ARCHITETTURA HARDWARE DEGLI ELABORATORI

ARCHITETTURA HARDWARE DEGLI ELABORATORI

Elementi della macchina di Von Neumann

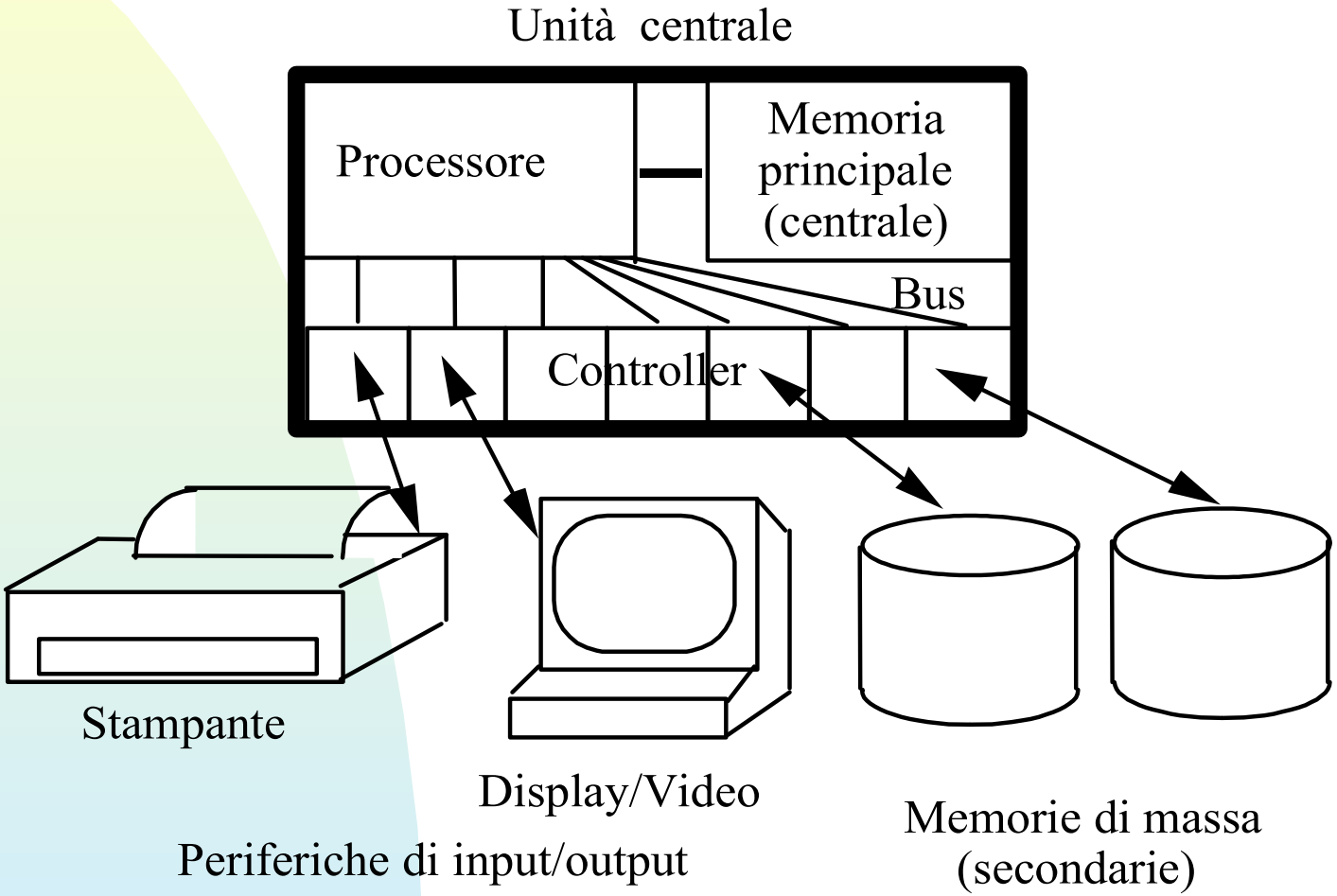
Memoria principale

Processore

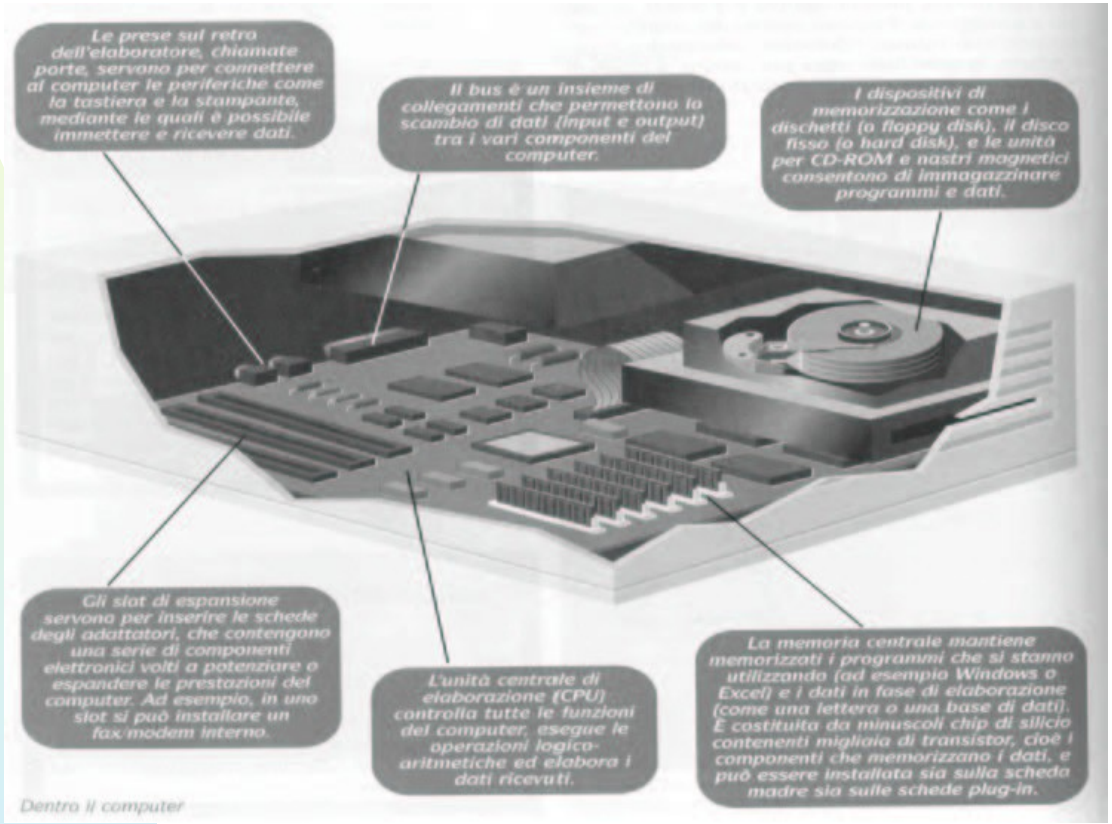
Memoria secondaria

Dispositivi di Input/Output

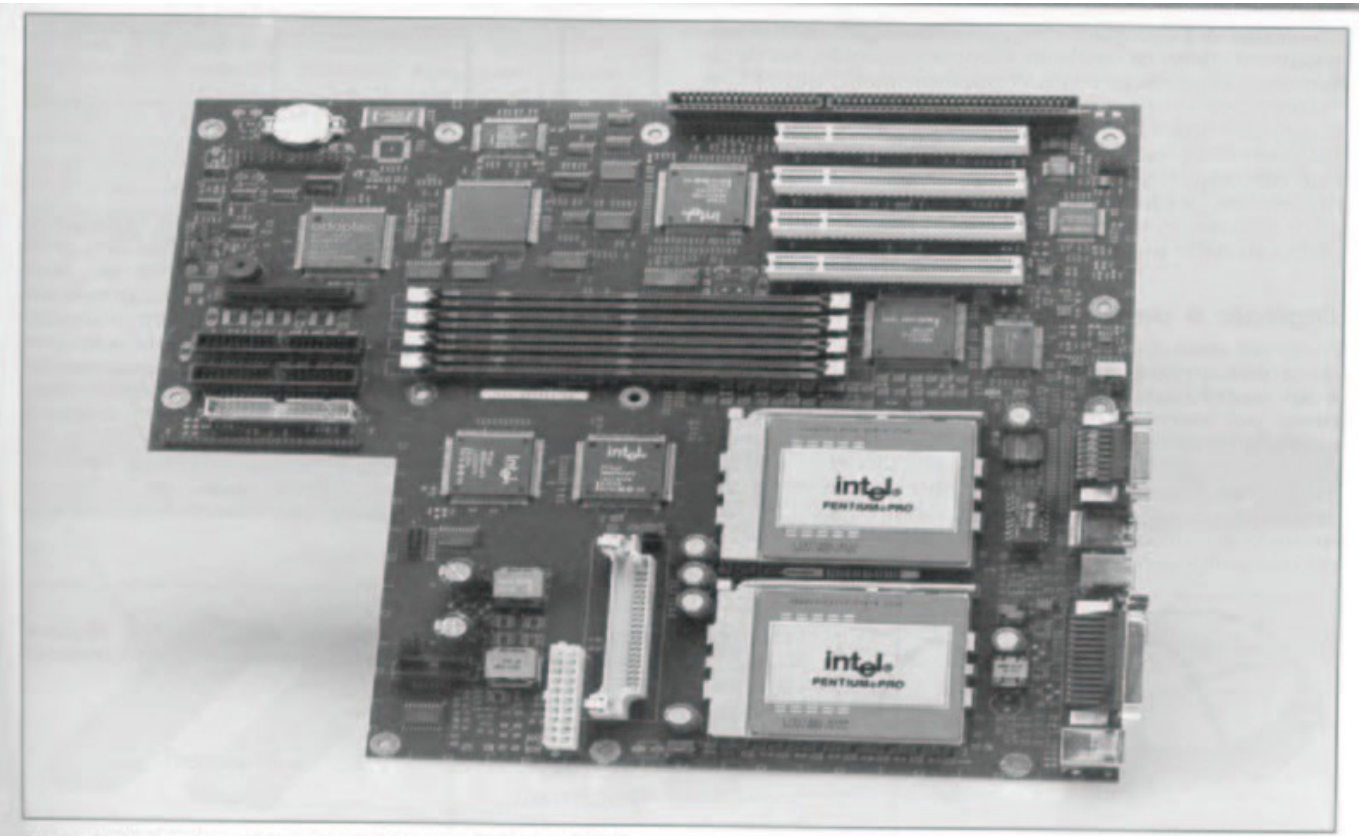
ARCHITETTURA HARDWARE DEGLI ELABORATORI



Struttura interna

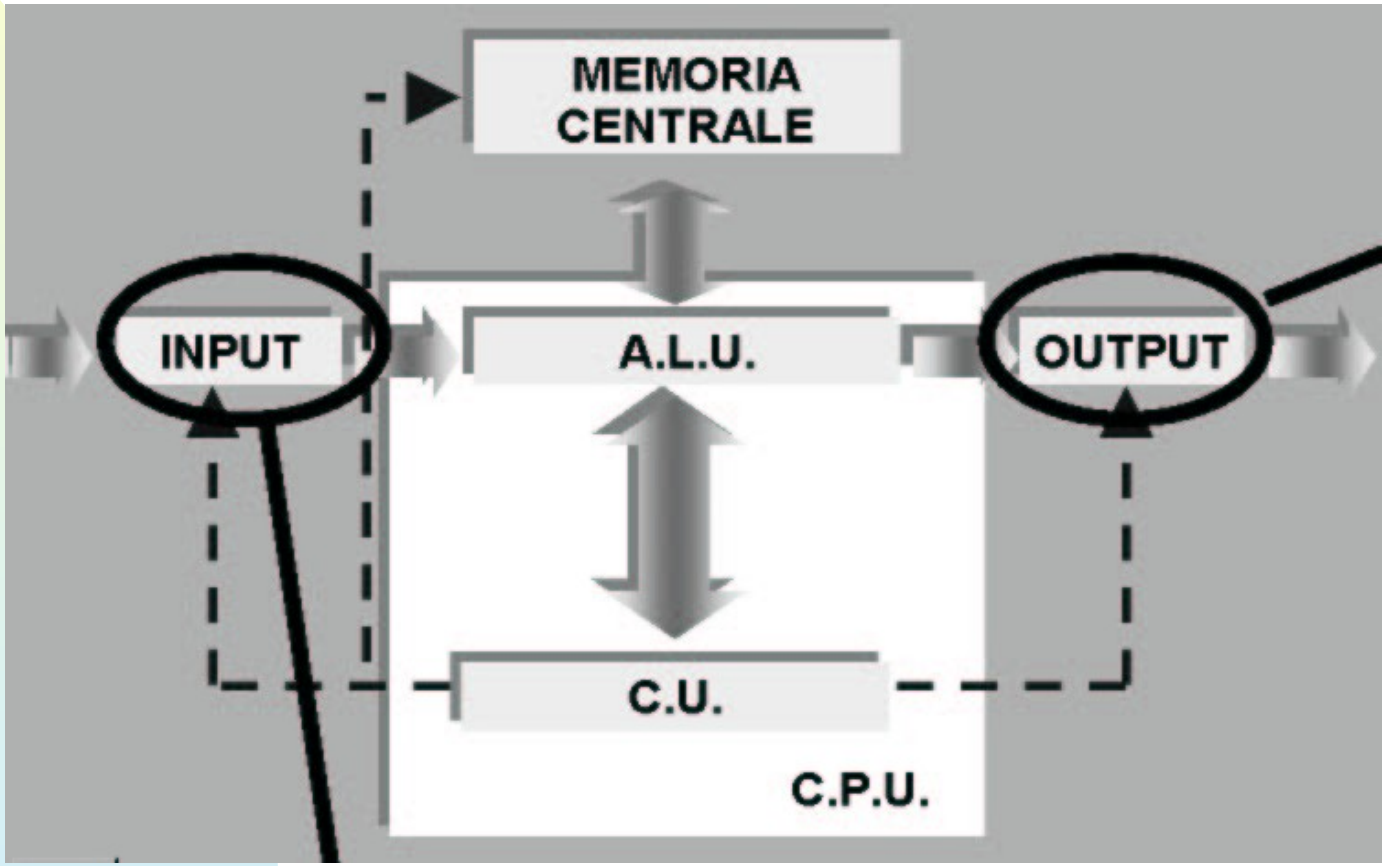


Motherboard



Una scheda madre – per gentile concessione della Intel

Architettura I/O



Il principio di funzionamento

