

INFORMATICA DI BASE

- 6 crediti -

Docente: Michele Piana

Email: michele.piana@univr.it

URL: <http://www.di.univr.it/~piana>

Ricevimento: Lunedì ore 14:00 – 15:00

Studio: 2.05 (Ufficio del Preside)

PROMEMORIA

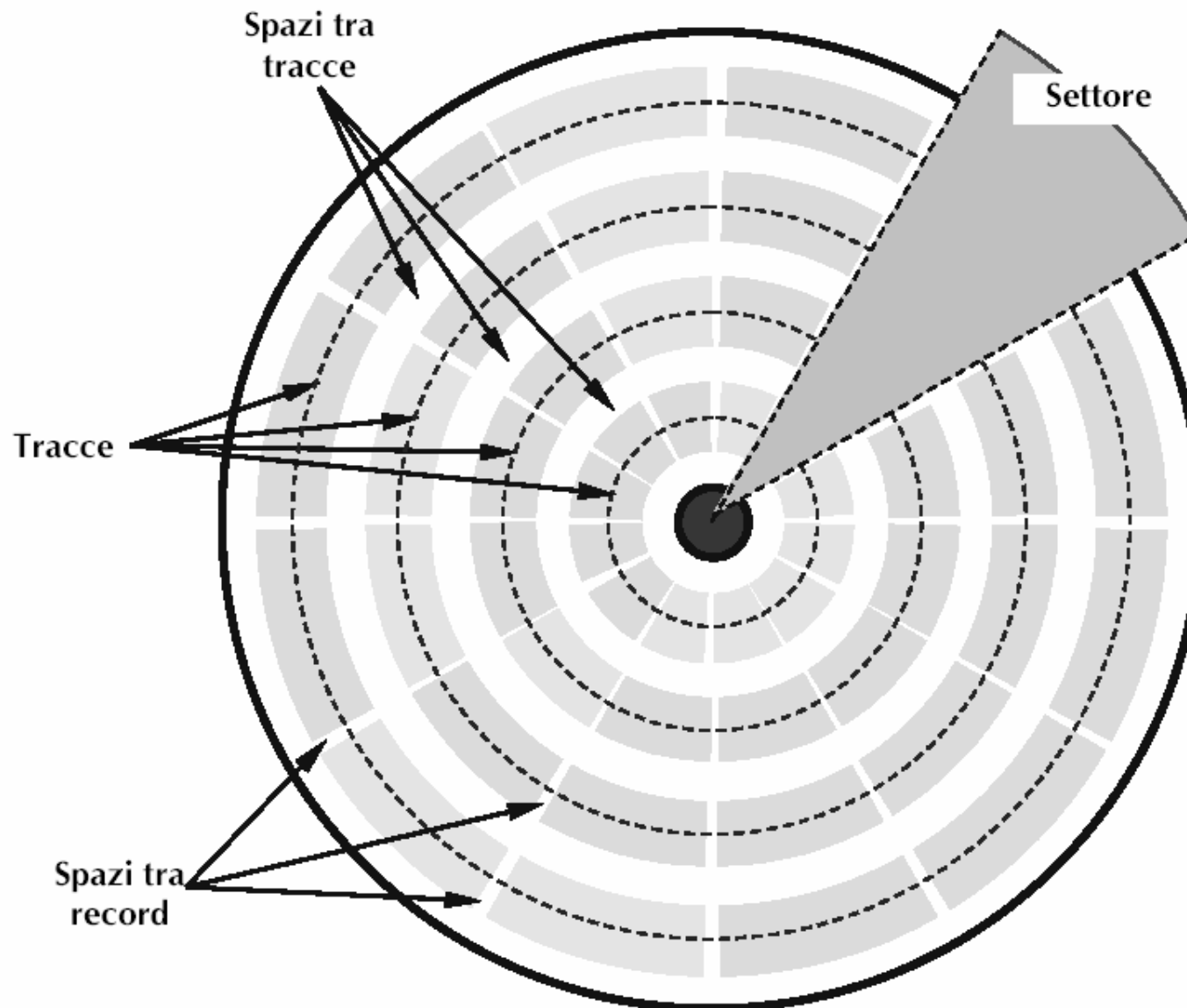
LUNEDI 24 APRILE NON C'E' PONTE

RIASSUNTO DELLA LEZIONE PRECEDENTE

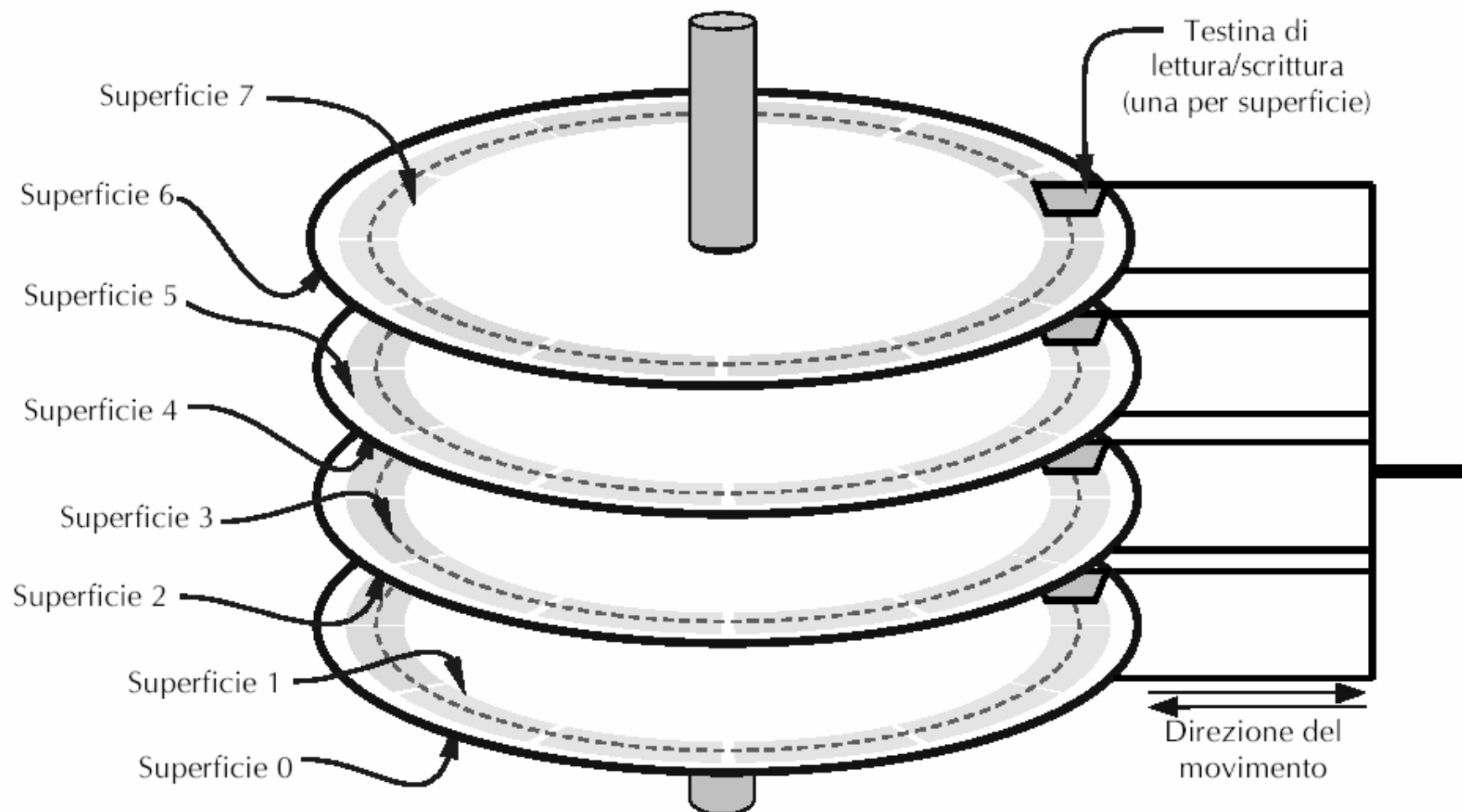
MEMORIA DI MASSA

- **Non volatilità**
- **Grande capacità**
- **Contiene:**
 - **Programmi che servono agli utenti**
 - **Dati**
 - **Programmi che servono al funzionamento del sistema (S.O.)**
- **Funzionano anche in assenza di alimentazione**
- **Sono basati su tecnologie magnetiche e ottiche**

Tracce e settori

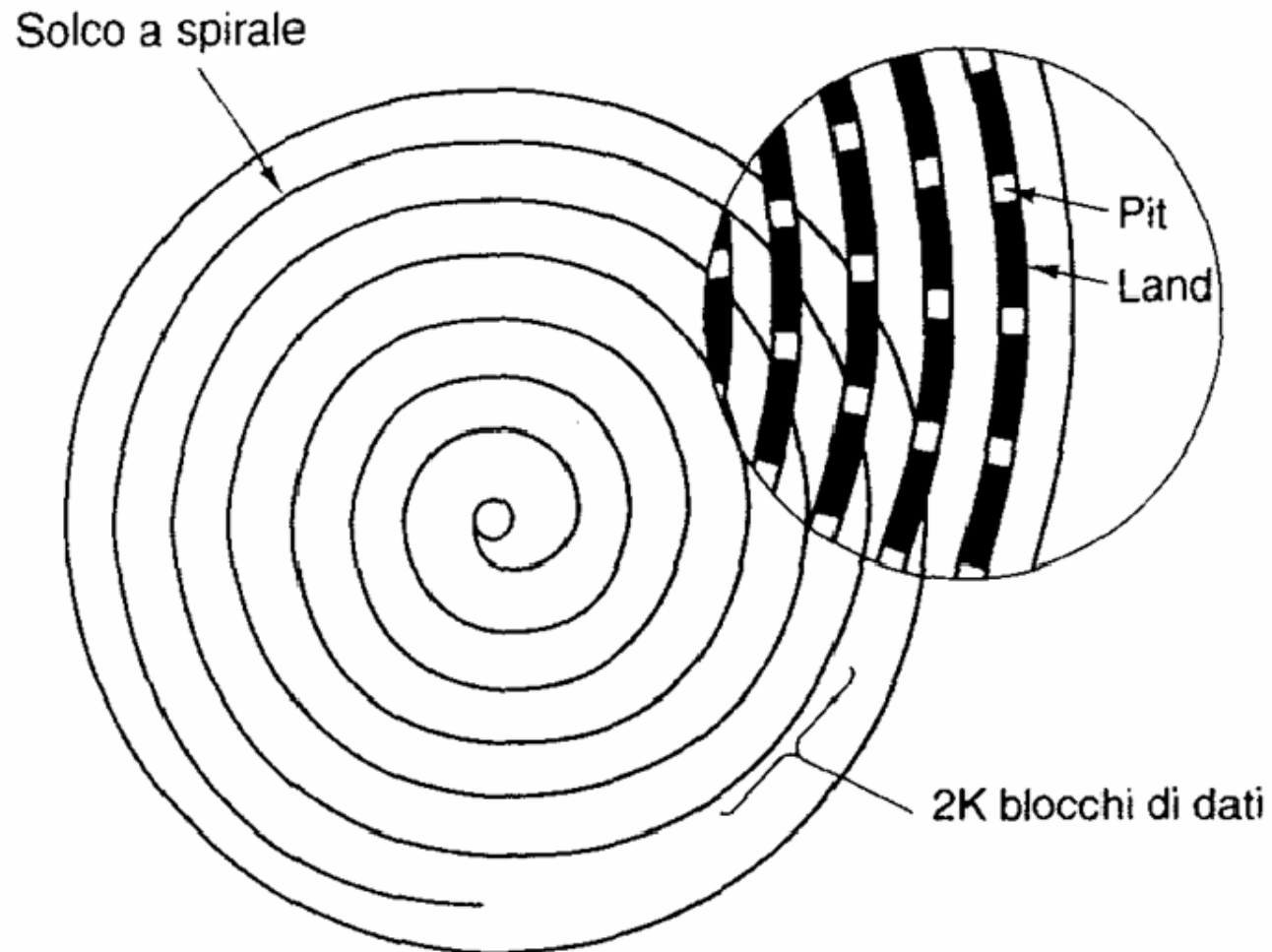


Schema di un Hard Disk



Le tracce in grigio formano un "cilindro"

Pit e land su un CD



INTERFACCIA I/O

- **La comunicazione tra il calcolatore e l'ambiente esterno avviene attraverso dei dispositivi di I/O detti *periferiche***
- **La periferica viene controllata tramite interfacce che traducono i segnali interni al calcolatore in formati comprensibili alla periferica**

PORTE di I/O

- Le interfaccie sono realizzate tramite schede che vengono inserite sulla scheda madre (ad incastro o cablate)
- La periferica si collega attraverso opportuni connettori detti *porte di I/O*

TASTIERA

- **Meccaniche o elettromagnetiche**
- **Quando si preme un tasto viene generato un segnale di *interruzione* che viene mandato al *gestore degli interrupt di tastiera*. Ad ogni tasto si associa un segnale diverso**
- **Quando il tasto viene rilasciato si verifica una seconda interruzione**

Monitor CRT (Cathode Ray Tube)

- Un cannone spara un **raggio di elettroni** contro uno **schermo fosforescente** (per la riproduzione dei colori si usano tre cannoni, per il **rosso**, il **verde** e il **blu**).
- Il raggio viene **deflesso** in modo da coprire tutti i punti dello schermo, una riga per volta (**raster scan**).
- Un'immagine a schermo pieno viene completata **30/60** volte al secondo.
- Davanti allo schermo c'è una **griglia** che lo divide in **punti**:
 - quando la griglia ha una **carica positiva** gli elettroni vengono accelerati **raggiungono lo schermo**;
 - quando la griglia ha una **carica negativa** gli elettroni vengono respinti e il punto sullo **schermo rimane spento**.

