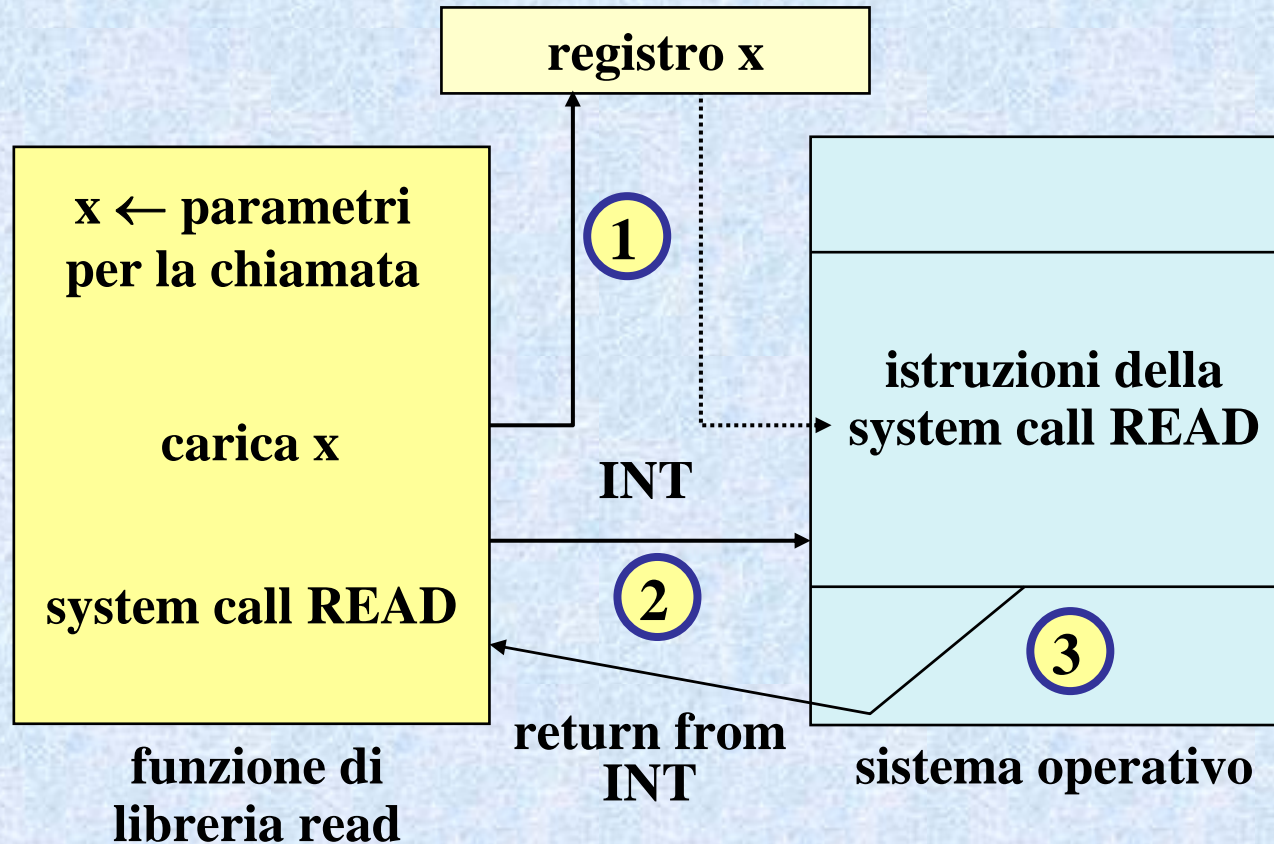
`read` = procedura di libreria

`file` = nome del file

`buffer` = area di memoria per i dati

`Nbyte` = numero di byte da leggere

## Invocazione di chiamate di sistema



# **Sistemi modulari**

**Suddivisione del sistema in moduli, ciascuno destinato a fornire una determinata