- Programmi e dati risiedono in memoria secondaria.
- Per essere eseguiti (i programmi) e usati (i dati) vengono copiati nella memoria primaria.
- La CPU (Central Processing Unit) è in grado di eseguire le istruzioni che compongono i Programmi.

Memoria centrale (Principale)

memoria RAM (Random Access Memory)

l'accesso diretto alle celle

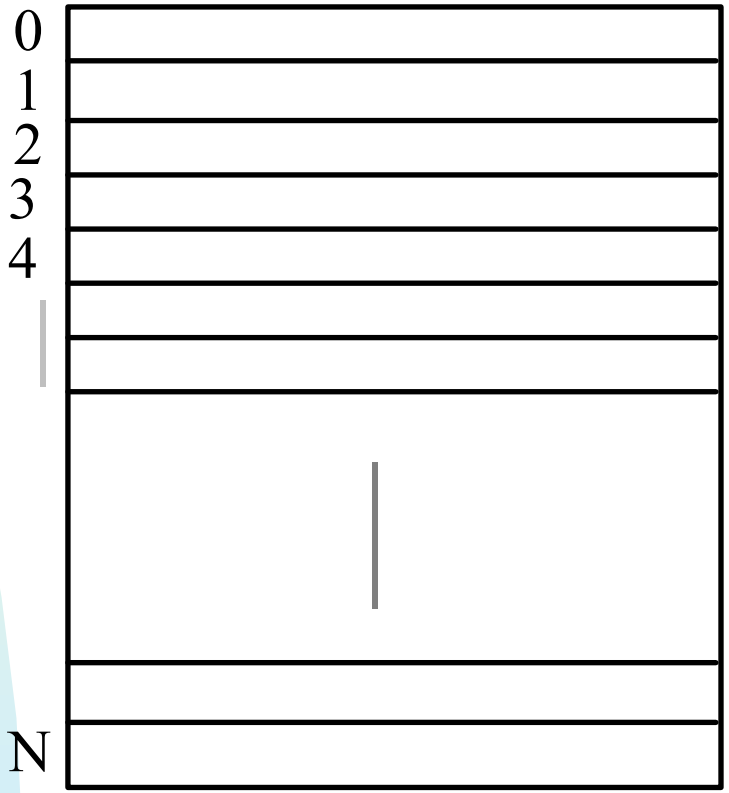
il tempo necessario per accedere ad una particolare cella è sempre il medesimo indipendentemente dalla posizione della cella nella sequenza

random (tradotto in italiano con *casuale*)

