

# Informatica di base e Laboratorio

a.a. 2022/2023

# Sommario della Lezione

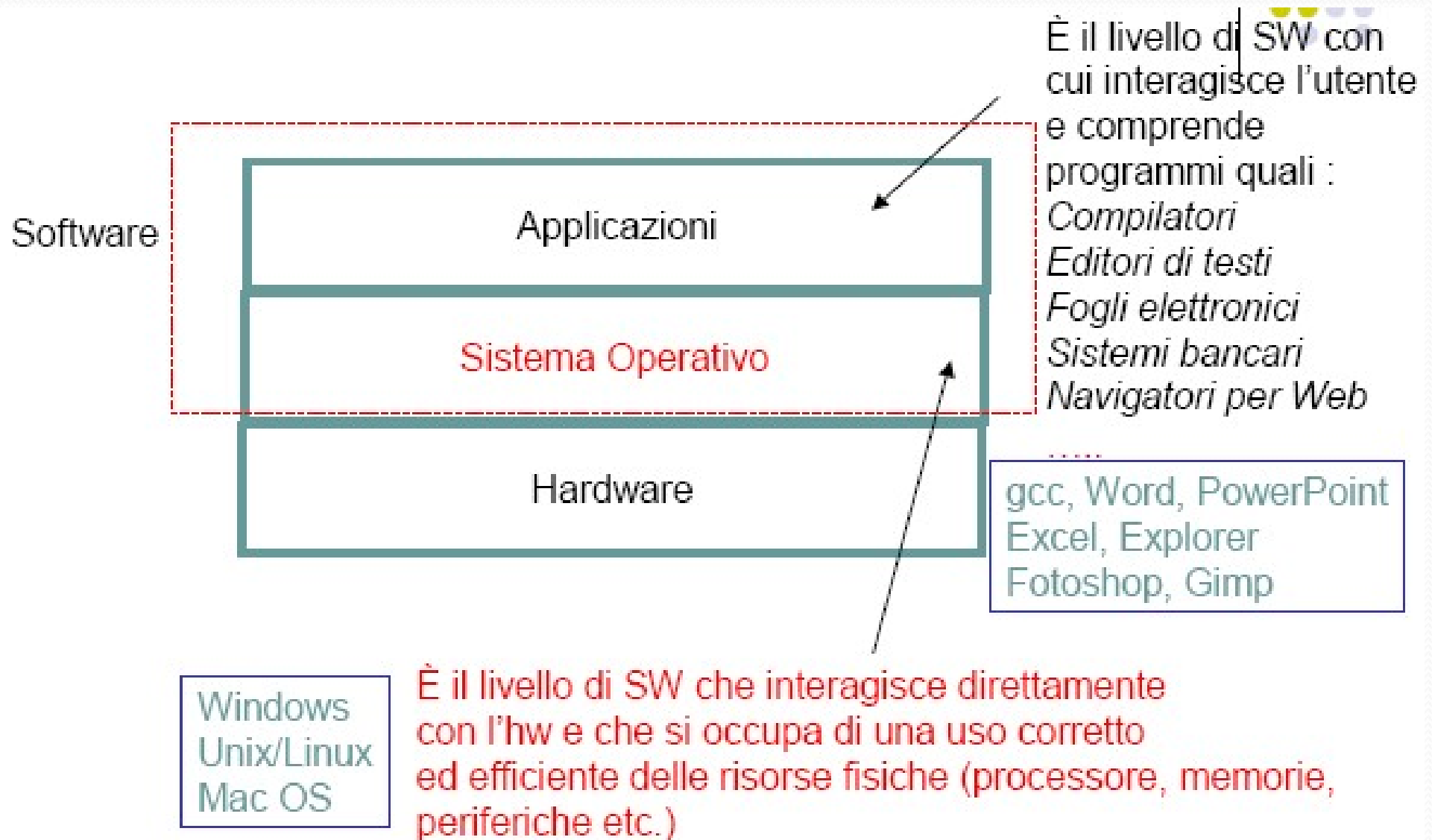
---

- **Il Sistema Operativo:**
  - **Gestore del Processore e dei Processi**
  - **Gestore della Memoria**
  - **Gestore delle Periferiche**
  - **File System**

# Il Sistema Operativo

- Il sistema operativo è la componente software fondamentale di un computer.
- È formato da un insieme di programmi che interagiscono e cooperano tra di loro per realizzare due obiettivi fondamentali:
  - gestire efficientemente l'elaboratore e le sue periferiche, cercando di sfruttare al massimo tutte le componenti hardware
  - creare un ambiente virtuale per facilitare l'interazione uomo-macchina

# Il Sistema Operativo



# Quali sono le funzioni di un SO?

- Esegue applicazioni :
  - carica il programma binario prodotto della compilazione (e residente su disco) nella RAM,
  - cede il processore all'applicazione da eseguire
- Facilita l'accesso ai dispositivi di I/O
  - interagisce con le periferiche facendosi carico di tutti i dettagli fisici (es. modem, reti, video...)
  - mette a disposizione operazioni di lettura/scrittura ad alto livello che possono essere usate senza conoscere i dettagli tecnici della periferica

# Quali sono le funzioni di un SO?

- Archivia dati e programmi :
  - mette a disposizione dell'utente una visione astratta della memoria secondaria (il file system basato sulle astrazioni : *file/archivi e folder/cartelle*)
  - gestisce la realizzazione di queste astrazioni sul supporto fisico (disco) gestendo tutti i dettagli legati alla lettura/scrittura dei settori