LCD (Liquid Crystal Display)

- Schermi “**piatti**”, **leggeri** e facilmente trasportabili.
- **Cristalli liquidi**: molecole organiche viscosi
 - **scorrono** come un liquido;
 - hanno una struttura **tridimensionale**, come un **cristallo**;
 - quando tutte le molecole sono allineate le proprietà ottiche del cristallo dipendono da **direzione** e **polarizzazione** della luce in ingresso;
 - un **campo elettrico modifica** l'allineamento molecolare e quindi **le proprietà ottiche**.

STAMPANTI

- **E' la periferica che riporta su carta i risultati di una elaborazione**
- **Stampanti ad aghi: la testina di scrittura contiene un insieme di aghi pilotabili singolarmente. I caratteri si ottengono dalle diverse configurazioni di punti (aghi) stampati su carta.**

STAMPANTI (II)

- Stampanti a getto d'inchiostro: i caratteri si formano da configurazioni di punti realizzati trasferendo sulla carta gocce microscopiche di inchiostro.
- Sono *economiche, silenziose (rispetto alle stampanti ad aghi), ottima qualità di stampa, colore.*

STAMPANTI(III)

- **Stampanti laser: l'immagine della pagina da stampare viene impressa su un rullo di materiale fotosensibile da un raggio laser che carica elettricamente le zone raggiunte; queste zone attirano le particelle di inchiostro (toner) sulla carta.**
- ***Qualità di stampa elevata, buona velocità, costi non eccessivi.***

Interazione mouse-computer

- Ogni volta che si **sposta**, il mouse invia una sequenza di 3 byte al calcolatore lungo una linea **seriale**:
 - un intero che indica lo **spostamento X**;
 - un intero che indica lo **spostamento Y**;
 - un intero che indica lo stato dei **pulsanti**.
- Il SO accetta queste informazioni e converte le indicazioni **relative** inviate dal mouse nella posizione **assoluta** del cursore.

MODEM

- Dispositivi per il trasferimento dei dati da un calcolatore ad utilizzando la tradizionale rete telefonica
- *Modulatore e demodulatore ?* Occorre tradurre i segnali da *digitale* ad *analogico* (Computer? Rete telefonica) e da *analogico* a *digitale* (Rete telefonica ? Computer)
- Attualmente raggiunge i 56 Kbit/s

RETI di CALCOLATORI

- **Insieme di calcolatori autonomi tra loro collegati mediante una rete di comunicazione**
- **Gli utenti sono in grado di interagire in modo esplicito con la rete**
- **I calcolatori connessi alla rete mantengono un grado di interdipendenza: in caso di guasto della rete (o parte della rete) i singoli calcolatori continuano a funzionare**

Tassonomia delle reti:

2. la dimensione delle reti

- **Reti locali (Local Area Network, LAN)**
 - di limitata estensione
 - collegano dispositivi collocati nello stesso edificio o in edifici adiacenti.
- **Reti metropolitane (Metropolitan Area Network, MAN)**
 - collegano di dispositivi collocati nella stessa area urbana.
- **Reti geografiche (Wide Area Network, WAN)**
 - collegano di dispositivi diffusi in un'ampia area geografica (nazione, continente, ...);
- **"Reti di reti" (Internetwork),**
 - collegameno più reti differenti (in termini sia hardware che software) mediante opportuni elementi di interfaccia, che si possono estendere su tutto il pianeta (e.g. Internet).

CONTENUTO DELLA LEZIONE

- **Il sistema operativo: generalita'**
- **Il sistema operativo: funzionalita'**
- **Multiprogrammazione**
- **Processi**

Il Sistema operativo (S.O.)

- **I componenti fisici sono accessibili attraverso software dedicati**

Software di sistema o di base

- **Sono forniti assieme all'hardware**
- **L'insieme di questi programmi forma il *Sistema Operativo***

FUNZIONI DEL SISTEMA OPERATIVO - I

Il SO rende utilizzabili le risorse fisiche presenti nel sistema informatico

In particolare, il SO:

- 1. Garantisce correttezza e precisione nell'elaborazione dell'informazione**
- 2. Consente all'utente di superare il problema della localizzazione delle risorse sfruttando la presenza di una rete che permetta di accedere alle applicazioni da ogni luogo e in ogni momento**
- 3. Garantisce il massimo livello di affidabilit , disponibilit  e sicurezza del sistema**

FUNZIONI DEL SISTEMA OPERATIVO - II

- 4. Assicura la privatezza dei dati**
- 5. Consente la realizzazione di soluzioni aperte, che permettano l'interoperabilit  tra elementi forniti da diversi produttori di hardware e di software**
- 6. Supera i problemi legati alla limitazione del numero di risorse e ne regola l'impiego evitando conflitti di accesso**

FUNZIONI DEL SISTEMA OPERATIVO - III

Il sistema operativo puo' essere inteso come uno strumento che virtualizza le caratteristiche dell'hardware sottostante, offrendo di esso la visione di una macchina astratta piu' potente e piu' semplice da utilizzare di quella fisicamente disponibile

SO: funzionalità

- **SO come GESTORE DELLE RISORSE,**
controlla tutte le risorse del calcolatore e le gestisce in modo efficiente:
 - tiene traccia di chi utilizza la risorse
 - accetta e soddisfa le richieste di utilizzo di una risorsa
 - fa da mediatore fra le risorse che risultano in conflitto.
- **SO come MACCHINA ESTESA:**
 - costituisce la base su cui è possibile scrivere i programmi applicativi.
 - presenta all'utente una macchina estesa più facile da programmare dell'HW sottostante.

FUNZIONI DI SERVIZIO DEL SO - I

1. Esecuzione di applicazioni

- **Caricamento del programma (istruzione e dati) nella memoria centrale**
- **Inizializzazione dei dispositivi di ingresso/uscita**
- **Preparazione e gestione di altre risorse (rete di comunicazione)**

2. Accesso ai dispositivi di ingresso e uscita

- **Gestione dei segnali necessari per il trasferimento dei dati**
- **Consente all'utente di ragionare in termini di operazioni astratte di lettura e scrittura**

FUNZIONI DI SERVIZIO DEL SO - II

3. Archiviazione di dati e programmi

- **Fornire un'organizzazione logica dei dati sotto forma di cartelle (directory) e file**
- **Gestire le operazioni di basso livello per l'i/o**

4. Controllo di accesso

- **Condivisione di risorse da parte di piu' utenti o applicazioni**
- **Meccanismi di protezione e politiche di risoluzione degli eventuali conflitti**

FUNZIONI DI SERVIZIO DEL SO - III

5. Contabilizzazione

- **Ottimizzare il tempo di risposta dei programmi interattivi**
- **Far conoscere all'utente il costo dell'impiego del sistema**

6. Gestione dei malfunzionamenti

- **Rilevare e risolvere malfunzionamenti provocati da guasti hardware o da operazioni scorrette compiute dal software applicativo**