Memoria volatile (finchè c'è energia)

Memoria centrale (Principale)

Indirizzamento delle celle (locazioni di memoria)



Memoria centrale (Principale)

bit	1 cifra binaria	memorizza 0 oppure 1
byte	8 bit	memorizza un carattere
parola	da 16 a 64 bit	numeri e indirizzi di memoria
Kilobyte (KB)	1024 byte	circa mezza pagina di testo
Megabyte (MB)	1024 KB	un libro di 200 pagine
Gigabyte (GB)	1024 MB	alcuni volumi
Terabyte (TB)	1024 GB	una biblioteca
Petabyte (PB)	1024 TB	molte biblioteche

Memoria centrale (Principale)

Spazio di indirizzamento –

Utilizzando 16 bit si possono indirizzare

2^{16} = 64 KB di memoria

Utilizzando 32 bit è possibile indirizzare

2^{32} = 4 GB di memoria

Gli indirizzi a 32 bit sono lo standard nella maggior parte degli elaboratori anche se molti processori di ultima generazione lavorano già con indirizzi a 64 bit

Memoria centrale (Tipologie)

Le memorie **SRAM** (RAM statiche), dette anche flash-RAM, hanno costi elevati, tempi di accesso inferiori ai 2 nanosecondi e sono generalmente usate per la memoria *cache* .

Le memorie **DRAM** (RAM dinamiche) hanno tempi di accesso pari a 40-60 nanosecondi e sono utilizzate per la memoria principale degli elaboratori.

Le memorie **VRA** (RAM video) hanno caratteristiche molto particolari e sono usate dagli adattatori video per la gestione delle immagini. Sono più costose delle DRAM.

Memoria centrale (Tipologie)

Memorie DRAM per personal computer

- **SIMM da 30 pin**, usate nei computer di vecchia generazione;

- **SIMM da 72 pin**, usate in elaboratori con processore 80486 o Pentium;

- **DIMM da 168 pin** per elaboratori con processore Pentium II o superiori;

- **DDR da 178 pin** per elaboratori con processore Pentium IV o superiori - tempi di accesso dimezzati

- **SODIMM**, come le DIMM, ma specifiche per i portatili.

WORD(o PAROLA)

La **parola (word)** di una architettura:

quanti bit possono essere letti/scritti/usati dalla cpu con un unico accesso alla memoria (16, 32, 64, 128 bit)

Piu' grande e la **word**, maggiore e' la potenza del computer.

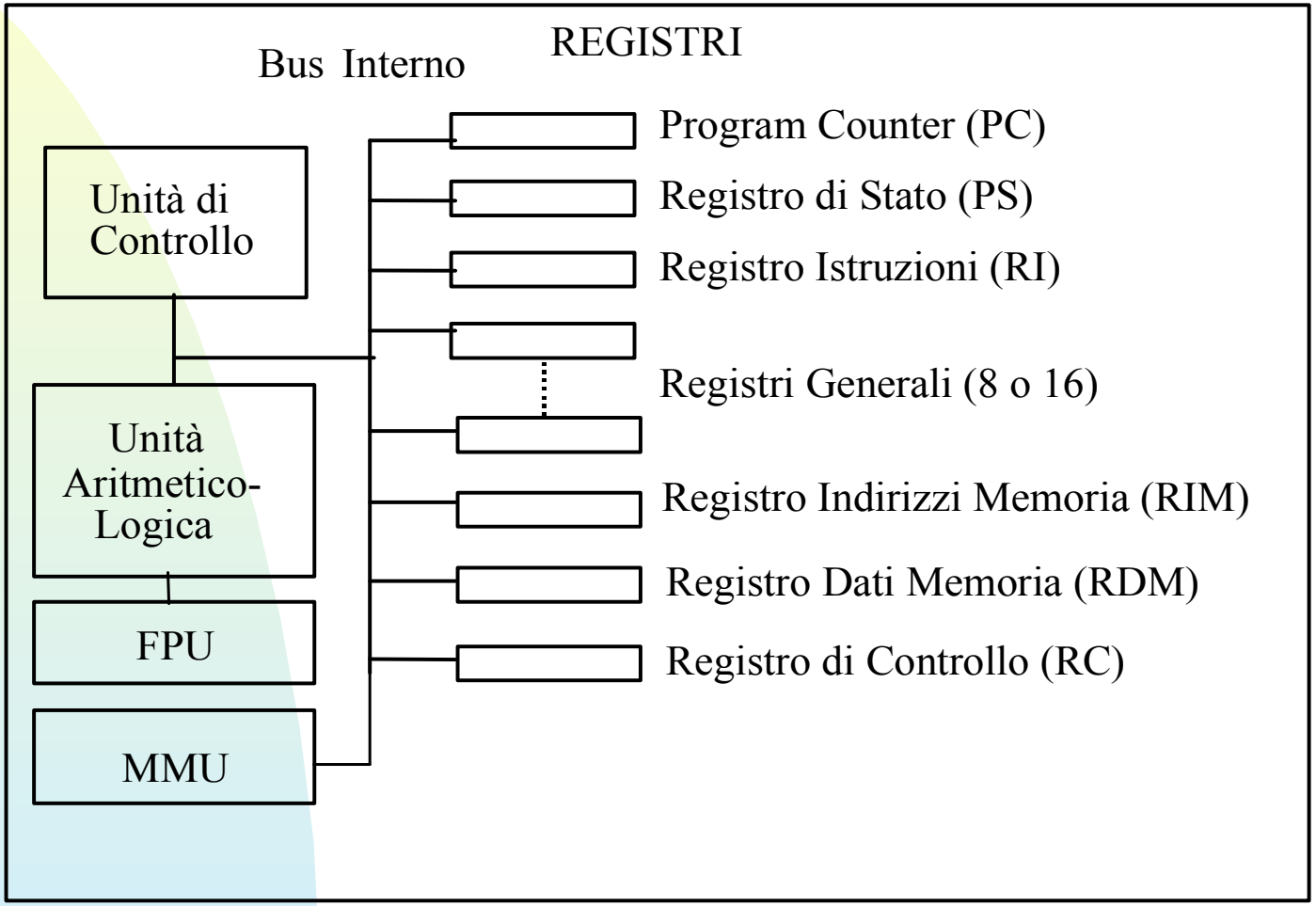
Attenzione – non influisce sullo spazio di indirizzamento - l'unità indirizzabile rimane sempre il byte.

Memoria ROM (Flash)

ROM = Read Only Memory (Memoria di sola lettura)

- Non può essere modificata
- A differenza della RAM non è volatile
- Veloce come la RAM
- Usata per memorizzare programmi e istruzioni di uso particolare
  - programmi di boot (accensione)
  - configurazione del sistema

IL PROCESSORE



IL PROCESSORE

CPU dall'inglese **C**entral **P**rocessing **U**nit.

E' l'unità che si occupa dell'elaborazione dei dati.

Esegue una e una sola istruzione alla volta.

Tale istruzione deve essere scritta in linguaggio macchina, cioè in sequenze di bit adatte al processore.

Il processo si chiama Fetch-execute

inserire le slide di CFE.PDF

IL PROCESSORE (BUS)

Generalmente i processori hanno *tre bus di base*:

il bus dati – che trasporta le informazioni da e verso il processore

il bus indirizzi – che trasporta gli indirizzi dei dati che devono essere letti o scritti

il bus di controllo – che trasporta i segnali di controllo usati dal processore e dalle altre componenti.

IL PROCESSORE (BUS)

La dimensione di un bus - detta ampiezza - determina la quantità di informazione trasferita contemporaneamente e generalmente si hanno bus a 32 bit ma nelle architetture più recenti anche a 64 bit. Ogni bus ha anche una velocità di trasferimento misurata in MHz

Generalmente gli elaboratori personal hanno processori con bus interno a 400 o 533 MHz anche se esistono già in commercio bus ad 800 MHz

IL PROCESSORE (U.C.)

L'Unità di Controllo (UC)

Ha il ruolo di coordinamento delle diverse attività che devono essere svolte all'interno del processore.

La frequenza con cui vengono eseguiti i cicli di esecuzione è scandita da un'ulteriore componente, detta **clock**, che genera impulsi a distanza di tempo costante; ad ogni impulso di clock la UC esegue un ciclo di esecuzione di istruzioni macchina.

La velocità di elaborazione di un processore dipende dalla frequenza del suo clock; tipici valori di frequenza dei processori attuali variano tra 1 GHz e 3 GHz (ossia tra 1000 e 3000 milioni di impulsi al secondo) e quindi essi possono eseguire fino a migliaia di milioni di istruzioni macchina al secondo.