funzione**

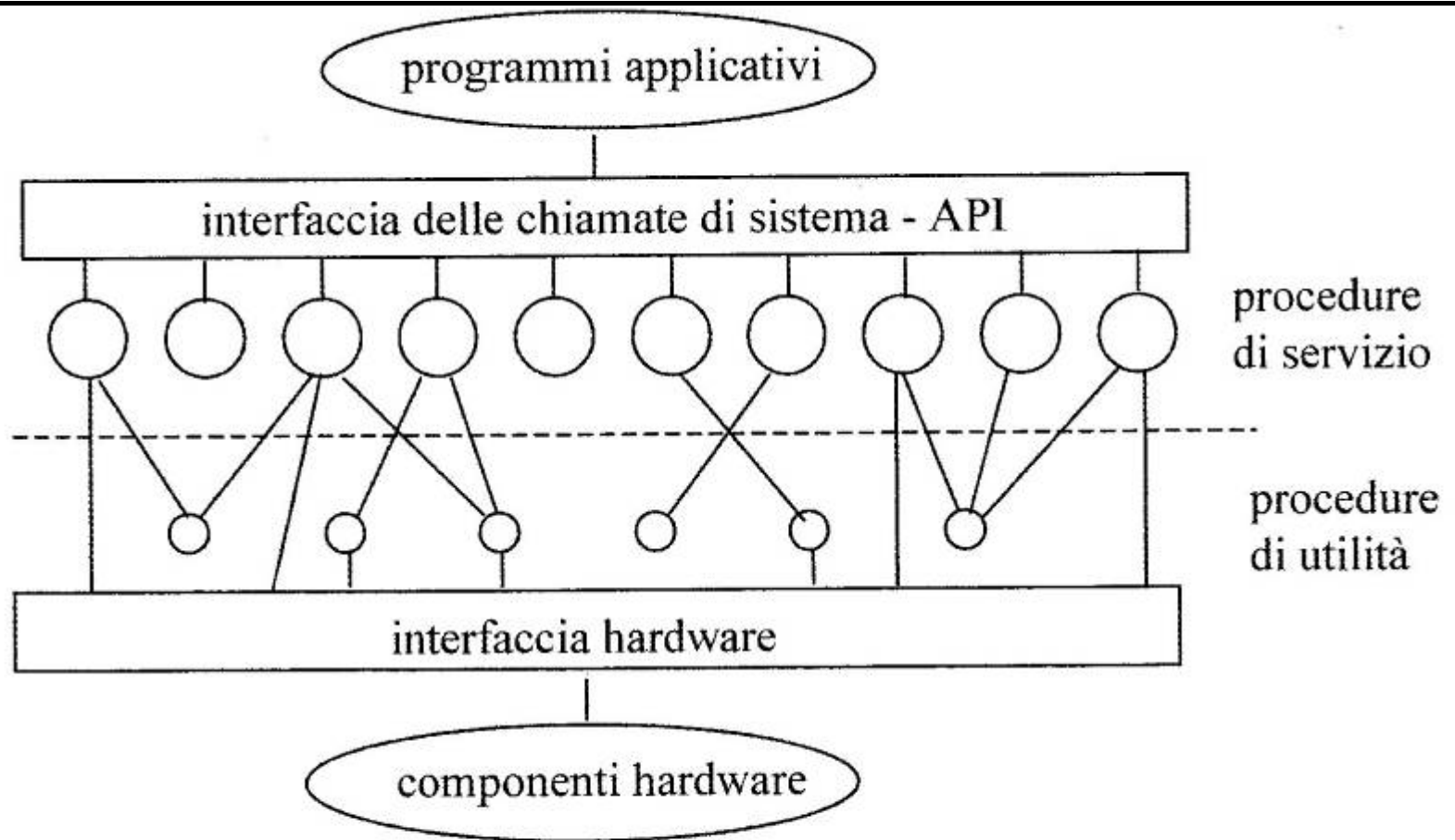
## **INTERFACCIA**

**Specificità delle funzionalità offerte dal modulo**

## **CORPO**

**Realizzazione delle funzionalità non visibile all'esterno**

# Struttura di un sistema modulare



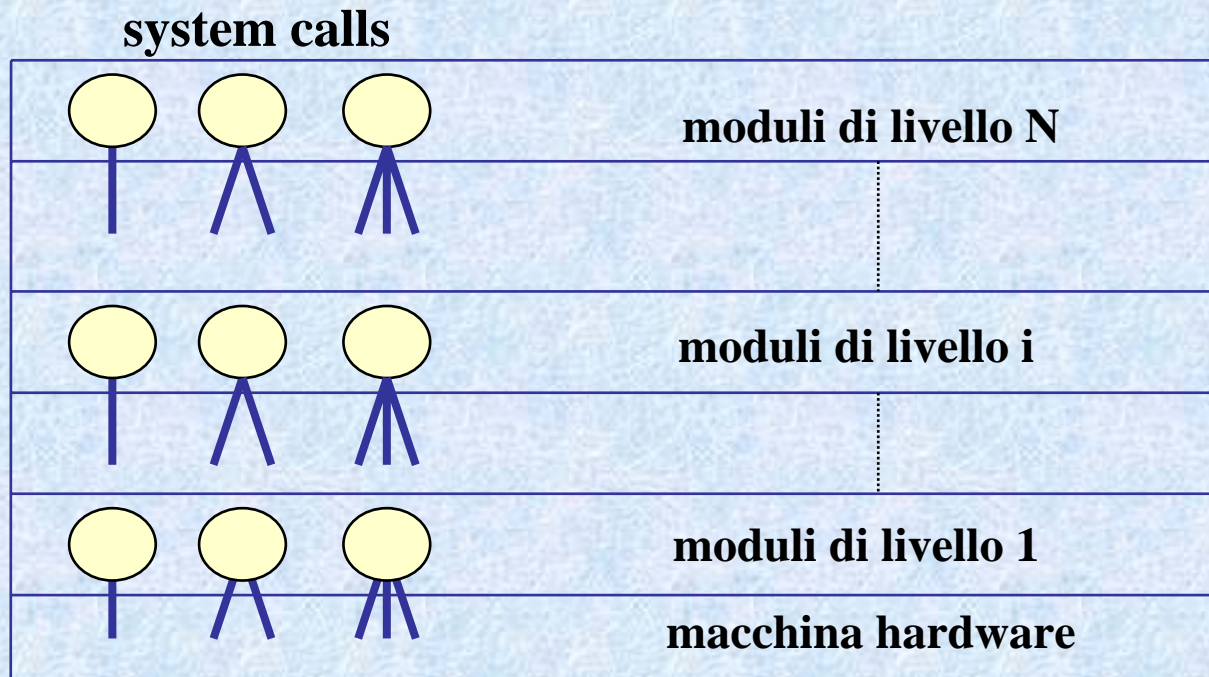
