

## Il Sistema Operativo

## Il Sistema Operativo

- Il **Sistema Operativo** (S.O.) è un insieme di programmi interagenti che consente agli utenti e ai programmi applicativi di utilizzare al meglio le risorse del Sistema di Elaborazione.



- Il S.O. può essere visto come strato di interfaccia fra l'utente e l'hardware.
- L'utente non vede direttamente la macchina, ma una macchina virtuale più semplice da usare e programmare rispetto alla macchina hardware.

## Struttura a livelli del S.O.

- La struttura è organizzata su diversi livelli, cui corrisponde una gerarchia di macchine virtuali.
- Ogni macchina virtuale è un insieme di programmi che realizza funzionalità più potenti, utilizzando i servizi forniti dal livello inferiore.
- Ogni macchina virtuale ha il compito di gestire risorse specifiche del sistema, fornendo meccanismi logici di accesso

<b>Programmi utente</b>
<b>Interprete comandi</b> <i>attivazione programmi d'utente o di sistema</i>
<b>File system</b> <i>controllo e gestione degli accessi a file</i>
<b>Gestione delle periferiche</b> <i>gestione di ingresso/uscita da periferica</i>
<b>Gestione della memoria</b> <i>allocazione e gestione della memoria</i>
<b>Gestione dei processi (nucleo)</b> <i>gestione dei processi e delle interruzioni</i>
<b>Macchina fisica</b>

## Funzioni del S.O.

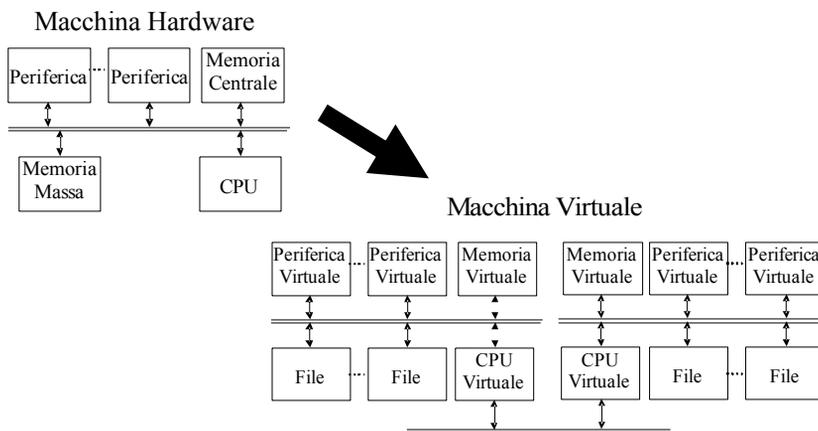
- **Il gestore dei processi (o nucleo) è responsabile dell'esecuzione dei programmi da parte dell'unità di elaborazione**
  - Questo strato offre agli strati superiori una macchina virtuale in cui ciascun programma opera come se avesse a disposizione un'unità di elaborazione dedica.
- **Il gestore della memoria ha la funzione di allocare la memoria e partizionarla tra i vari programmi**
  - Questo strato offre agli strati superiori una macchina virtuale in cui ciascun programma come se avesse a disposizione una memoria dedicata.

## Funzioni del S.O.

- Il gestore delle periferiche è responsabile delle operazioni di *ingresso/uscita* che coinvolgono le periferiche
  - Maschera le caratteristiche hardware delle periferiche dandone una visione astratta.
  - Da l'impressione che ciascun programma abbia a sua completa disposizione una periferica
- Il *file system* è responsabile della gestione dei file in memoria di massa
  - Fornisce una serie di funzionalità per potere operare ad alto livello mascherando i dettagli sulle operazioni che realmente sono effettuate
- *L'interprete comandi* consente all'utente di attivare in modo semplice i programmi

## Macchina virtuale

Le funzioni del S.O. creano una macchina virtuale in cui ogni programma in esecuzione ha l'illusione di avere ad esso dedicate tutte le risorse della macchina hardware



## Processi

- Un processo può essere visto come un programma in fase di esecuzione.
- Un processo é il codice di un programma in esecuzione e lo stato S del processo (l'insieme di valori contenuti in ogni istante nei registri e nelle locazioni coinvolte nell'esecuzione).
- Lo stesso programma può essere associato a più processi:
  - Un programma può essere scomposto in varie parti e ognuna di esse può essere associata ad un diverso processo
  - Lo stesso programma può essere associato a diversi processi quando diverse copie del medesimo processo sono mandate in esecuzione

## Gestione dei processi

- Un sistema di elaborazione opera in due stati:
  - **stato supervisore**: esegue processi del sistema operativo deputati alla gestione del sistema;
  - **stato programma**: esegue processi relativi a programmi a livello applicativo e d'ambiente
- Nello stato supervisore non viene posta alcuna limitazione alle operazioni possibili sull'intero sistema.
- Nello stato programma sono presenti delle limitazioni sulle operazioni eseguibili per proteggere il S.O e gli altri processi.
- Il passaggio da un stato all'altro avviene in seguito ad un *interruzione*.
- La gestione delle interruzioni è compito di un modulo del supervisore, *gestore interruzioni*.