# Quali sono le funzioni di un SO?

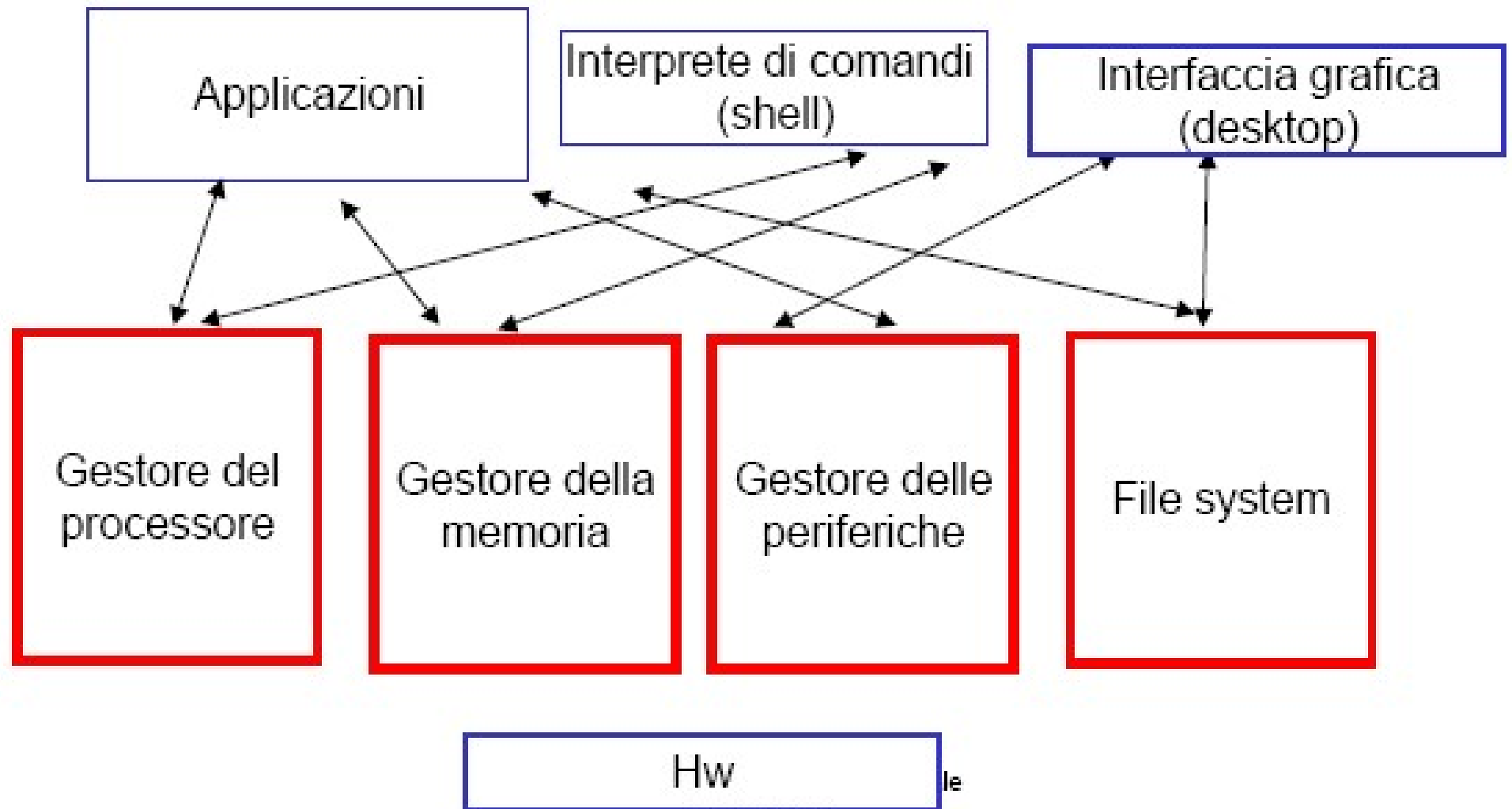
- Gestisce le risorse
  - ripartisce le risorse disponibili fra le varie applicazioni (processore, RAM, periferiche)
  - evita che ci siano malfunzionamenti dovuti all'uso contemporaneo di risorse
    - es: un word processor e un web browser che inviano contemporaneamente dati alla stampante provocano una stampa erronea
  - ottimizza le prestazioni scegliendo delle politiche che permettano di sfruttare al meglio tutte le parti del computer

# Quali sono le funzioni di un SO?

- Gestisce malfunzionamenti del sistema
  - rileva e gestisce situazioni anomale
    - es: se il disco ha un settore difettoso, il SO può trasportare automaticamente le informazioni residenti su quel settore da un'altra parte
    - es: se un'applicazione cerca di effettuare una operazione non permessa (come leggere i dati di un'altra applicazione) può bloccare l'applicazione segnalando all'utente la situazione erranea
- ... i SO sono sistemi complessi che non sempre riescono a fare correttamente tutto quello che dovrebbero ....



# Quali sono le parti di un SO?



# Gestione del processore e dei processi

- Il processore è la componente più importante di un sistema di elaborazione e pertanto la sua corretta ed efficiente gestione è uno dei compiti principali di un sistema operativo
- Il ruolo del processore è quello di eseguire programmi
- Consideriamo la seguente definizione di **processo**:

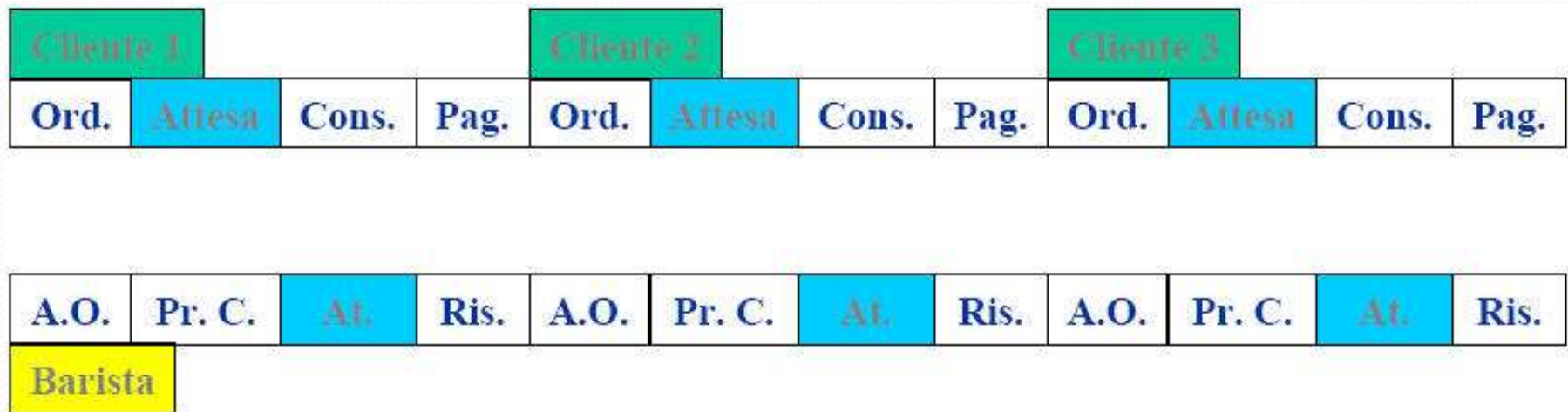
**un processo è un programma in esecuzione**

# Gestione del processore e dei processi

- Uno dei compiti principali del sistema operativo riguarda la gestione dei processi
- Per il momento assumiamo di trattare sistemi monoprogrammati (anche detti **mono-tasking**)
- In sistemi di questo tipo è possibile eseguire un solo programma alla volta: i programmi devono essere eseguiti in modo sequenziale e si può mandare in esecuzione un programma solo quando quello precedente ha terminato l'esecuzione (es. MS/DOS)

# Gestione del processore e dei processi

- Supponiamo che il nostro sistema sia un bar in cui un barista serve diversi clienti
  - Il barista è il corrispondente del processore, i clienti sono l'equivalente dei processi da eseguire
- Esecuzione mono-tasking:



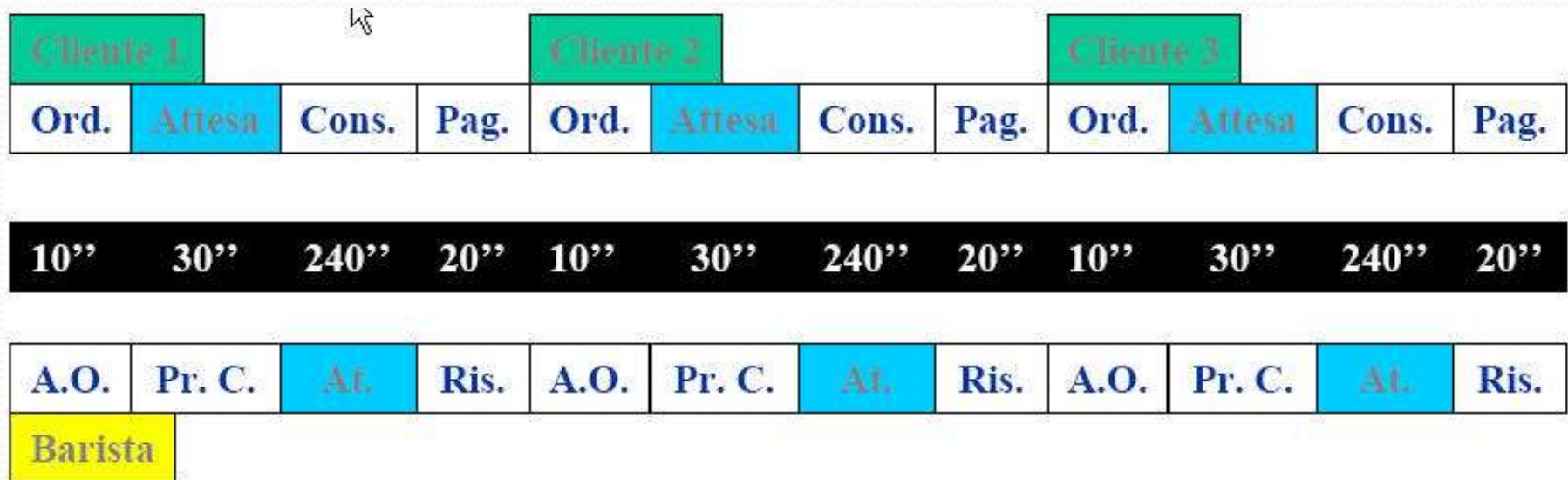
# Gestione del processore e dei processi

- Quello che viene evidenziato dall'esempio precedente è vero in generale:

*qualunque processo  $P$  alterna fasi di esecuzione a fasi in cui è bloccato in attesa di qualche evento esterno*

- Un processo può essere in attesa che sia terminata un'operazione di input di dati oppure in attesa di poter usare una risorsa in quel momento occupata

# Gestione del processore e dei processi



- Il tempo impiegato per servire un cliente è pari a 300''
- Durante questo tempo il barista attende (senza fare altro) per un tempo pari a 240''
- Il tempo di attesa (non produttivo) è 80% del tempo totale

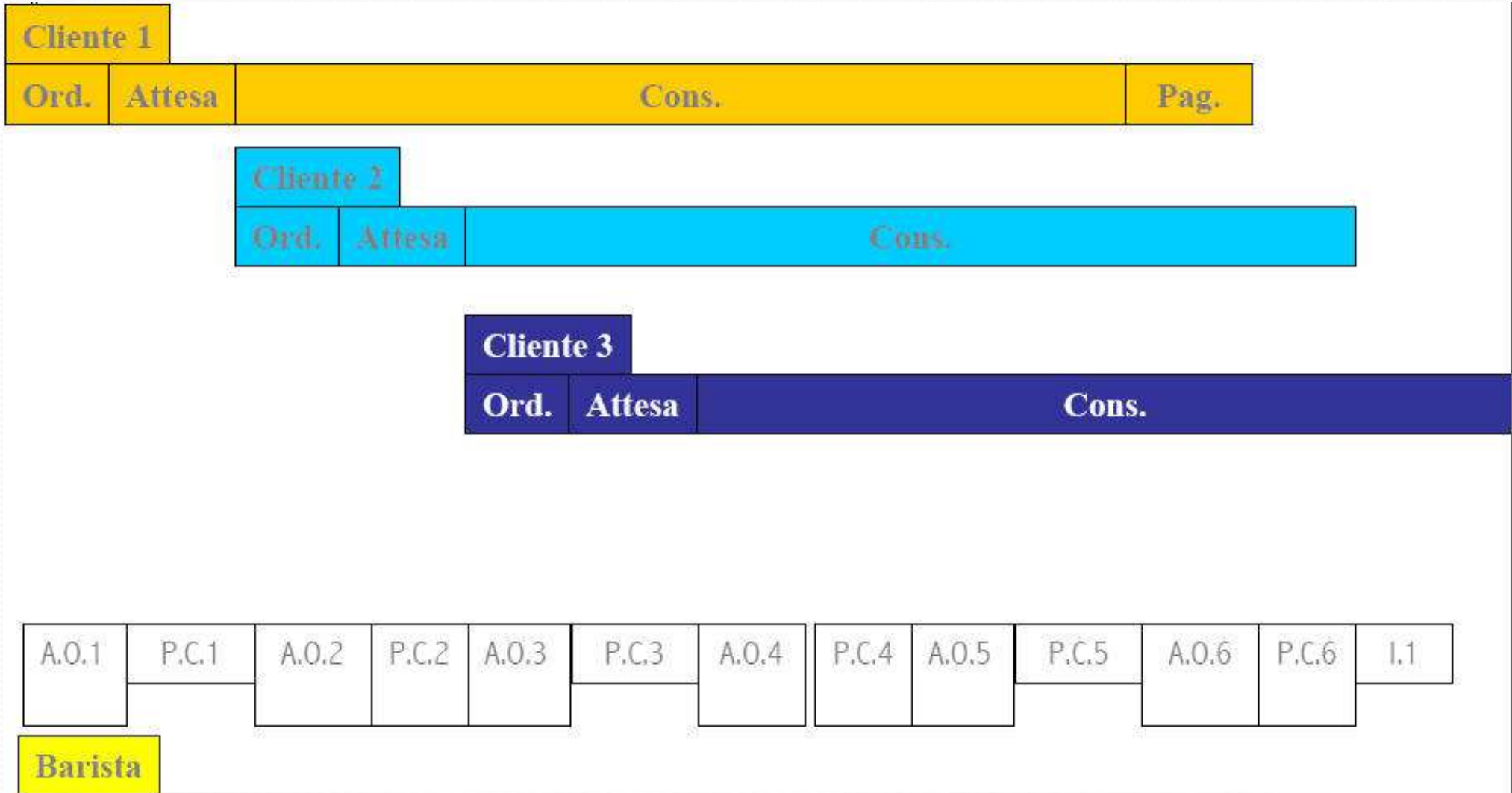
# Gestione del processore e dei processi

- Una soluzione ragionevole per migliorare sia la “produttività” del barista che la qualità del servizio (i clienti arrivano e si mettono in coda aspettando di essere serviti) potrebbe essere quella di servire i clienti:

## *contemporaneamente*

- Ogni volta che un cliente inizia la consumazione del suo caffè, il barista, invece di restare inattivo può iniziare il servizio del prossimo cliente