IL PROCESSORE (Registri)

All'interno del processore ci sono due tipi di registri:

Registri speciali usati dall'unità di controllo per scopi particolari, ad esempio il registro *Program Counter, il Registro Istruzioni, il Registro di Stato e i registri di comunicazione con la memoria;*

Registri generali (o aritmetici) usati per salvare risultati parziali durante il corso dell'elaborazione.

I REGISTRI (P.C.)

Il Program Counter (PC)

Come abbiamo già detto il processore esegue, ad ogni ciclo, un'istruzione prelevata dalla memoria principale. Il problema diventa quello di conoscere l'indirizzo della cella di memoria in cui si trova la **prossima istruzione** da eseguire: questo indirizzo è memorizzato nel registro **P**rogram **C**ounter (**PC**).

I REGISTRI (RI e RS)

Registro Istruzioni (RI) contiene l'istruzione attualmente in esecuzione.

Ad ogni ciclo di clock l'istruzione letta dalla memoria principale viene scritta nel registro istruzioni RI.

Il Registro di Stato (PS) contiene le informazioni sullo stato di esecuzione del processore e può segnalare eventuali errori avvenuti durante l'esecuzione. Vi sono diversi tipi di errori che possono avvenire nel corso dell'esecuzione di un programma. Ad esempio overflow.

I REGISTRI (Generali)

Registri generali - sono in genere sedici o trentadue e servono come memoria temporanea per le operazioni interne del processore.

Le dimensioni dei registri coincidono con quelle delle parole di memoria.

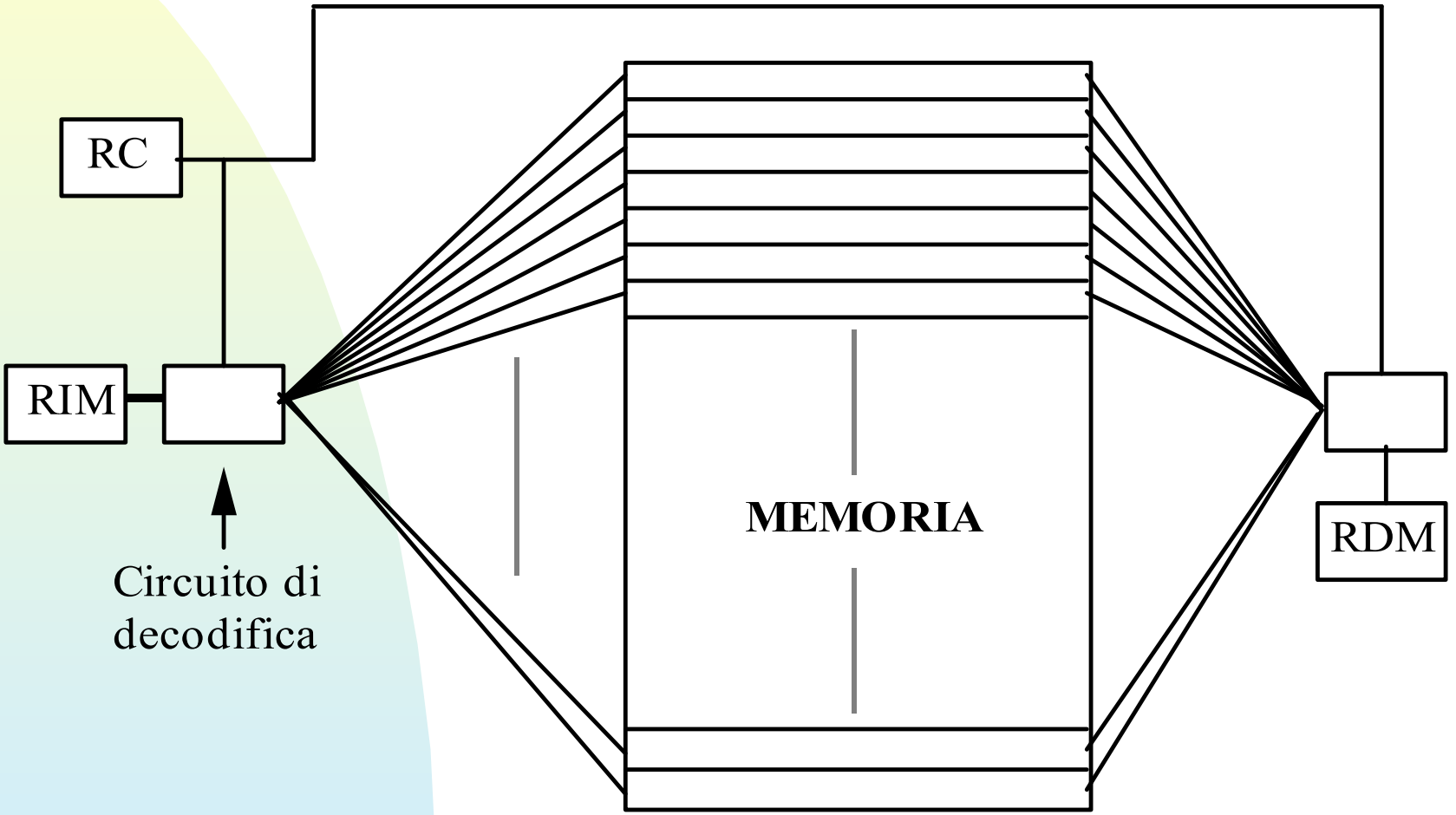
I REGISTRI (RIM, RDM, RC)

Registro Indirizzi Memoria (RIM), mediante il quale il processore può specificare l'indirizzo della cella su cui vuole operare

Registro Dati Memoria (RDM) che contiene l'informazione letta dalla memoria o quella che deve essere scritta nella memoria

Registro di Controllo (RC) mediante il quale il processore specifica l'operazione che deve essere eseguita e che, al termine dell'operazione, viene utilizzato per segnalare eventuali errori che possono essere avvenuti

Collegamento memoria- processore



IL PROCESSORE (ALU)

L'Unità Aritmetico-Logica (ALU) è costituita da un insieme di circuiti in grado di svolgere le operazioni di tipo aritmetico e logico.

La ALU legge i dati contenuti all'interno dei registri generali, esegue le operazioni e memorizza il risultato in uno dei registri generali.

I processori oggi in commercio hanno anche un particolare modulo integrato all'interno del processore chiamato **FPU** (**F**loating **P**oint **U**nit) o **NPU** (**N**umerical **P**rocessing **U**nit) che si occupa di potenziare e supportare la ALU nelle operazioni matematiche complesse.

IL PROCESSORE (MMU)

Unità di gestione della memoria (MMU)

Questo modulo, interno alla CPU e gestito direttamente dall'unità di controllo, si occupa di tutte le operazioni di indirizzamento da e verso la memoria.

MEMORIA CACHE

memoria cache serve per conservare i dati di uso molto frequente nel transito fra CPU e Memoria principale

Ha solitamente dimensioni abbastanza ridotte (qualche centinaio di KB) ed è molto più veloce della RAM. (usa tecnologia statica)

- **Cache interna** – o primaria, chiamata cache **L1**, realizzata all'interno del circuito della CPU (su bus veloce)

- **Cache esterna** – o secondaria, chiamata cache **L2**, situata sulla scheda madre, più lenta ma espandibile.

MEMORIA CACHE

L'efficacia della cache dipende dai criteri funzionali adottati.

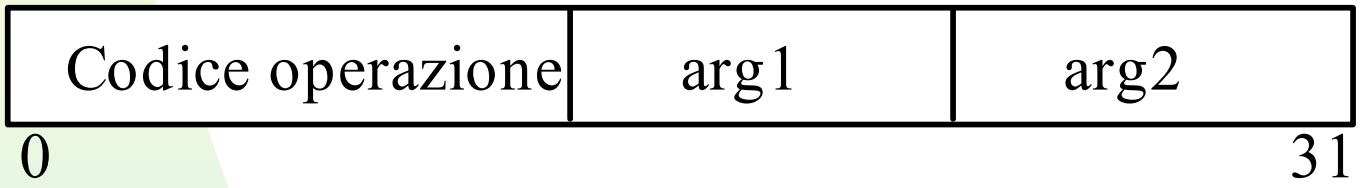
-***locality of reference*** Statisticamente, se il processore ha letto un dato in memoria, molto probabilmente avrà bisogno anche di quello successivo

-cache di tipo ***associativo*** ossia basate su una tecnica di accesso alle informazioni logica e non fisica.