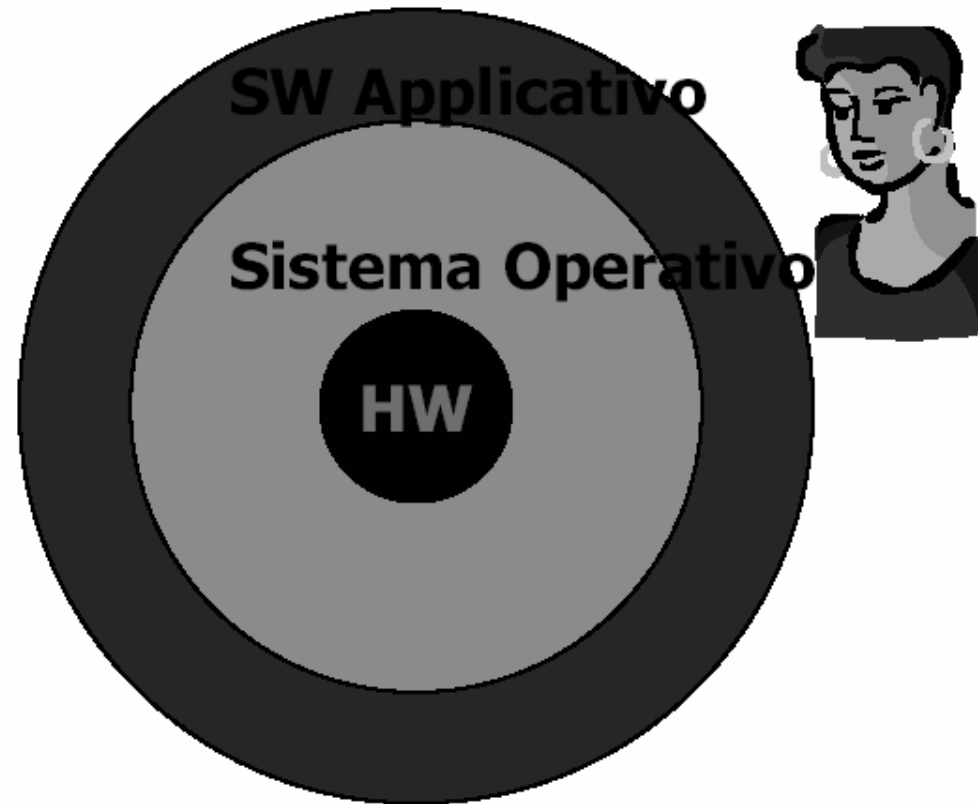
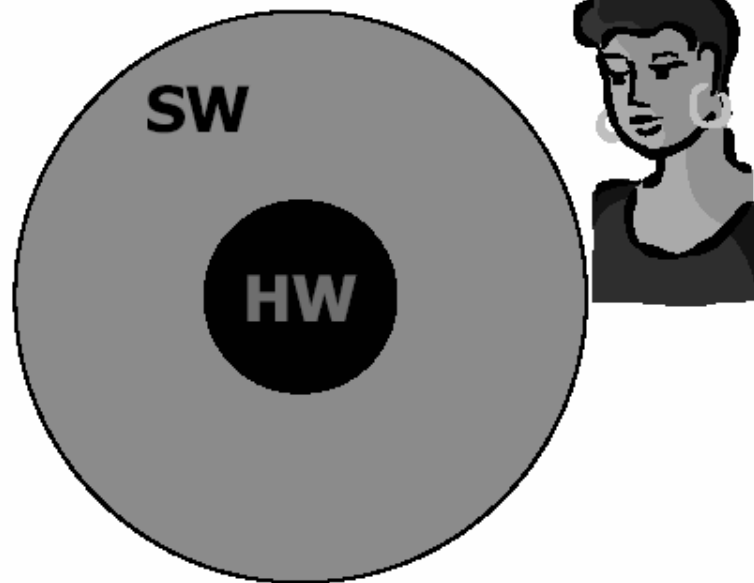
Vantaggi di un SO

- Sono legati alla possibilità di definire modalità standard di interfaccia con i dispositivi fisici, cosicché sia possibile:
 - sviluppare programmi in modo semplice, modulare ed indipendente dallo specifico calcolatore su cui viene fatto funzionare il sistema operativo;
 - aggiornare il software di base e l'hardware in modo trasparente ai programmi applicativi e all'utente, ossia senza che vengano influenzati dall'operazione.

Visioni fornite da un SO

- **Dall'alto:** il sistema operativo fornisce all'utente un'interfaccia conveniente.
- **Dal basso:** gestisce tutti le parti di un sistema complesso, allocando in modo ordinato le diverse risorse della macchina: processori, memorie, dischi, interfacce di rete, stampanti e altre periferiche.

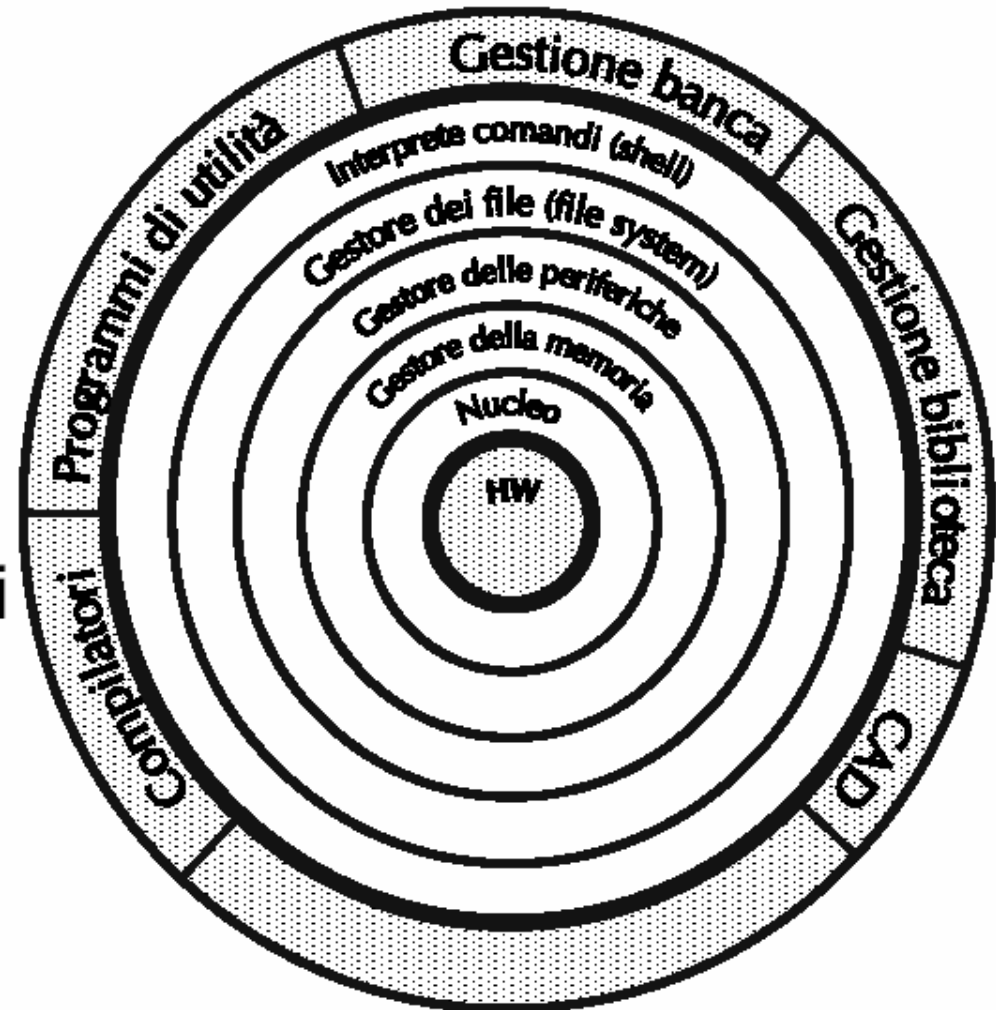
Il software



- $SW = \text{Sistema Operativo} + \text{SW applicativo}$
- Il S.O. come necessario **intermediario**