# Gestione del processore e dei processi





# Gestione del processore e dei processi

- Si noti che il termine

*contemporaneamente*

è sempre scritto in corsivo

- Infatti, poiché vi è un solo barista (un solo processore nel caso di un computer) il servizio non è realmente eseguito in parallelo ma avviene alternando il servizio tra i vari clienti
- In questo modo i tempi di inattività del barista sono eliminati

# Gestione del processore e dei processi

- L'idea illustrata mediante l'esempio precedente è l'idea di base del **multi-tasking**:  
più programmi vengono eseguiti *contemporaneamente* sullo stesso processore. Il numero di processi attivi viene detto **grado di multi-programmazione** del sistema
- Dal punto di vista del processore, in ogni istante vi è un solo processo in esecuzione
- Se l'alternanza tra i processi è frequente (ad esempio ogni 10 millisecondi), l'utente ha l'impressione che l'esecuzione dei programmi sia veramente simultanea

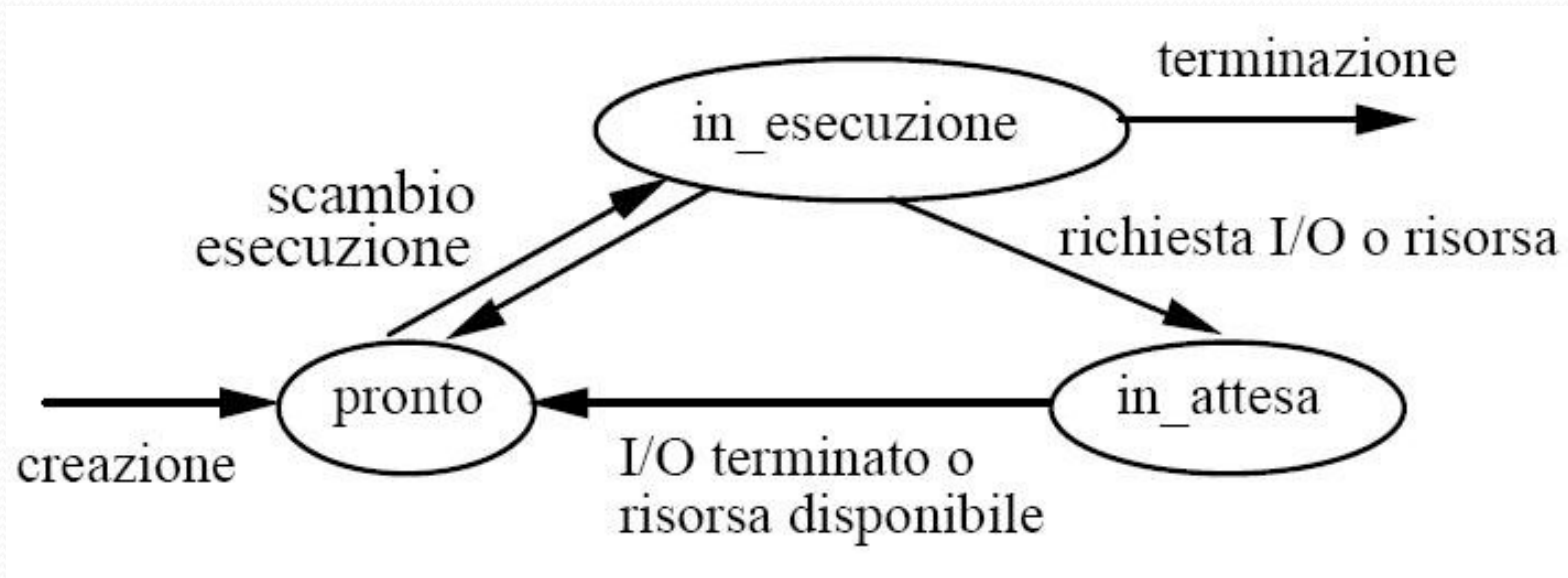
# Gestione del processore e dei processi

- A livello macroscopico si ha quindi l'impressione della *contemporaneità*, mentre a livello microscopico si ha una semplice alternanza sequenziale molto veloce
- Si noti che dal punto di vista di ogni singolo programma il tempo di esecuzione, cioè il tempo che intercorre tra l'inizio e la fine del processo, risulta aumentato rispetto al caso monoprogrammato a causa dell'alternanza con gli altri processi

# Gestione del processore e dei processi

- Un processo può trovarsi in tre diversi **stati**:
  - in **esecuzione**, quando sta utilizzando il processore;
  - in **attesa** (bloccato), quando è in attesa del verificarsi di un evento esterno (terminazione di un'operazione di input/output o altro)
  - **pronto**, quando è potenzialmente in condizione di poter utilizzare il processore che è occupato da un altro processo

# Gestione del processore e dei processi



# Gestione del processore e dei processi

- Quando un processo viene creato (alla richiesta da parte di un utente di eseguire un programma) viene messo nello stato di *pronto* e in tale stato rimane fino a quando non arriverà il suo turno
- Non appena un processo *pronto* viene selezionato, passa nello stato di *esecuzione*
- Un processo può abbandonare lo stato di *esecuzione* per tre diverse ragioni

# Gestione del processore e dei processi

- **Terminazione:** il processo termina la sua esecuzione e abbandona il sistema
- Richiesta di un'operazione di input/output o di una **risorsa** occupata. In questo caso il processo passa allo stato di *attesa*, il processore viene liberato e può essere concesso ad un altro processo *pronto*
- **Cambio di esecuzione:** per realizzare in modo equo l'alternanza tra i vari processi, in certi casi può essere opportuno fermare un processo, rimetterlo nello stato di *pronto* e concedere il processore ad un altro processo

# Gestione del processore e dei processi

- Il multi-tasking funziona efficacemente se il S.O. è in grado di offrire alcune assicurazioni
- Il cambio di contesto deve essere invisibile al processo. Nell'esempio del bar i clienti non si accorgono che il barista serve un gruppo di clienti contemporaneamente
- Quando il barista riprende a servire un cliente **il servizio viene ripreso esattamente nel punto in cui era stato interrotto**
- Il barista ha un cronometro e ogni  $n$  secondi sospende il servizio che sta effettuando e passa a servire un altro cliente