TIPI DI PROCESSORI

Processori **CISC**
(**C**omplete **I**nstruction **S**et **C**omputers)
Molte istruzioni complesse, ma richiedono più cicli macchina
Processori **RISC**
(**R**educed **I**nstruction **S**et **C**omputers)
Ridotto set di istruzioni, ma processate in un solo ciclo macchina
Microcontrollori
Unità autonome complete di memoria, I/O, bus, sistema operativo, ecc.

Assembler



Assembler

Istruzioni di lettura dalla memoria. Solitamente con due argomenti: uno specifica l'indirizzo della cella di memoria da cui si deve leggere un'informazione; l'altro il registro in cui l'informazione deve essere scritta.

Istruzioni di scrittura in memoria. Solitamente con due argomenti: uno specifica il registro da cui l'informazione deve essere prelevata; l'altro l'indirizzo della cella di memoria in cui l'informazione deve essere scritta.

Assembler

Istruzioni aritmetiche. Permettono di eseguire le operazioni aritmetiche operando sui dati contenuti nei registri. Di solito queste istruzioni hanno due o tre argomenti. Due argomenti specificano i registri in cui si trovano gli operandi; il terzo specifica il registro in cui deve essere scritto il risultato. Qualora non vi sia il terzo argomento, il risultato viene scritto in uno dei registri in cui si trovavano gli operandi.

Istruzioni logiche. Permettono di eseguire le operazioni logiche sugli operandi contenuti nei registri specificati come argomenti dell'istruzione. Esempi di operazioni logiche sono il confronto o l'inversione di bit.

Assembler

Istruzioni di spostamento. Permettono di spostare le informazioni da una cella di memoria ad un'altra; gli indirizzi delle due celle sono gli argomenti dell'istruzione.

Istruzioni di salto. Per spostare il flusso del programma da un'istruzione ad un'altra non immediatamente successiva

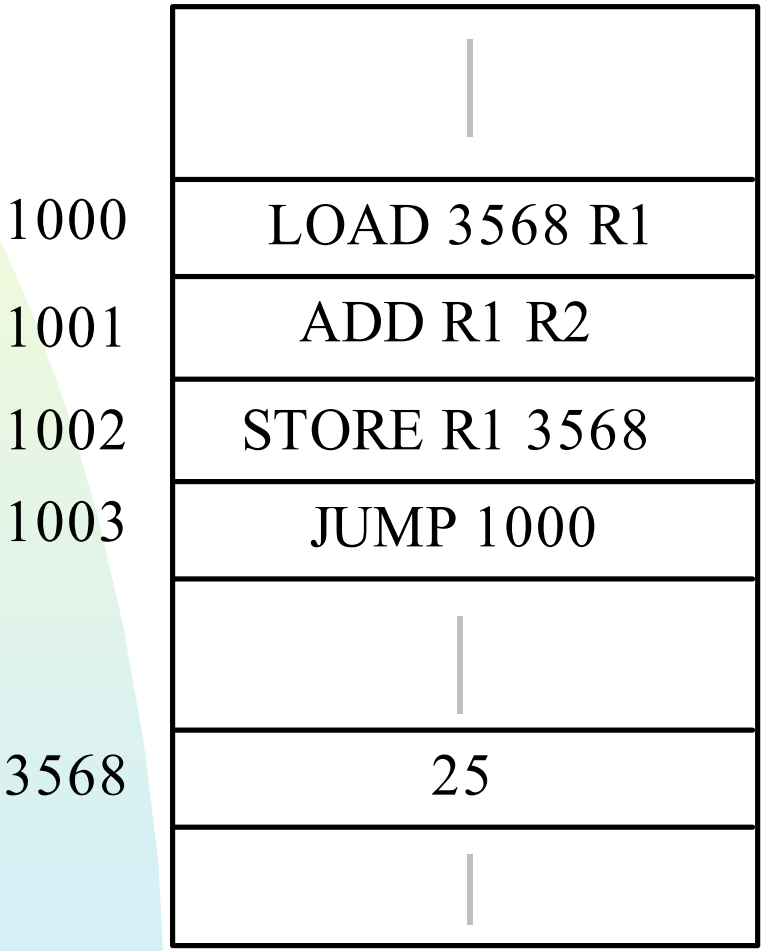
Assembler

Ad ogni ciclo di clock
- si legge dalla memoria principale l'istruzione che si trova all'indirizzo indicato dal registro PC;
- l'istruzione viene scritta all'interno del registro RI;
- si modifica il valore del PC aumentandolo di 1;
- la UC decodifica l'istruzione e individua la sequenza di azioni che devono essere svolte all'interno del processore;
- si eseguono le azioni specificate dall'istruzione, e cioè:
5.1 in corrispondenza di istruzioni di lettura dalla memoria, la UC attiva i meccanismi di lettura;
5.2 in corrispondenza di istruzioni di scrittura in memoria, la UC attiva i meccanismi di scrittura;
5.3 in corrispondenza di istruzioni aritmetiche o logiche la UC attiva la ALU per eseguire l'operazione;
5.4 in corrispondenza di istruzioni di salto si modifica il valore del registro PC scrivendo l'indirizzo della istruzione cui si deve saltare.

Assembler

Qualsiasi assembler richiede però una compilazione in linguaggio macchina attraverso un opportuno COMPILATORE (e Linker)

Assembler



Assembler

LOAD 3568 R1
Operazione di lettura dalla memoria: richiede la lettura del valore contenuto nella cella con indirizzo 3568 e il suo caricamento nel registro R1.

ADD R1 R2
Operazione aritmetica di somma: prevede la somma del contenuto dei registri R1 e R2 e il caricamento del risultato nel registro R1.

STORE R1 3568
Operazione di scrittura in memoria: richiede la scrittura del valore contenuto nel registro R1 nella cella con indirizzo 3568.

JUMP 1000
Istruzione di salto: prevede il salto all'istruzione con indirizzo 1000.

Prima istruzione (PC=1000)

-Passo 1. Legge dalla memoria principale la prossima istruzione da eseguire. Il valore del PC (1000) viene copiato nel registro RIM e viene scritto il comando di lettura nel registro RC. Come effetto, il contenuto della cella di indirizzo 1000 viene copiato nel registro RDM che conterrà quindi l'istruzione LOAD 3568 R1.

-Passo 2. Copia il contenuto del registro RDM nel registro istruzioni RI.

-Passo 3. Incrementa di uno il valore del PC che contiene ora il valore 1001, l'indirizzo della prossima istruzione da eseguire.

Prima istruzione (PC=1000)

-*Passo 4*. La UC si preoccupa di decodificare l'istruzione appena copiata nel RI. Poiché si tratta dell'istruzione LOAD, cioè di un'istruzione di lettura dalla memoria, la UC predispone le azioni per tale operazione.

-*Passo 5*. Vengono eseguite le operazioni corrispondenti ad una operazione di lettura dalla memoria. L'indirizzo della cella da leggere (3568) viene scritto nel registro RIM e viene dato un comando di lettura. Appena il dato è disponibile in RDM, esso viene letto e copiato nel registro R1, come richiesto dall'istruzione.

-Al termine dell'operazione il registro R1 conterrà il numero 25.

Seconda istruzione (PC=1001)

Al successivo impulso di clock si legge l'istruzione di indirizzo 1001 e la si scrive nel registro RI (ciò avviene esattamente come nel caso precedente). La UC incrementa di uno il valore di PC (che assume ora il valore 1002), decodifica l'istruzione e, accorgendosi che si tratta dell'istruzione aritmetica ADD, attiva gli opportuni circuiti della ALU affinché, come specificato dall'istruzione, si sommi il contenuto dei registri R1 e R2. Se supponiamo che il registro R2 contenga il valore 30, il risultato della somma è 55 e viene scritto nel registro generale R1 (assumendo che il risultato della somma venga depositato nel primo dei due registri da sommare).

Terza istruzione (PC=1002)

Al successivo impulso di clock si legge l'istruzione di indirizzo 1002 e la si scrive nel registro RI. Intanto il valore del registro PC viene aumentato di uno e diventa 1003. Poiché l'istruzione richiede la scrittura in memoria del contenuto del registro R1, vengono attivate le operazioni di scrittura; al loro termine il nuovo valore memorizzato nella cella di indirizzo 3568 è 55

Quarta istruzione (PC=1003)

Al successivo impulso di clock si analizza l'istruzione di indirizzo 1003 che viene letta e caricata nel registro RI; il registro PC, aumentato di uno, vale ora 1004. La UC decodifica l'istruzione e, poiché si tratta dell'istruzione di salto (JUMP), scrive nel registro PC l'indirizzo 1000 a cui saltare. La prossima istruzione sarà quindi di nuovo quella con indirizzo 1000. L'operazione di JUMP permette infatti di realizzare una ripetizione della stessa sequenza di istruzioni.