Organizzazione di un SO

- Gerarchia di "macchine virtuali"
- La visione della macchina virtuale a livello **n** è quella fornita dall'HW e dagli strati del SO fino all'ennesimo (incluso)



Organizzazione a “strati”

- Ogni macchina virtuale è un insieme di programmi che realizza delle funzionalità che utilizzano i servizi forniti a livello inferiore.
- Ogni macchina virtuale ha il compito di gestire risorse specifiche di sistema regolandone l’uso e mascherandone i limiti.
- I **meccanismi** che garantiscono la correttezza logica sono separati dalle **politiche** di gestione (maggiore flessibilità).



ogni “strato” risolve un problema specifico

ELEMENTI DI UN SO - I

1. Sistema di gestione del processore

- Controlla la CPU**
- Definisce quali programmi sono da eseguire e quali compiti sono da assegnare alla CPU**

2. Sistema di gestione della memoria

- Controlla l'allocazione della memoria di lavoro ai diversi programmi che possono essere contemporaneamente in esecuzione**

ELEMENTI DI UN SO - II

3. Sistema di gestione delle periferiche

- **Garantisce l'accesso ai dispositivi di i/o**
- **Maschera i dettagli di basso livello e gli eventuali conflitti che possono insorgere nel caso che diverse richieste arrivino contemporaneamente allo stesso dispositivo**

4. Sistema di gestione dei file (file system)

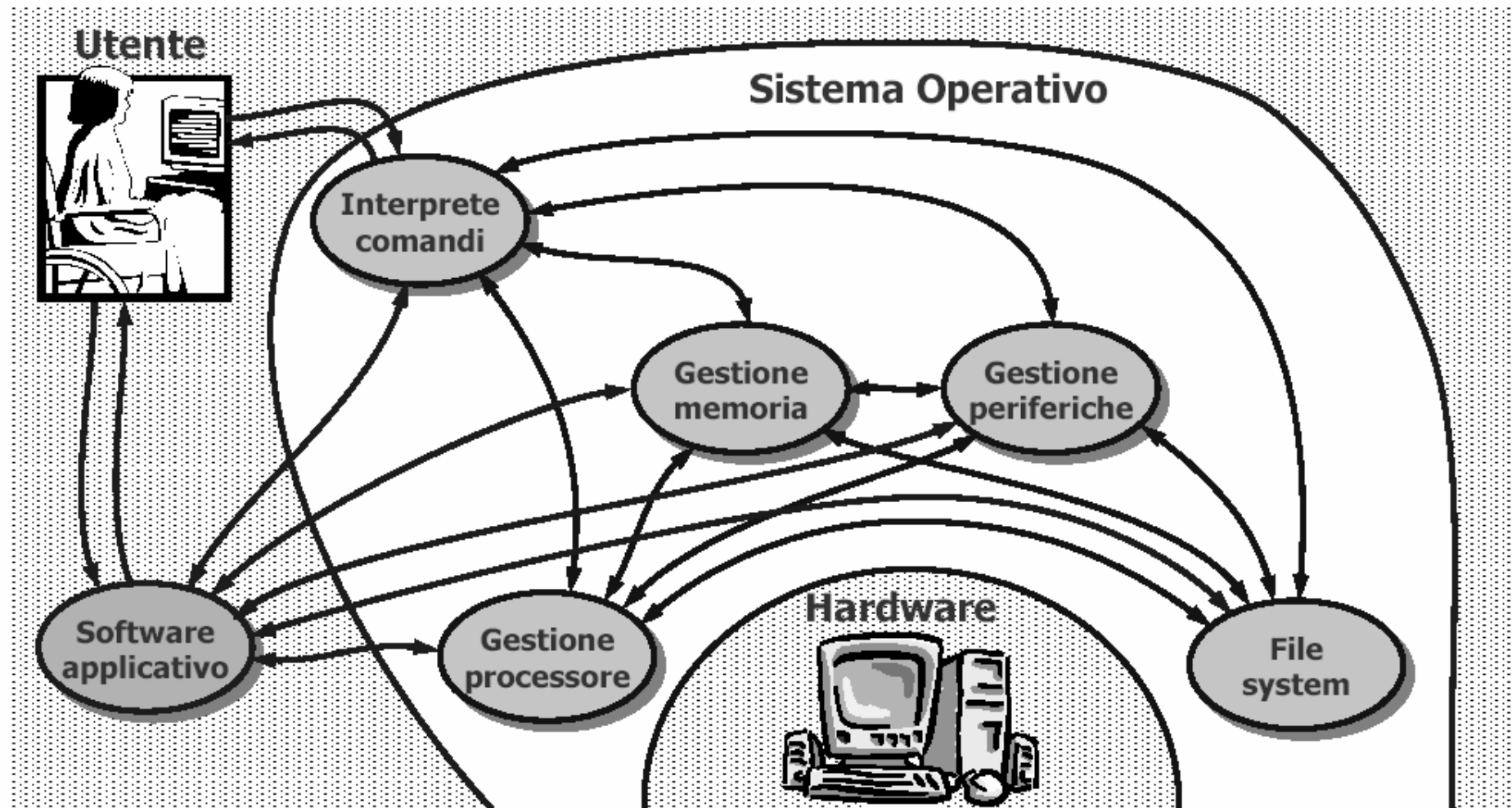
- **Consente l'archiviazione e il reperimento dei dati sfruttando le periferiche che costituiscono la memoria di massa**

ELEMENTI DI UN SO - III

5. Sistema di gestione degli utenti e dei relativi comandi (interprete comandi)

- **Interfaccia diretta con gli utenti**
- **Permette agli utenti di accedere in maniera semplice e intuitiva alle funzionalita' disponibili**

Elementi di un SO



SO vs applicazioni

➤ Programmi applicativi

- hanno accesso a un insieme ridotto di risorse;
- possono utilizzare solo un sottoinsieme delle istruzioni del processore (esecuzione in **modalità utente**);
- non possono decidere autonomamente quando e come avere accesso alle risorse del sistema (richiedono al sistema operativo l'esecuzione di alcuni servizi);
- ...