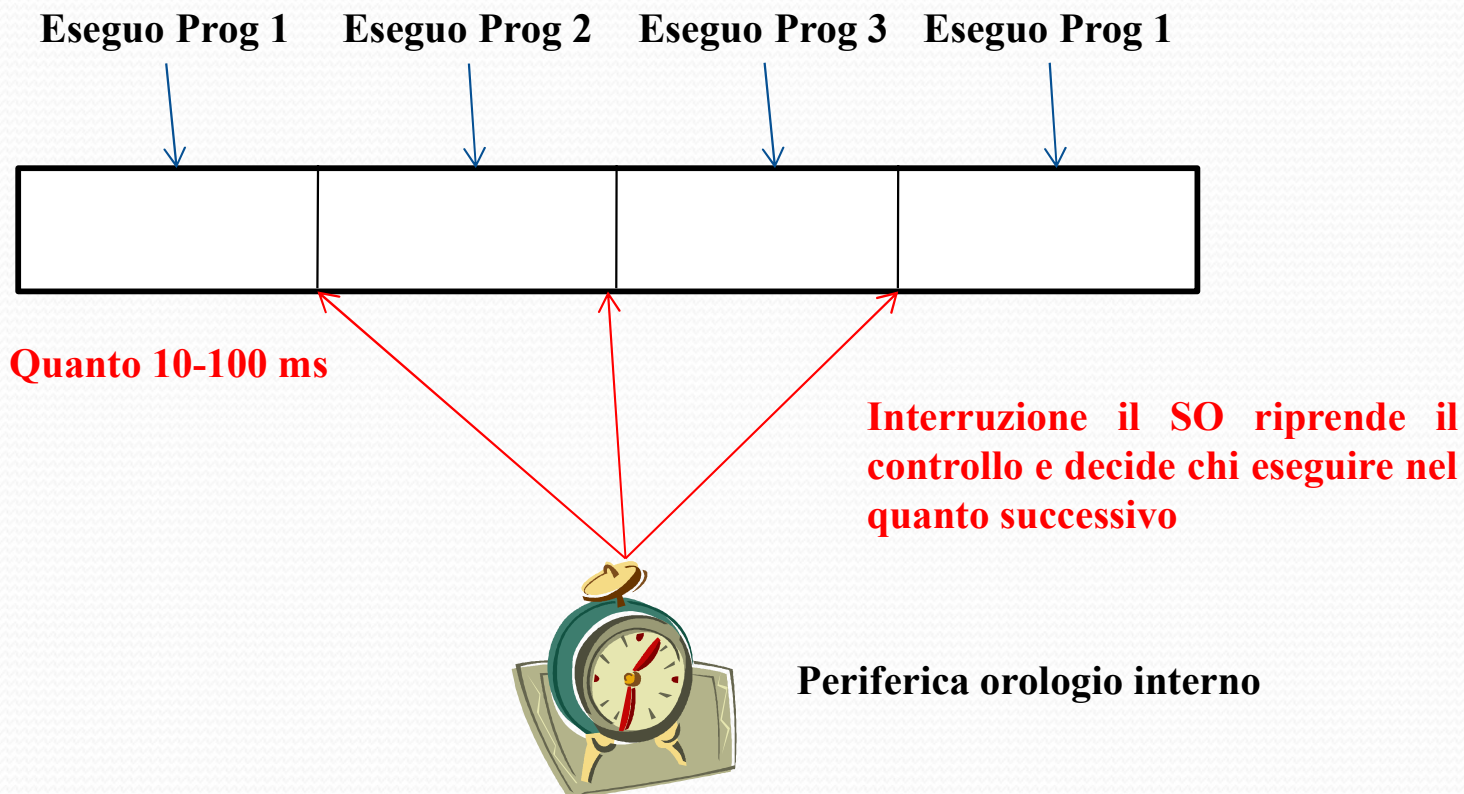


# Gestione del processore e dei processi

- La politica di gestione in cui il barista divide il suo tempo tra i vari clienti assegnando a ciascuno un certo intervallo di tempo è implementata nei S.O. multi-tasking e prende il nome di **Round-Robin**
- La politica di gestione in cui il barista serve in modo sequenziale i vari clienti viene chiamata **FIFO**

# Gestione del processore e dei processi

- Si usa la tecnica dei *quanti di tempo*
  - es: 3 programmi attivi Prog1, Prog2, Prog3
  - vengono mandati in esecuzione ciclicamente



# Il gestore della memoria

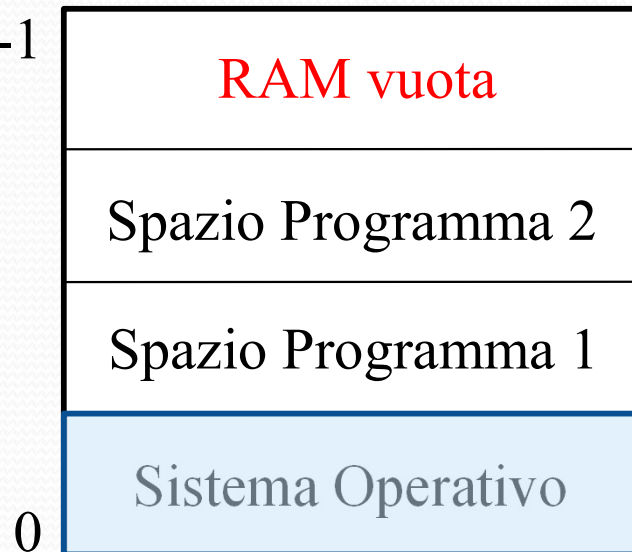
- Tutti i programmi che compongono il SO ed i programmi applicativi attivi usano contemporaneamente la RAM
- Il *gestore della memoria* si preoccupa di fare condividere la RAM ai vari processi in esecuzione in modo che :
  - ogni processo abbia il suo spazio *privato* distinto dagli altri (e inaccessibile agli altri)
  - ogni processo abbia abbastanza memoria per eseguire il proprio algoritmo e raccogliere i suoi dati

# Il gestore della memoria

- La gestione statica
  - ricopiare *interamente* lo spazio di indirizzamento di un programma da memoria secondaria a RAM all'inizio dell'esecuzione

AmpiezzaRAM -1

Una possibile organizzazione della RAM con più programmi attivi contemporaneamente