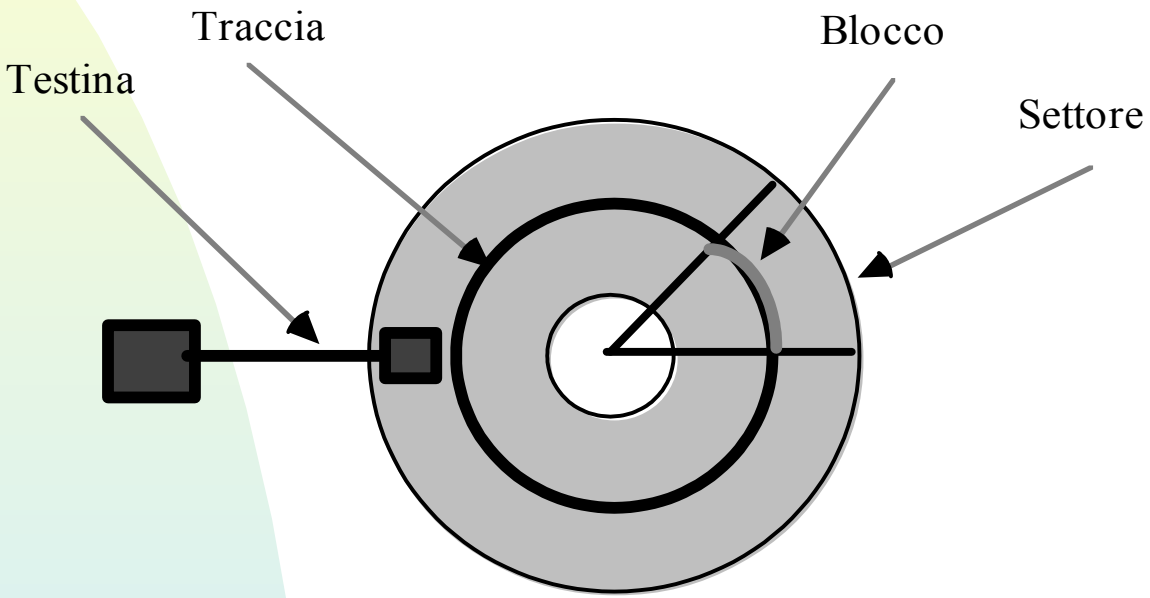
La memoria secondaria

La RAM dispone di uno spazio limitato. Programmi e dati risiedono normalmente in memoria secondaria (o memoria di massa) Quando si lancia un programma questo viene copiato dalla memoria secondaria (di solito un hard disk) in memoria primaria. Questa operazione si chiama **caricamento** (eseguita dal sistema operativo).

MEMORIA SECONDARIA

- Alta capacità di archiviazione
- Tempi di accesso lenti
- Costi bassi
- Memorizzazione permanente
- Meccanica in movimento
- Richiede una tecnologia e un supporto
    - Magnetico (Floppy, Hard disk)
    - Ottico (CDRom, DVD)
    - Magneto-ottico

Struttura del disco



Fattore di blocco

Il blocco di informazione che viene letto (scritto) in una singola operazione prende il nome di **record fisico** e costituisce la minima quantità di informazione indirizzabile.

Le dimensioni dei record logici non corrispondono alle dimensioni dei record fisici e il rapporto tra tali dimensioni prende il nome di **fattore di blocco**. Per essere memorizzato, un file viene suddiviso in blocchi della dimensione dei record fisici e quindi un record logico può essere contenuto in uno o più record fisici.

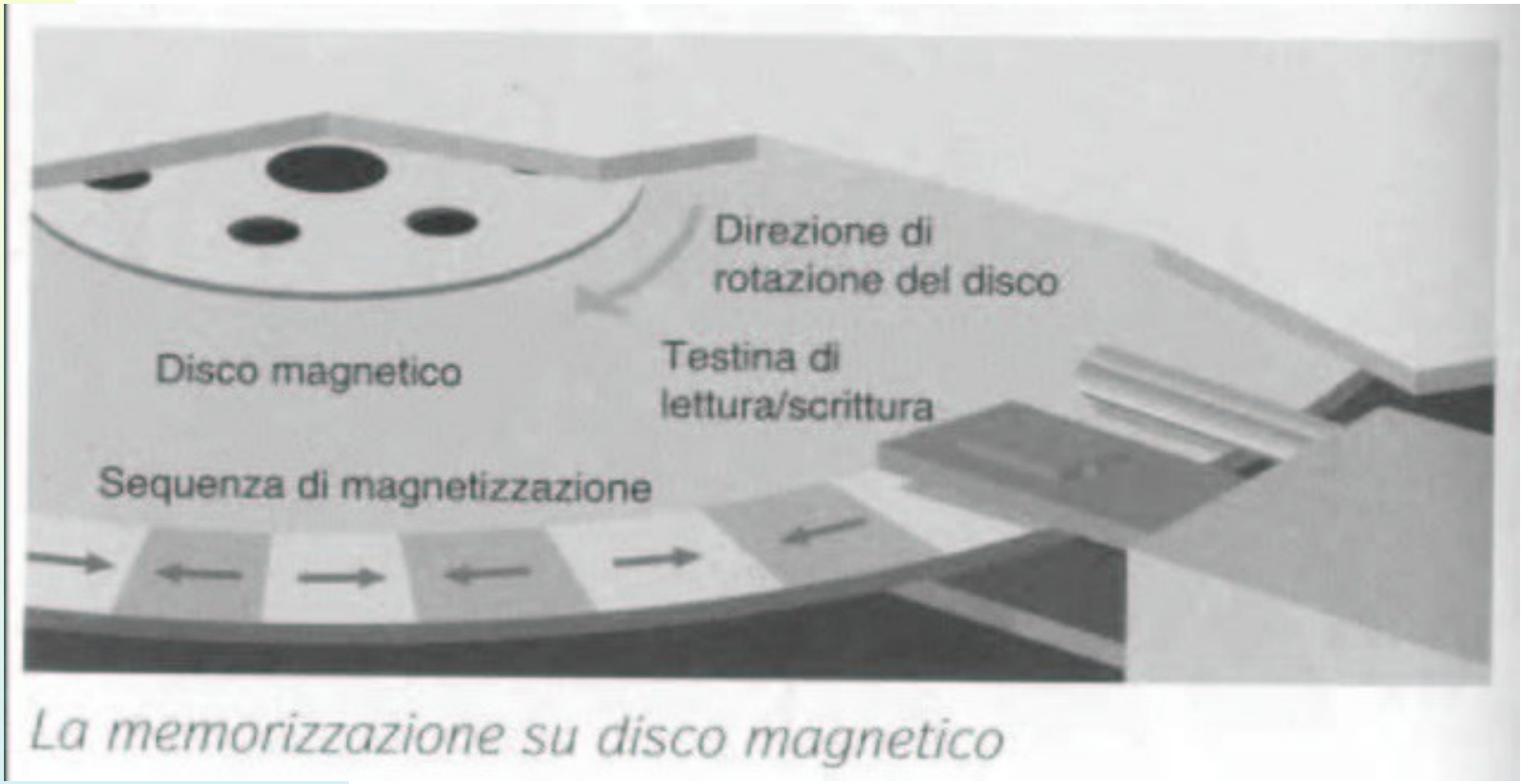
Tempi di accesso

- Lo spostamento della testina in senso radiale fino a raggiungere la traccia desiderata (**seek time**)

- L'attesa che il settore desiderato si trovi a passare sotto la testina; tale tempo dipende dalla velocità di rotazione del disco (**latency time**);