➤ Sistema operativo

- ha accesso a tutte le risorse;
- può utilizzare tutte le istruzioni del processore (esecuzione in **modalità supervisore**);
- stabilisce in che ordine e come le richieste che riceve devono essere soddisfatte;
- ...

EVOLUZIONE DEI SO - I

1. Seconda meta' degli anni '40

- non c'era un sistema operativo; il programmatore interagiva direttamente con i dispositivi fisici**
- introdurre nel software applicativo anche le funzionalita' di gestione dell'hardware**
- esecuzione di un solo programma alla volta**
- problemi da risolvere: evoluzione dell'interfaccia utente; miglioramento dell'efficienza di gestione delle risorse disponibili**

EVOLUZIONE DEI SO - II

2. Sistemi a lotti (batch)

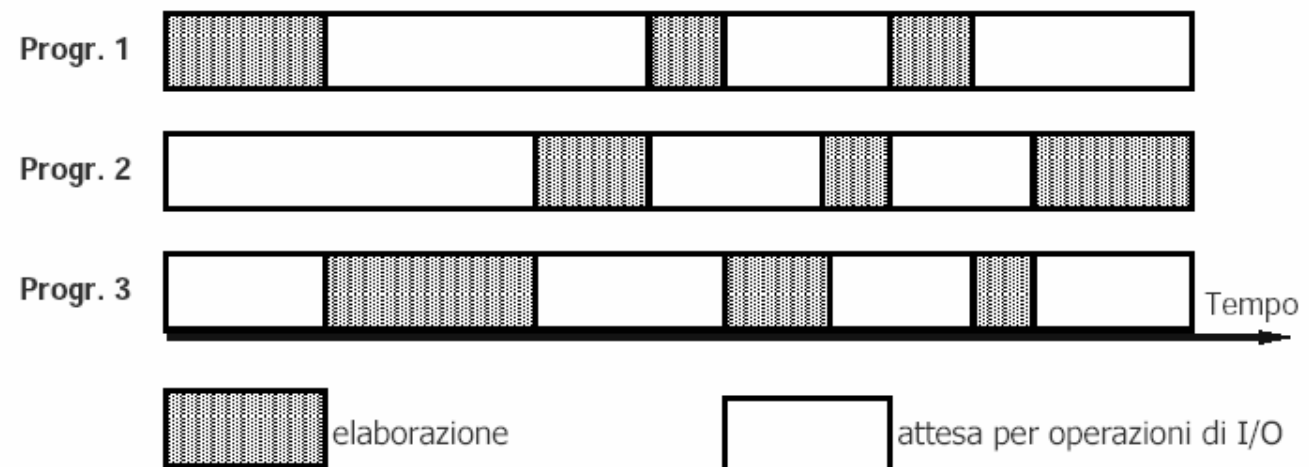
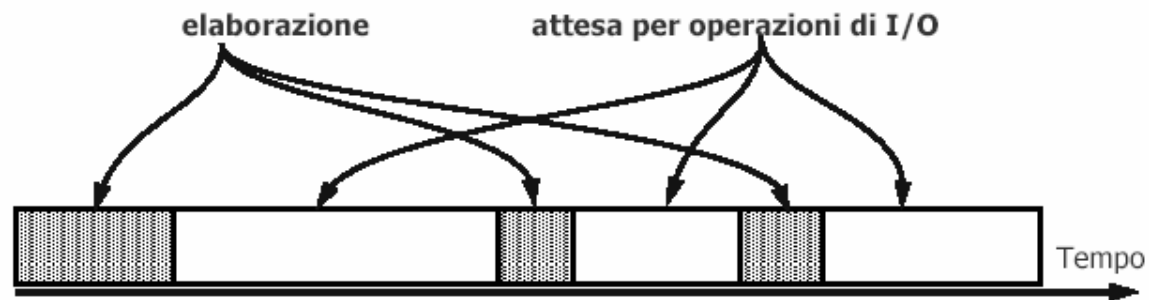
- **impiego di un componente software (monitor) per automatizzare avviamento ed esecuzione del programma**
- **apposito linguaggio (job control language)**
- **le operazioni di i/o vengono eseguite tramite file**
- **ottimizzazione delle risorse di elaborazione tramite distribuzione del tempo di calcolo tra i programmi**
- **svantaggio: tempo di latenza tra il momento in cui il job e' sottoposto al sistema e il momento in cui vengono presentati i risultati dell'elaborazione**

EVOLUZIONE DEI SO - III

3. Sistemi multiprogrammati (la memoria centrale contiene piu' programmi contemporaneamente)

- interattività con utente in tempi di risposta accettabili**
- protezione della memoria**
- temporizzazione dell'esecuzione (nessun programma deve monopolizzare il sistema)**
- gestione delle attività di i/o**
- sfruttamento (anche in parallelo) delle risorse disponibili del sistema**

Multiprogrammazione



MULTIPROGRAMMAZIONE - I

- **nel sistema sono presenti diversi programmi, ognuno con il proprio tempo di elaborazione e propri tempi di attesa per le operazioni di i/o**
- **per evitare che la CPU venga utilizzata in modo esclusivo da parte di un solo programma, il tempo viene suddiviso in unita' elementari, dette quanti, da assegnare, secondo opportune politiche, a tutti i programmi**
- **Round-robin: assegnare a rotazione la disponibilita' di un quanto di tempo della CPU ai vari programmi presenti contemporaneamente in memoria**

MULTIPROGRAMMAZIONE - II

- **Quanto di tempo troppo lungo: il parallelismo tende a scomparire e la sequenzializzazione dei programmi viene degradata**
- **Quanto di tempo troppo corto: se diventa dello stesso ordine del tempo di commutazione fra programmi (circa 100 ms per Unix) il sistema perde efficienza**

Processo vs programma

- **Programma:**
entità statica composta dal codice eseguibile dal processore.
- **Processo:**
entità dinamica che corrisponde al programma in esecuzione, composto da:
 - codice (il programma);
 - dati (quelli che servono per l'esecuzione del programma);
 - stato (a che punto dell'esecuzione ci si trova, cosa c'è nei registri, ...).

GESTIONE DEI PROCESSI: elaborazione parallela

- Il concetto di **elaborazione parallela** si riferisce specificamente:
 - **ai dati**, che potrebbero essere trattati parallelamente;
 - **alle istruzioni**, che potrebbero essere eseguite contemporaneamente nel caso di operazioni indipendenti che trattano dati distinti;
 - **ai programmi**, che potrebbero essere in esecuzione nello stesso tempo.
- Il parallelismo a livello di dati e di istruzioni è possibile solo con l'impiego di architetture di elaborazione parallela, basate sulla presenza di più unità di elaborazione in grado di eseguire istruzioni in modo concorrente ma anche, per esempio, di adeguati linguaggi di programmazione.
- Il parallelismo a livello di programma ricade nell'ambito dei sistemi operativi.
- Le condizioni che un sistema operativo deve soddisfare sono:
 - **efficienza**: occorre assicurare un impiego ottimale di tutte le risorse;
 - **interattività**: il tempo di risposta è un fattore determinante per definire la qualità del sistema operativo;
 - **sincronizzazione/cooperazione**: gestire la conseguente presenza di più **agenti** in grado di operare sul sistema in modo concorrente, al fine di evitare possibili malfunzionamenti (per esempio il blocco di un programma a causa di operazioni errate compiute da altri).