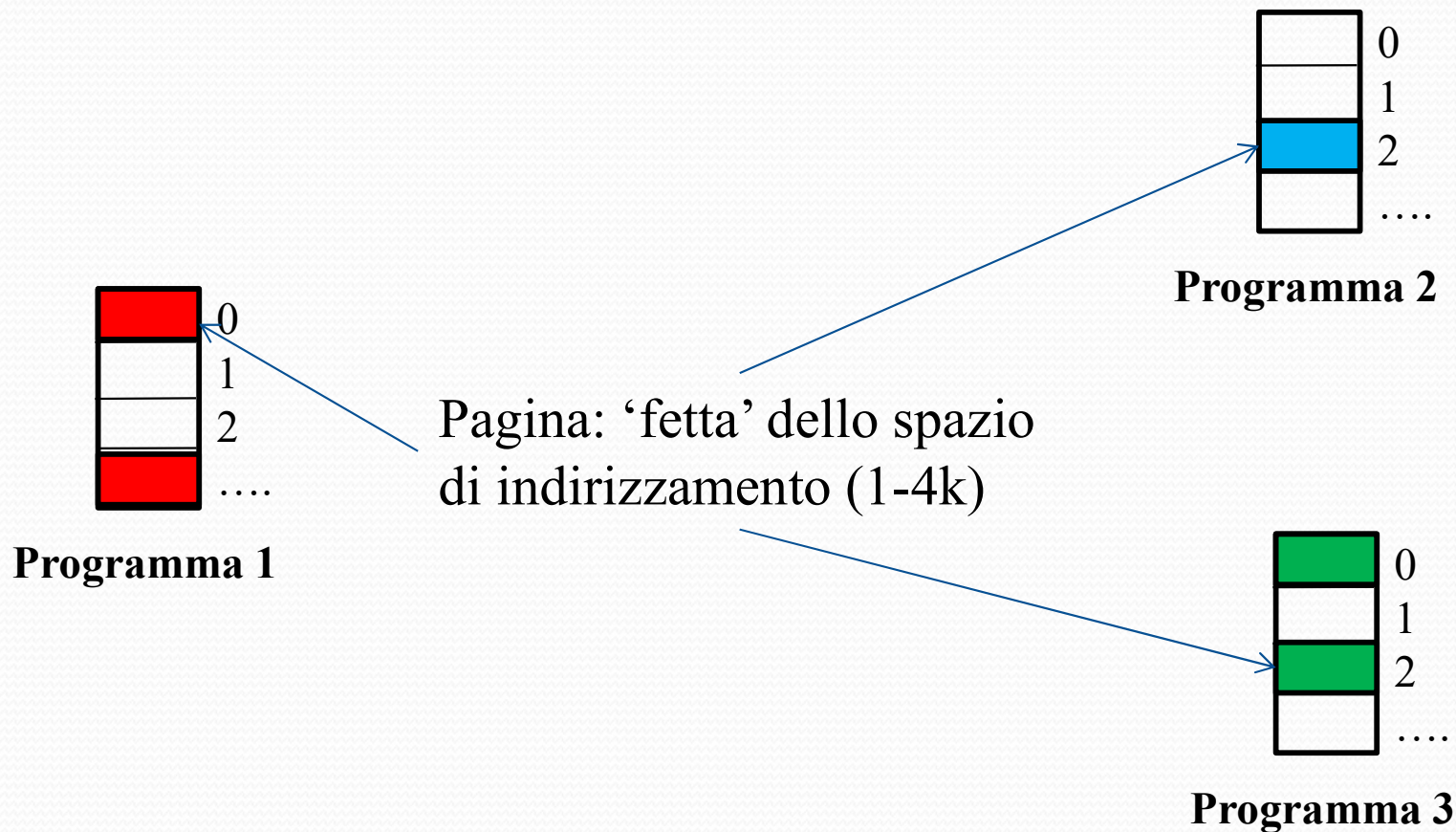
Area riservata, non accessibile in modalità utente.

# Il gestore della memoria

- Problemi della gestione statica
  - non posso girare programmi con spazio di indirizzamento più grande della RAM !
    - Attualmente ogni programma usa almeno 4GB di spazio di indirizzamento, con RAM assai più piccole...
- Soluzione : gestione dinamica (a pagine)
  - ad ogni istante carico in RAM solo i pezzi che mi servono per l'esecuzione corrente

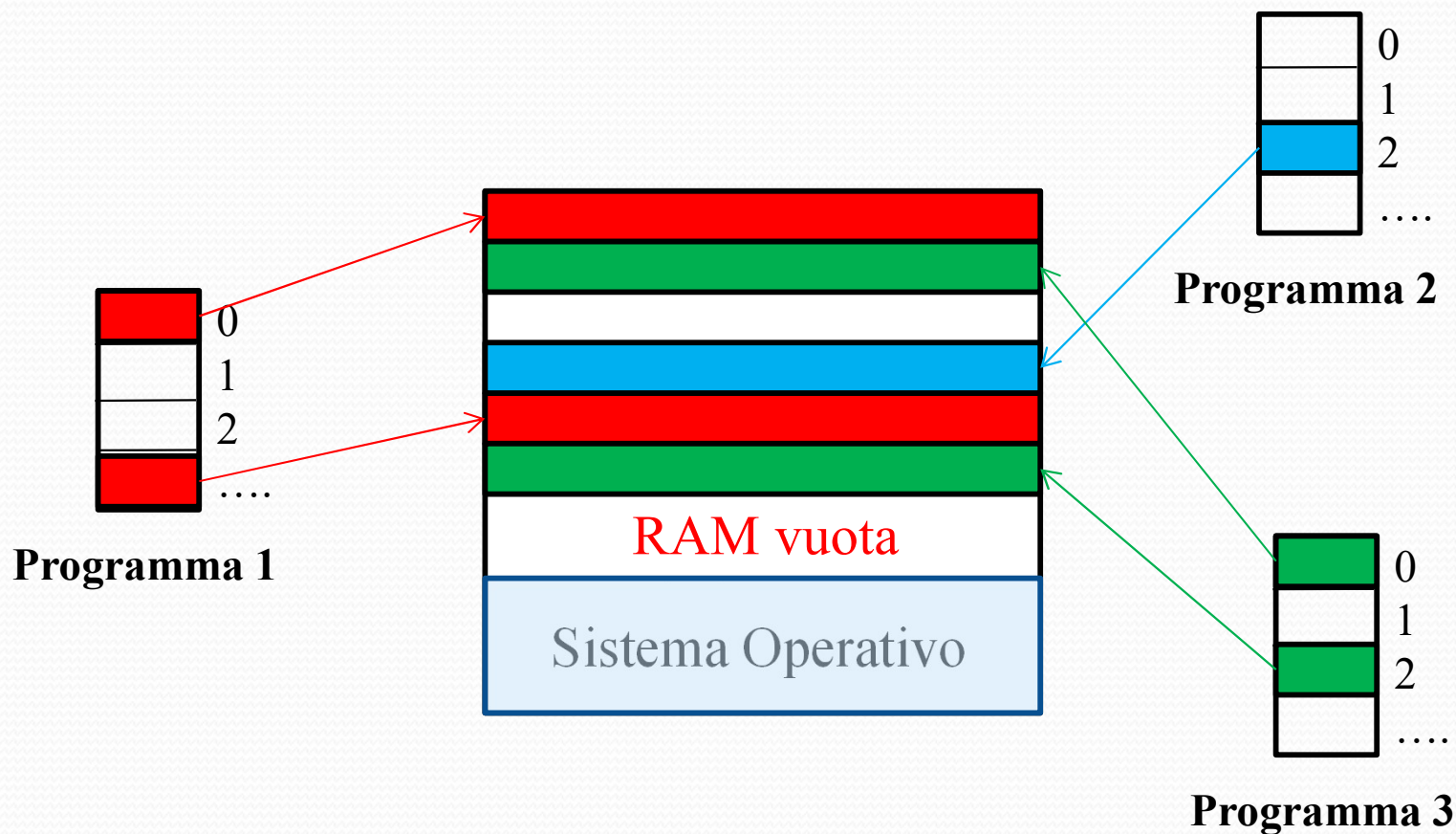
# Gestione dinamica della memoria

- Paginazione
  - lo spazio di indirizzamento di ogni processo è diviso in 'fette' (*pagine*) tutte della stessa ampiezza



# Gestione dinamica della memoria

- Paginazione
  - ad ogni istante solo le pagine necessarie sono caricate in memoria (località!)



# Gestione delle periferiche

- Solitamente il driver non è sviluppato da chi costruisce il SO (es. Microsoft) ma da chi costruisce il dispositivo
- Quando viene acquistata una nuova periferica il driver corrispondente deve essere inserito nel Sistema Operativo (*installazione*)
  - operazione rischiosa, perché se il nuovo driver è difettoso può rovinare il vostro sistema (il driver essendo nel SO ha accesso a tutta la macchina senza restrizioni!....)