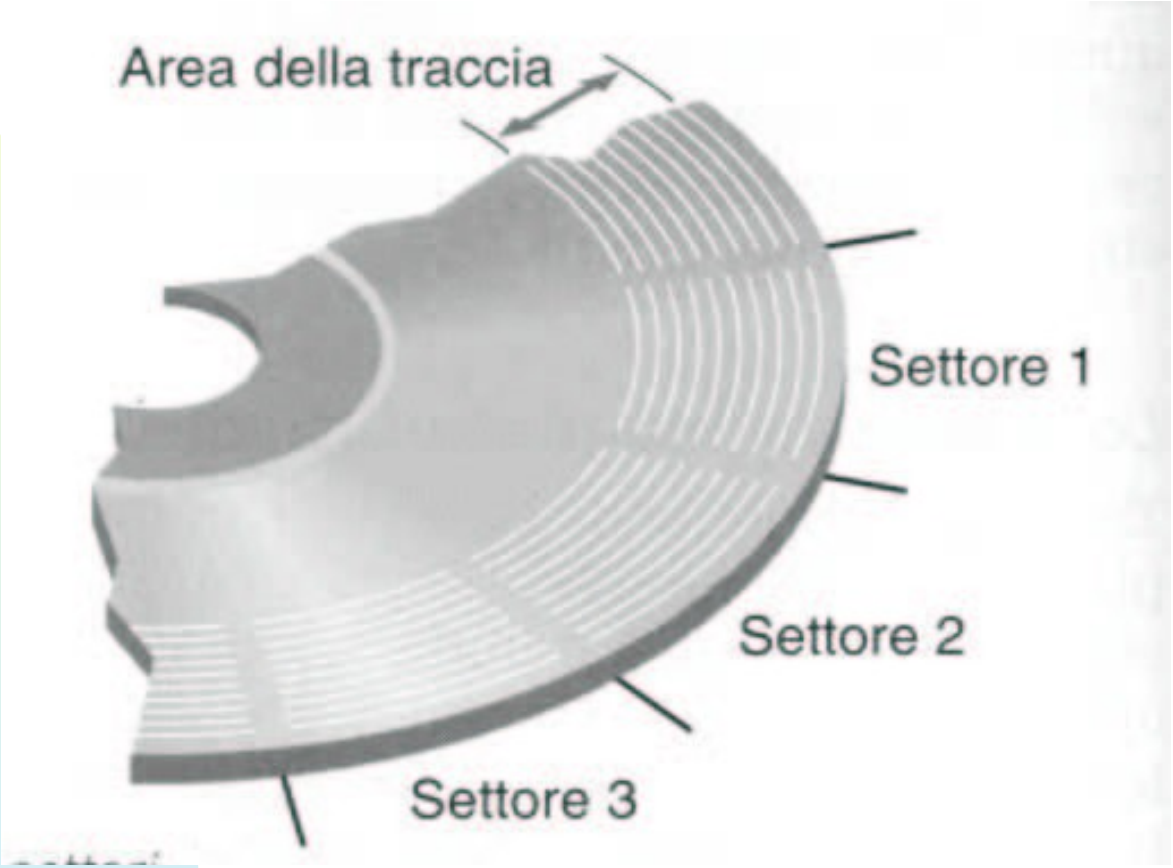
- Il tempo di lettura vero e proprio dell'informazione.