Processi non in esecuzione

- Processi in attesa di un evento esterno (ad esempio I/O)
- Processi pronti ad essere eseguiti in attesa della CPU
- Si tratta di due stati diversi: PRONTO e ATTESA realizzati con due code diverse

SO in time sharing

- Permette la condivisione della CPU tra più processi interattivi
- Il tempo di esecuzione del processore è condiviso tra più utenti
- Ogni processo in esecuzione ha a disposizione un quanto di tempo di utilizzo della CPU, al termine del quale viene sospeso per lasciare il posto ad un altro processo in attesa di esecuzione

Esecuzione di un processo



- Un processo utente può effettivamente essere in esecuzione sulla CPU
- Ogni operazione di I/O consiste in una chiamata al sistema operativo e quindi in una sospensione del processo utente per l'esecuzione dell'operazione di I/O da parte del kernel

Stati di un processo

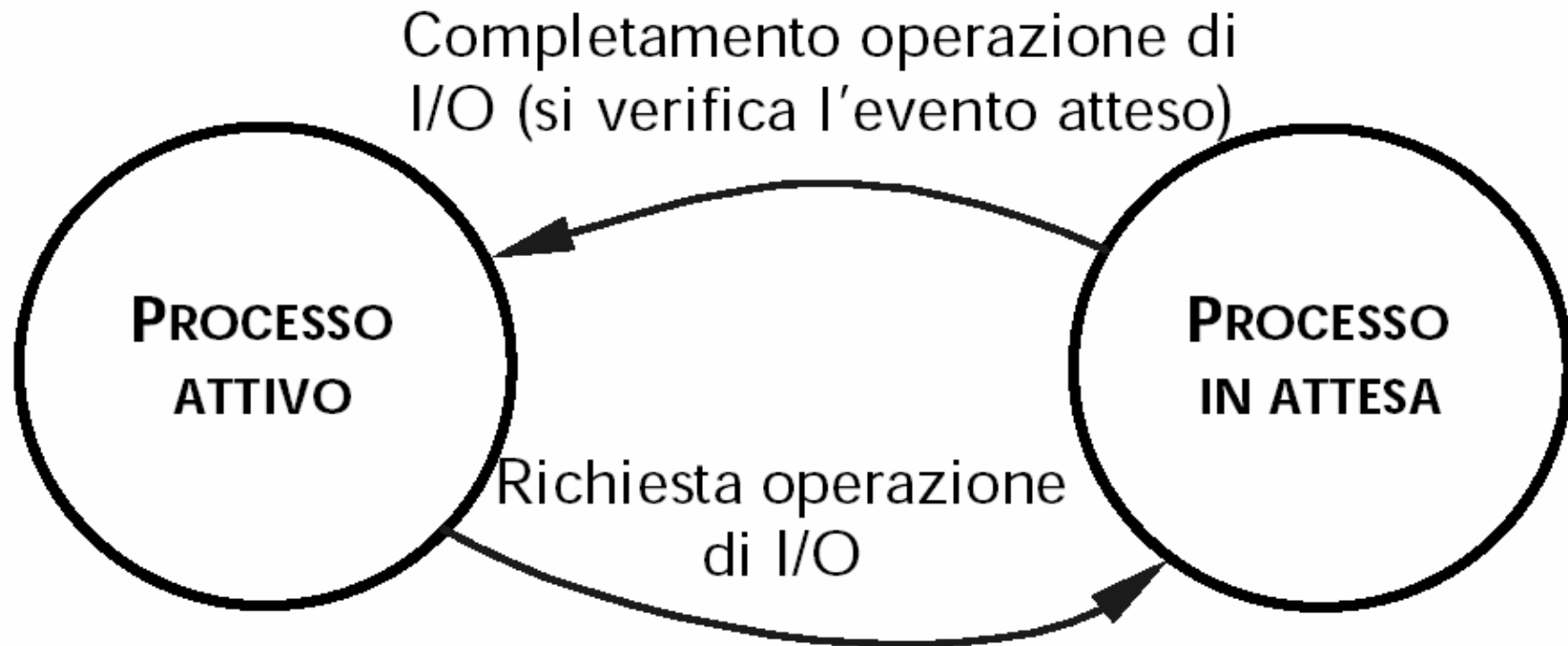
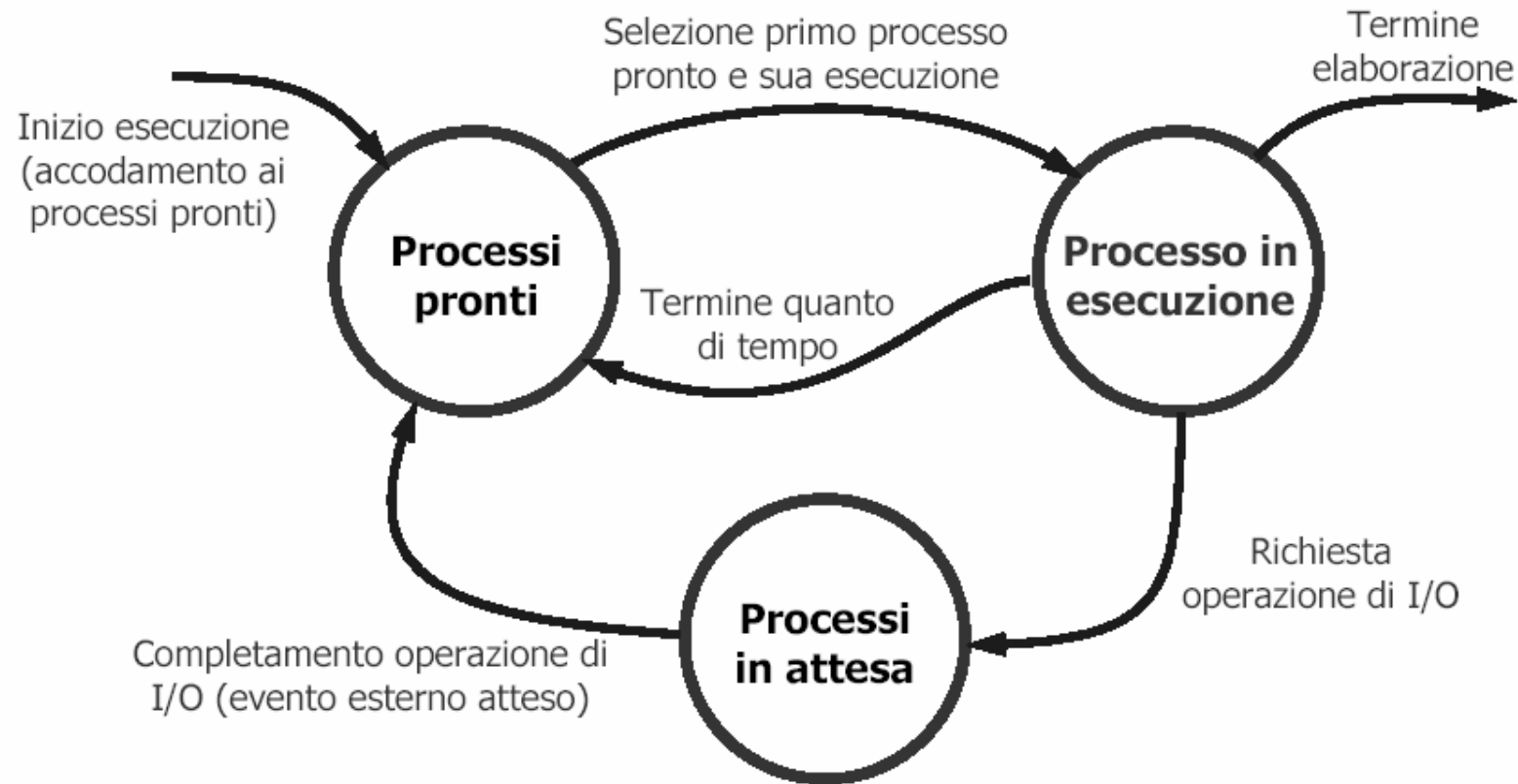