# Driver e controller

- Quando si installa un nuovo driver bisogna fornirgli delle informazioni a basso livello per permettergli di colloquiare correttamente con il resto del sistema (*configurazione*)
  - es: stabilire quali linee del bus utilizzare per inviare una interruzione ...
- Periferiche plug&play: quelle per cui la configurazione può essere fatta automaticamente dal sistema
  - sono generalmente tutte così nei PC odierni

# File System

- Esistono diversi tipi di supporti per la memorizzazione permanente delle informazioni: dischi magnetici (floppy disk, hard disk), dischi ottici (cd), nastri magnetici
- Il **File System** è quella parte del S.O. che si occupa di gestire e strutturare le informazioni memorizzate su supporti permanenti
- I file vengono utilizzati come supporto per la memorizzazione dei **programmi** (sia programmi di sistema che programmi utente) e dei **dati**

# File System

- Il sistema operativo deve fornire una visione **astratta** dei file su disco e l'utente deve avere la possibilità di:
  - identificare ogni file con un nome (**filename**) astraendo completamente dalla sua memorizzazione fisica
  - avere un insieme di operazioni per lavorare sui file: creare o rimuovere un file, copiarlo, cambiargli nome, inserire informazioni in un file
  - effettuare l'accesso alle informazioni mediante operazioni ad alto livello, che non tengono conto del tipo di memorizzazione (accedere ad un file memorizzato sul disco rigido oppure su un CD-ROM allo stesso modo)

# File System

- avere la possibilità di strutturare un insieme di file, organizzandoli in sottoinsiemi secondo le loro caratteristiche, per avere una visione ordinata e strutturata delle informazioni sul disco
- in un sistema multi-utente, inoltre l'utente deve avere dei meccanismi per proteggere i propri file, ossia per impedire ad altri di leggerli, scriverli o cancellarli
- i moderni sistemi operativi (es. Windows) forniscono supporto per queste attività

# Operazioni sui file

- Un insieme di operazioni minimale, presente in tutti i sistemi, è il seguente:
  - *creazione* di un file
  - *cancellazione* (rimozione) di un file
  - *copia* di un file
  - *visualizzazione* del contenuto di un file ad esempio sul video del terminale o in una finestra sul video
  - *stampa* di un file
  - *lettura* e *modifica* del contenuto di un file
  - *ridenomina* di un file, per cambiare il nome ad un file
  - *visualizzazione* delle caratteristiche di un insieme di file (dimensione, data di creazione e dell'ultima modifica, vincoli di protezione, ecc.)

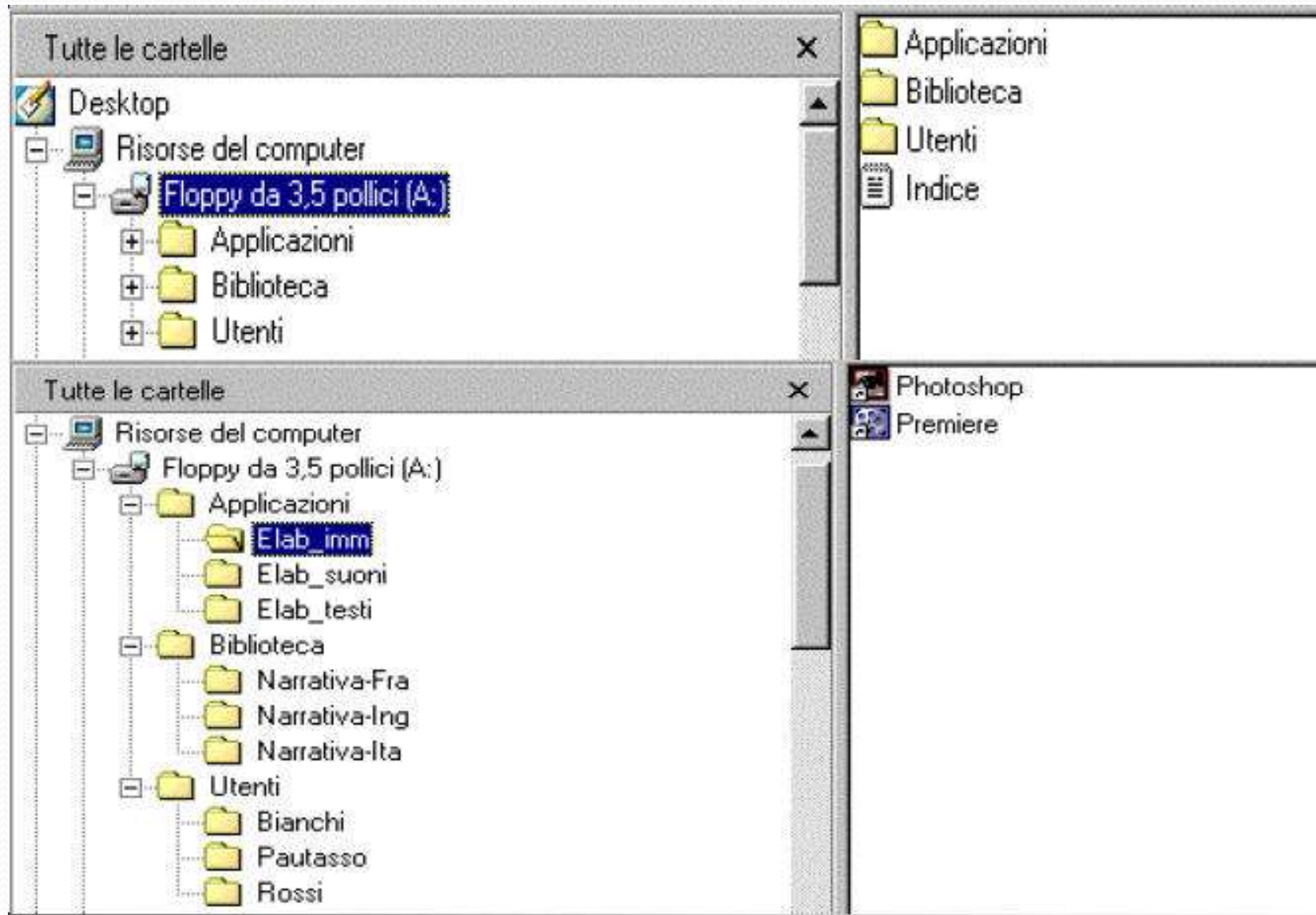
# Organizzazione gerarchica dei file

- Il numero di file che devono essere memorizzati su un disco può essere estremamente elevato
- Si ha quindi la necessità di mantenere i file in una forma ordinata
- Un unico spazio (contenitore) di file è scomodo perché le operazioni di ricerca di un file e di creazione di un nuovo file diventano onerose (non è possibile avere due file con lo stesso nome)
- L'idea è quella di raggruppare i file in sottoinsiemi (seguendo magari dei criteri di tipo logico)