## Interruzioni

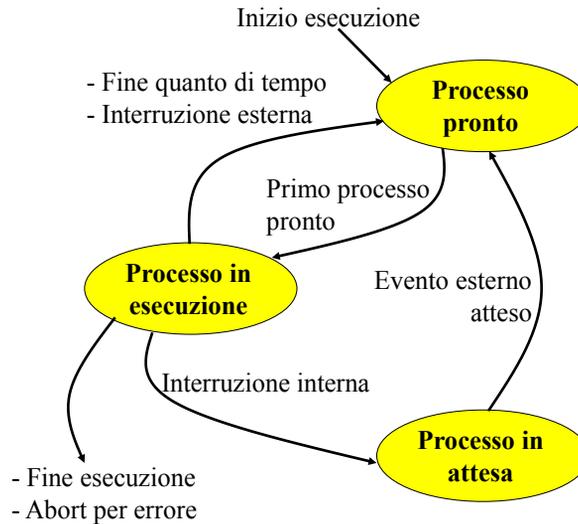
- **Interruzione:** messaggio inviato al supervisore indicante un evento a cui non è possibile fare fronte nello stato programma.
- Tipi di interruzione:
  - *interruzioni interne* (sincrone): generate da un programma in esecuzione, ad esempio per richiedere una operazione di I/O;
  - *interruzioni esterne* (asincrone), provocate dall'esterno per segnalare eventi particolari, esempio la fine di una operazione di I/O.
  - *interruzione per errore* (asincrone), per segnalare guasti hardware o errori software.

## Stati di un processo

Un processo può trovarsi in uno dei seguenti stati

- **In esecuzione:** assegnato al processore ed eseguito da esso
- **Pronto:** può andare in esecuzione, se il gestore dei processi lo decide
- **In attesa:** attende il verificarsi di un evento esterno per andare in stato di *pronto*

## Transizione tra gli stati di un processo

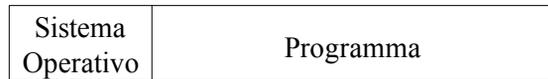


## Interazione tra processi

- I diversi processi possono interagire.
- Le interazioni possono essere classificate in:
  - **indesiderate e impreviste**, quando competono per l'accesso alle risorse, ad esempio nel caso di processi che vogliono usare lo stesso file;
  - **desiderate e previste**, quando i processi cooperano per il raggiungimento di un obiettivo, ad esempio nel caso di processi produttore/consumatore.
- La gestione delle interazione tra processi richiede meccanismi
  - di sincronizzazione tra le attività, quando più processi vogliono accedere alle stesse risorse;
  - di comunicazione per lo scambio di dati, quando un processo deve inviare dei dati ad altri processi;
- La sincronizzazione può avvenire usando il meccanismo dei "semafori logici".
- La comunicazione può avvenire usando
  - uno spazio di *memoria condivisa*;
  - mandando *messaggi* ai processi.

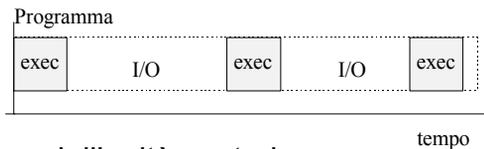
## S.O. Monoprogrammato

In un sistema **monoprogrammato, oltre al S.O.**, un solo programma utente è presente in memoria centrale.



Spazio degli indirizzi del programma

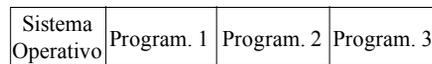
Un calcolatore spende più tempo nelle operazioni di I/O che in quelle di elaborazione dei dati.



Si ha un basso utilizzo dell'unità centrale.

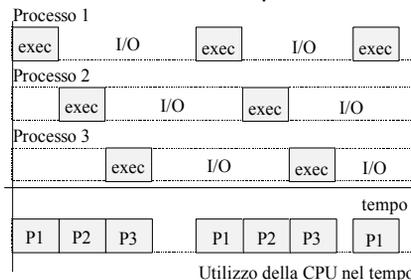
## S.O. Multiprogrammato

- Un sistema è multiprogrammato se più programmi sono residenti in memoria centrale.



Spazio degli indirizzi P1    Spazio degli indirizzi P2    Spazio degli indirizzi P3

- Nel caso in cui venga richiesta una operazione di I/O il sistema operativo manda in esecuzione un altro processo.