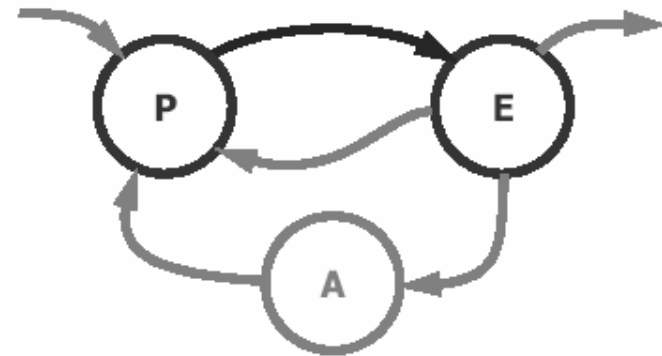
Diagramma a tre stati



Transizioni di stato

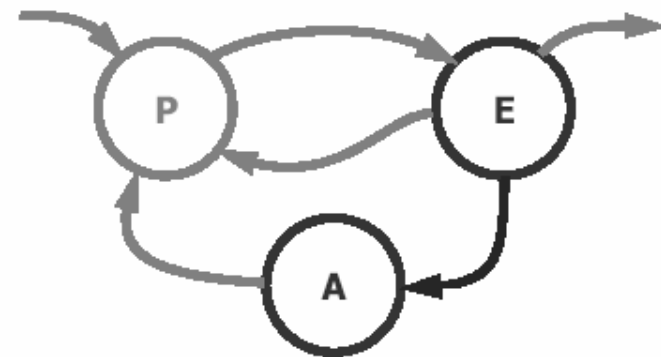
➤ Pronto → Esecuzione

- Il SO stabilisce quale dei processi "pronti" debba essere mandato in "esecuzione".
- La scelta è fatta dall'algoritmo di scheduling che deve bilanciare **efficienza** e **fairness**.



➤ Esecuzione → Attesa

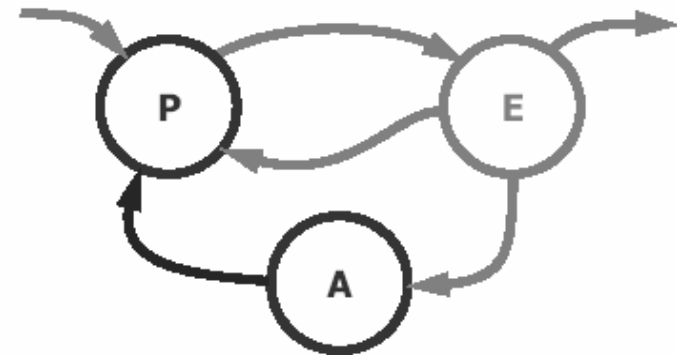
- il processo chiede delle risorse che non sono disponibili o attende un evento
- il SO salva tutte le informazioni necessarie a riprendere l'esecuzione e l'informazione relativa all'evento atteso nella tabella dei processi



Transizioni di stato

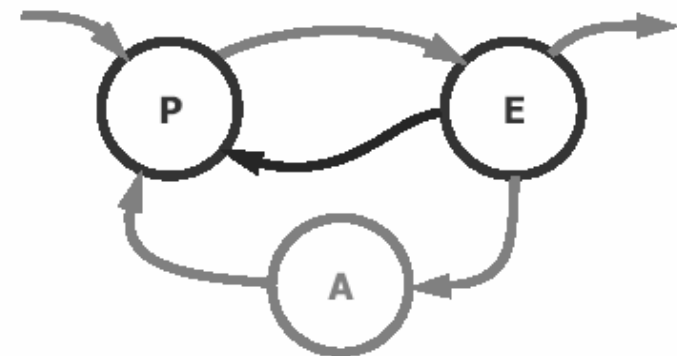
➤ **Attesa → Pronto**

- Si verifica l'evento atteso dal processo e il SO sposta quel processo nella coda dei processi pronti.



➤ **Esecuzione → Pronto**

- Termina il quanto di tempo e il processo in "esecuzione" lascia spazio a un altro processo "pronto".
- Il SO salva (nella tabella dei processi) tutte le informazioni per riprendere l'esecuzione del processo dal punto in cui viene interrotta.
- Contemporaneamente un altro processo passa da "pronto" a "esecuzione".



Una possibile evoluzione

1. Il processo viene creato e viene posto nella coda dei processi pronti;
 2. il primo processo tra i processi pronti viene posto in esecuzione;
 3. il processo in esecuzione dispone delle risorse del sistema fino a
 - a. il termine del quanto di tempo
 - il nucleo interrompe il processo e lo mette in coda ai processi pronti;
 - quando arriva in cima alla coda dei processi pronti, il processo torna in stato in esecuzione, proseguendo con l'elaborazione dell'istruzione successiva a quella su cui era stato interrotto;
 - b. la richiesta di un'operazione di ingresso/uscita
 - il nucleo sposta il processo attivo nello stato di attesa;
 - quando l'operazione di ingresso/uscita si completa il processo può proseguire l'elaborazione e viene messo in fondo alla coda dei processi in pronti e prosegue come nel punto precedente;
 - c. il termine della propria esecuzione (istruzione finale)
 - il processo viene eliminato e rimosso dall'elenco dei processi esistenti;
- in ogni caso il nucleo provvede a sostituirlo con il primo dei processi pronti.

RIFERIMENTI AL LIBRO

- Sistema operativo: Cap. 6 (introduzione);
paragrafi 6.1, 6.1.1, 6.1.2, 6.1.3
- Processi: paragrafi 6.2, 6.2.1