# Organizzazione gerarchica dei file

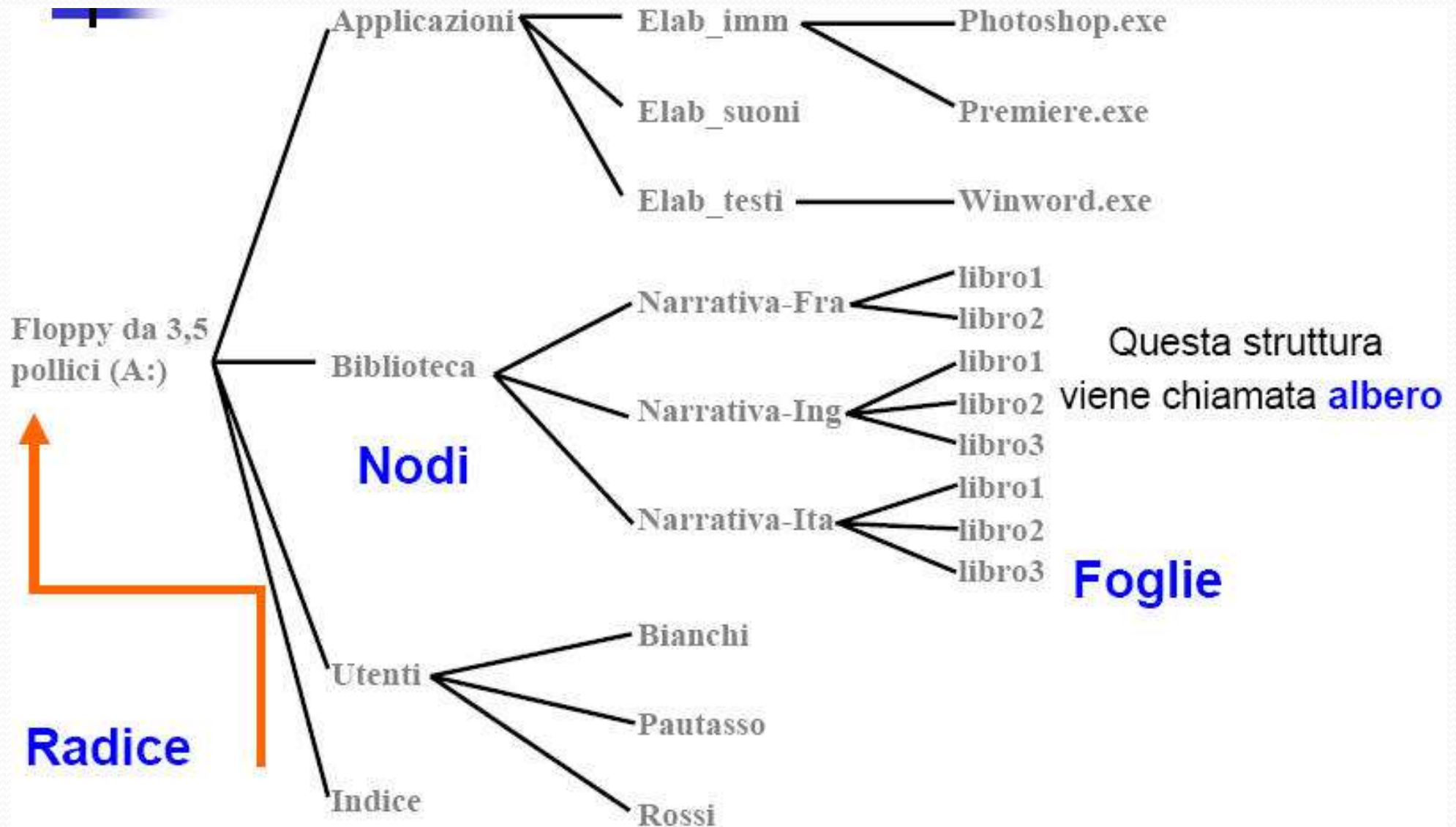
- Tutti i sistemi operativi forniscono operazioni per creare delle **directory**
- Una directory è costituita da un insieme di file
- I nomi dei file sono locali alle directory (si possono avere due file con lo stesso nome purché siano in due directory diverse)

# Organizzazione gerarchica dei file





# Organizzazione gerarchica dei file



# Organizzazione gerarchica dei file

- Se non esiste la strutturazione in directory, tutti i file potrebbero essere identificati mediante il loro nome
- Nel caso di un'organizzazione gerarchica a più livelli il nome non è più sufficiente ad identificare il file (nell'esempio precedente esistono diversi file con lo stesso nome)
- Per individuare un file o una directory in modo univoco si deve allora specificare l'intera sequenza di directory che lo contengono, a partire dalla radice dell'albero

# Organizzazione gerarchica dei file

- Ad esempio il file `libro1` di narrativa italiana è univocamente identificato dalla sequenza:

*A:\Biblioteca\Narrativa-Ita\libro1*

- La directory `Pautasso` di `Utenti` è identificata dalla sequenza:

*A:\Utenti\Pautasso*

*il carattere “\” (backslash) viene usato come separatore.*

- Una sequenza di questo tipo può essere vista come il cammino che si deve compiere per raggiungere un determinato file a partire dalla radice dell'albero, ed è chiamata **pathname**

# Organizzazione gerarchica dei file

- Per organizzare gerarchicamente i file, il sistema operativo deve fornire all'utente un insieme di operazioni sulle directory, per esempio:
  - *creare* una nuova directory
  - *rimuovere* una directory
  - *ridenominare* una directory
  - *elencare il contenuto* (l'insieme di file e sotto-directory) di una directory
  - *copiare* o *spostare* i file da una directory ad un'altra

# Avviamento dell'elaboratore

- In genere il sistema operativo viene mandato in esecuzione al momento dell'accensione della macchina
- Questa fase iniziale prende il nome di **bootstrap**
- In questa fase una parte del S. O. (ossia un insieme di programmi e un insieme di dati) viene caricata in memoria principale

# Avviamento dell'elaboratore

- In genere questa parte del S. O. comprende:
  - i programmi per la gestione dei processi e del processore
  - i programmi per la gestione della memoria
  - i programmi per la gestione delle periferiche e dell'input/output
  - i programmi per la gestione del file system
  - un programma che crea l'interfaccia verso l'utente (che può essere sia di tipo testuale che di tipo grafico)

# Avviamento dell'elaboratore

- Una parte del sistema operativo deve essere sempre mantenuta in memoria principale e deve essere sempre pronta per l'esecuzione
- Questo significa che parte della memoria principale dovrà essere dedicata a mantenere i programmi e i dati riguardanti il sistema operativo
- Durante la fase di bootstrap vengono inoltre effettuate operazioni di inizializzazione del sistema per tener conto delle risorse hardware collegate all'elaboratore

# Avviamento dell'elaboratore

- Spesso durante questa fase sono eseguiti anche dei programmi che verificano l'eventuale presenza di virus annidati sul disco dell'elaboratore
- I virus sono dei programmi pirata che possono essere trasmessi da un elaboratore ad un altro quando si copiano dei programmi
- Un virus può danneggiare il funzionamento dell'elaboratore generando operazioni di disturbo o condizioni di errore (in alcuni casi i virus possono anche portare a danneggiamenti seri)