**Figura 1.17** Semplice modello strutturale di un sistema modulare.



# Sistemi a livelli

Le funzioni del S.O. sono organizzate a livelli gerarchici

Ogni livello definisce un tipo di servizio e le modalità per essere utilizzato dai livelli superiori



## Struttura a microkernel

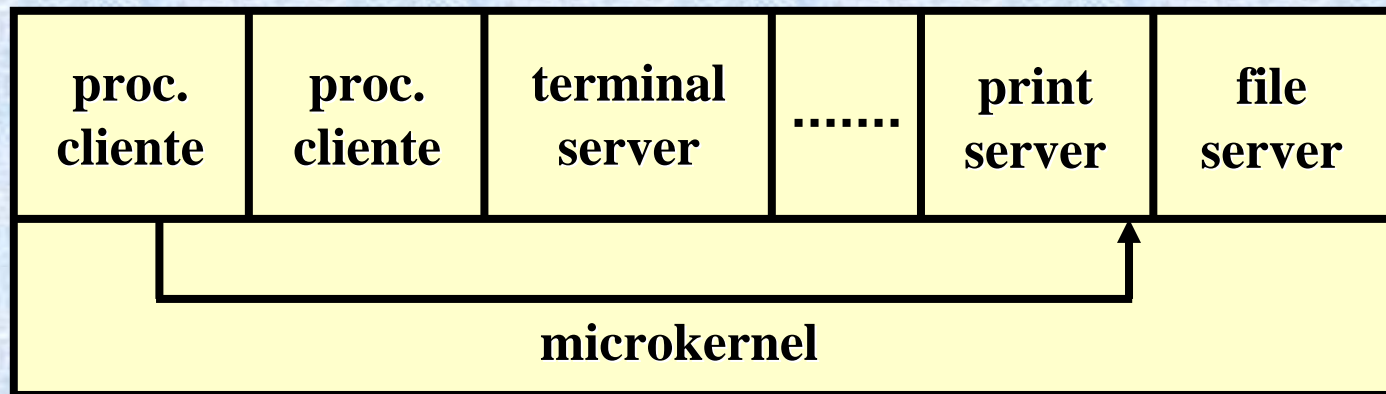
Due componenti del S.O. per ogni risorsa:

- meccanismi per la gestione
- politiche di gestione della risorsa

**Microkernel:** insieme dei meccanismi (stato privilegiato)

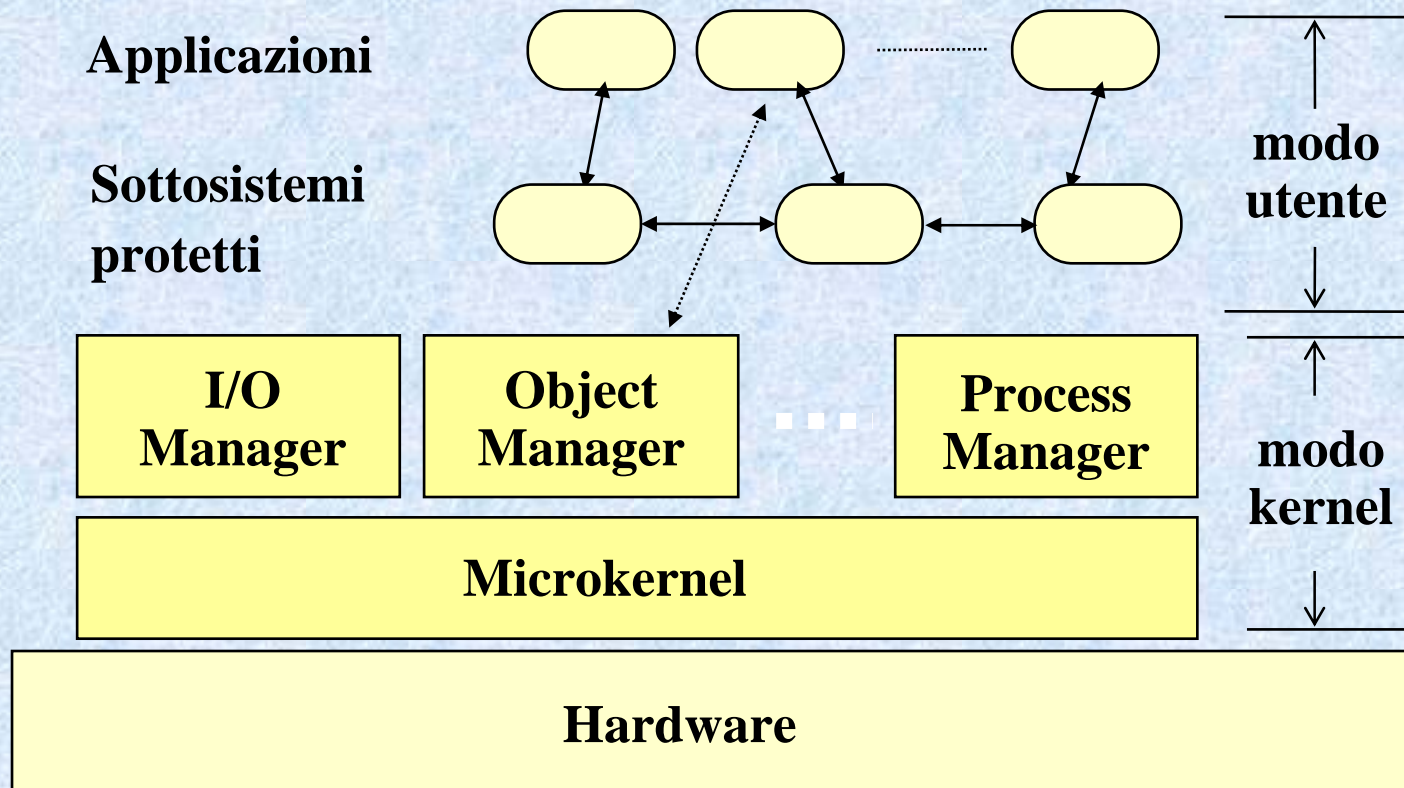
Tutte le politiche sono incorporate nei processi “server” delegate ai servizi di gestione delle risorse (stato utente).

## Struttura a microkernel



- i processi applicativi richiedono i servizi del S.O scambiando messaggi con i processi server

## Windows NT





## Modello cliente-servitore

- Un processo utente (**client**) richiede un servizio (es. lettura di un file) ad un processo di S.O. (**server**)
- Al termine del lavoro, il processo server rispedisce al client un messaggio di risposta
- Client e server operano nello spazio di utente

## Estensione ai sistemi distribuiti