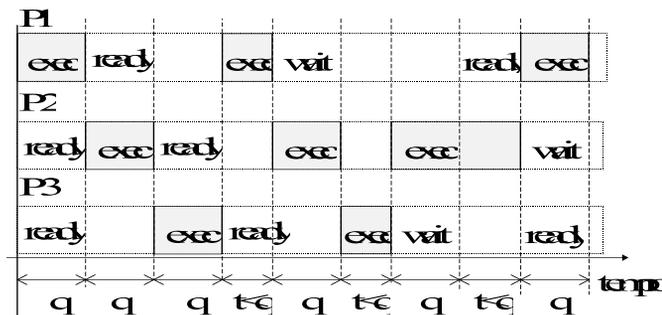
Utilizzo della CPU nel tempo

## Gestore dei processi

- E' il modulo del S.O. destinato ad assegnare l'unità centrale ad uno dei processi pronti.
- L'assegnazione dell'unità centrale viene realizzata usando opportune politiche.
- Gli obiettivi perseguiti nell'assegnazione sono:
  - massimizzare il numero di processi serviti nell'unità di tempo;
  - massimizzare il grado di utilizzo dell'unità centrale;
- Il gestore deve garantire che nessun processo resti in attesa infinita (starvation), che non riesca mai ad avere l'unità centrale.

## Politiche di gestione

- **TIME SHARING.** Il tempo viene diviso in quanti di tempo.
- Il gestore assegna a turno l'unità centrale ai processi pronti per un quanto.



## Politiche di gestione

- Diverse possono essere le politiche di assegnazione dei quanti di tempo.
  - Round Robin
  - Priorità
- Un processo abbandona lo stato di esecuzione se:
  - il quanto di tempo termina;
  - fa una richiesta di I/O, con interrupt interno
- Dimensioni tipiche del quanto di tempo sono di qualche decina di millisecondi

## Gestione della memoria

- Il **gestore della memoria** è responsabile assegnazione della memoria centrale ai programmi in esecuzione
  - Per ciascun processo viene allocata una area di memoria virtuale sufficiente per la sua esecuzione
  - Supera i limiti della memoria fisica, mostrando ai processi uno spazio di memoria virtuale
  - Partiziona la memoria tra i vari processi che la richiedono, garantendo la protezione delle diverse zone di memoria

## Memoria virtuale

- Il gestore della memoria può usare una parte della memoria di massa come area di memoria di appoggio
  - Quando un processo è sospeso, i suoi dati e il codice possono essere spostati su disco per liberare memoria centrale
  - Ciò consente di mostrare ai processi una memoria virtuale di dimensione superiore alla memoria centrale

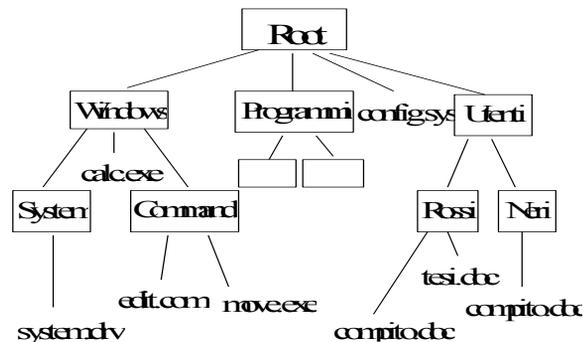
## Gestione delle periferiche

- Il gestore delle periferiche fa in modo che ogni programma veda una **periferica virtuale** dedicata
- Il gestore delle periferiche consente la gestione delle periferiche in modo indipendente dalle caratteristiche fisiche dei dispositivi.
- Garantisce agli utenti un'interazione semplice attraverso comandi ad alto livello.

## Gestione dei file

- Il **gestore dei file** (o **file system**) è responsabile della gestione della memoria secondaria (dischi)
- I dischi sono fisicamente organizzati in superfici, cilindri e settori
- Il gestore dei file consente l'uso della memoria secondaria mediante una organizzazione logica in volumi, cartelle e file
  - un **file** è una sequenza di dati di dimensioni variabili, con accesso sequenziale oppure casuale
  - una **cartella** (o **direttorio**) è un contenitore di un insieme di file e cartelle
  - un **volume** è un contenitore di file e cartelle, solitamente corrispondente ad una unità a disco

## Organizzazione ad albero dei file



## Interprete comandi

- L'**interprete comandi** è l'interfaccia utente del sistema operativo
- Definisce le operazioni che possono essere utilizzate direttamente dall'utente finale, chiamate **comandi**
- Esempi di comandi sono
  - esecuzione di una applicazione
  - apertura di un documento (eseguendo l'applicazione responsabile della manipolazione di quel tipo di documento)
- L'**interprete comandi** può essere
  - basato su una interfaccia a caratteri come nei sistemi operativi Unix, ma anche nel "prompt dei comandi" di Windows
  - basato su una interfaccia grafica come nei sistemi operativi Windows e nell'ambiente X Window di Unix