Memoria secondaria

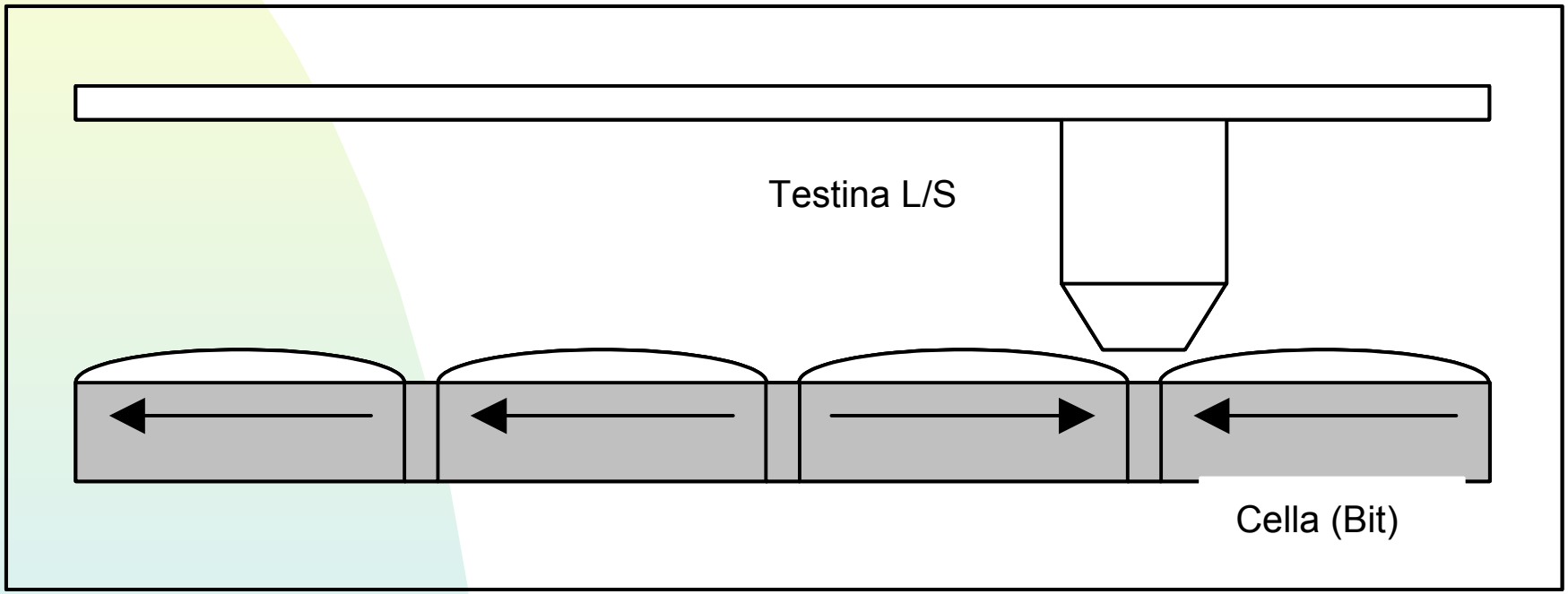


La memorizzazione su disco magnetico

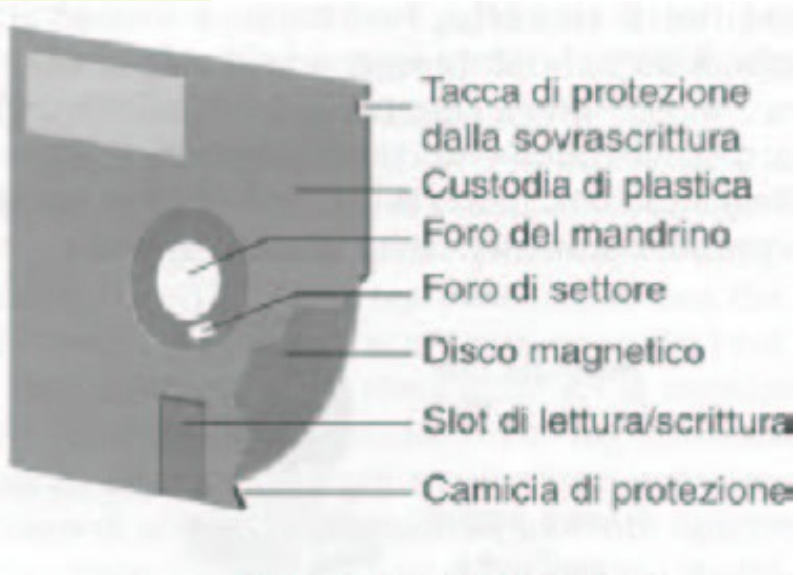
Organizzazione dei dati su disco

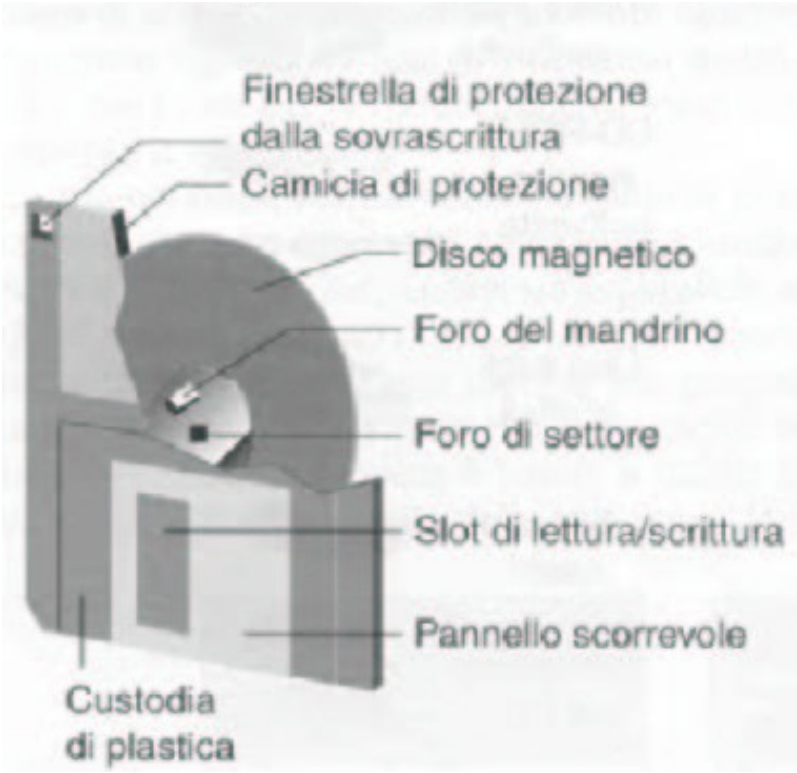


Tecnica di memorizzazione

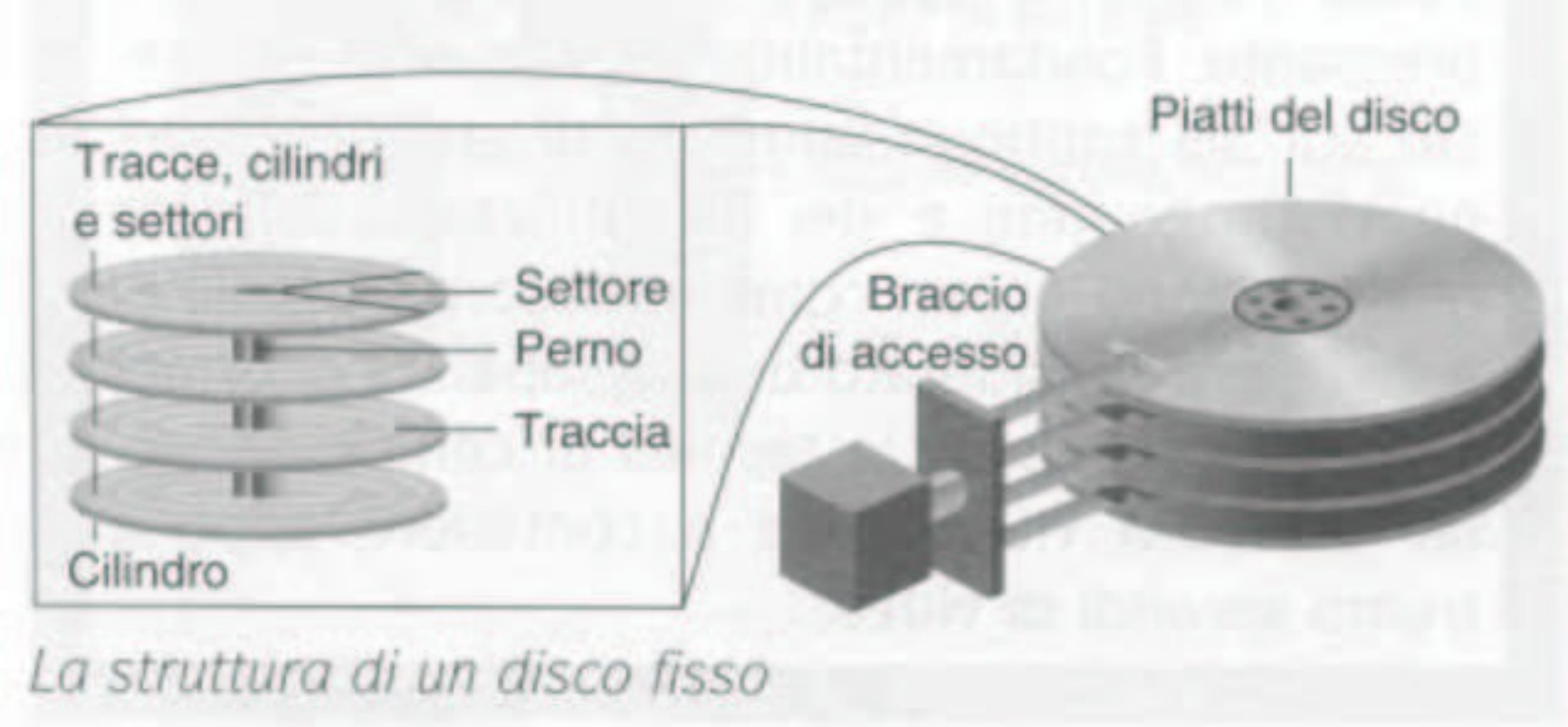


Floppy disk





Struttura HD



La struttura di un disco fisso

DISKPACK

RAID

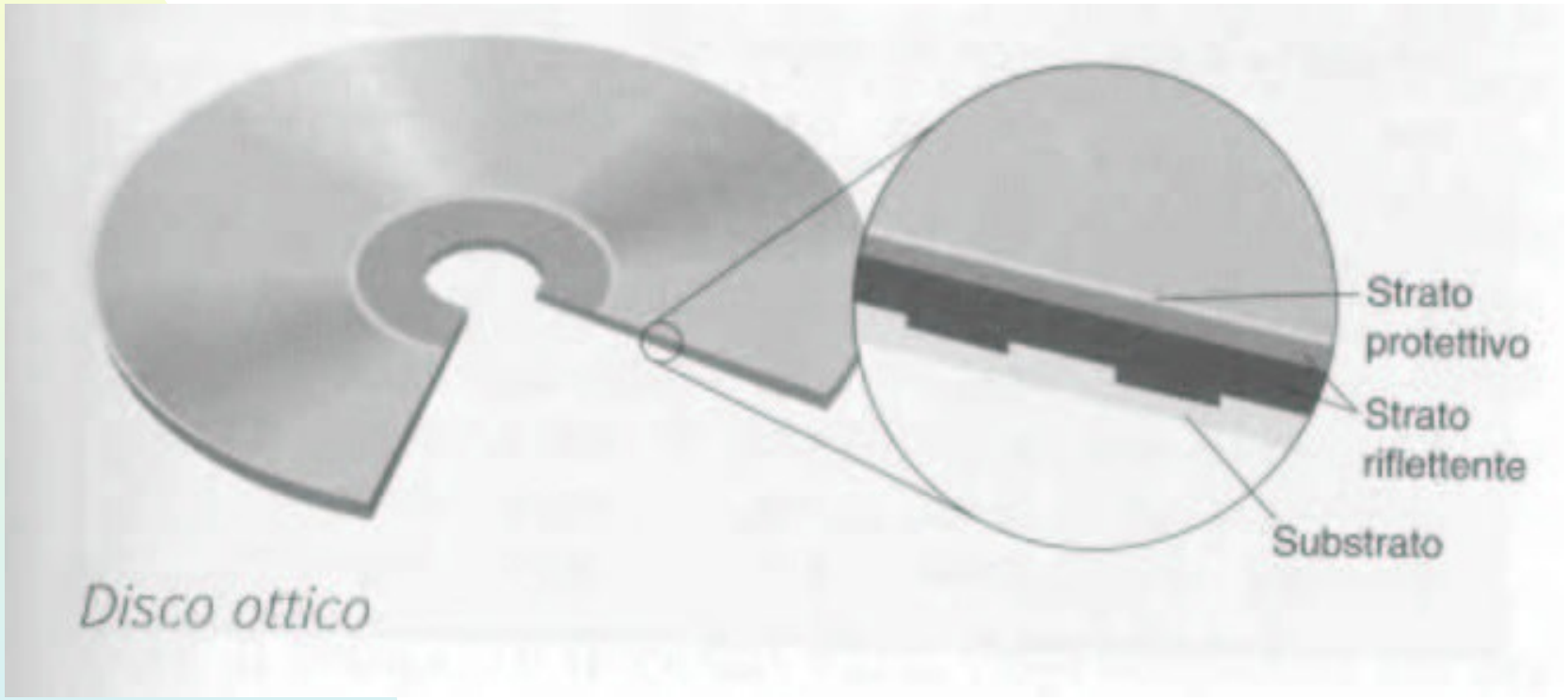
- **RAID Level 0**. È la più semplice: permette la distribuzione automatica dei dati su più dischi ma poiché i dati non sono replicati non garantisce la tolleranza ai guasti in caso di rottura di un disco.

- **RAID Level 1**. È la tecnica più diffusa; i dati vengono replicati (*mirroring*) e nel caso di rottura di un disco si passa automaticamente ad un altro disco senza alcuna perdita di dati.

- **RAID Level 3**. È molto simile alla tecnica di livello 0 ma dedica un hard disk al *recupero automatico d'errore* mediante la tecnica del controllo di parità.

- **RAID Level 5**. Si usa nel caso di sistemi che richiedono una garanzia di elevata protezione dei dati e alte prestazioni. Utilizza le tecniche di *data striping* completo e un disco dedicato alla correzione d'errore.

Struttura CDRom



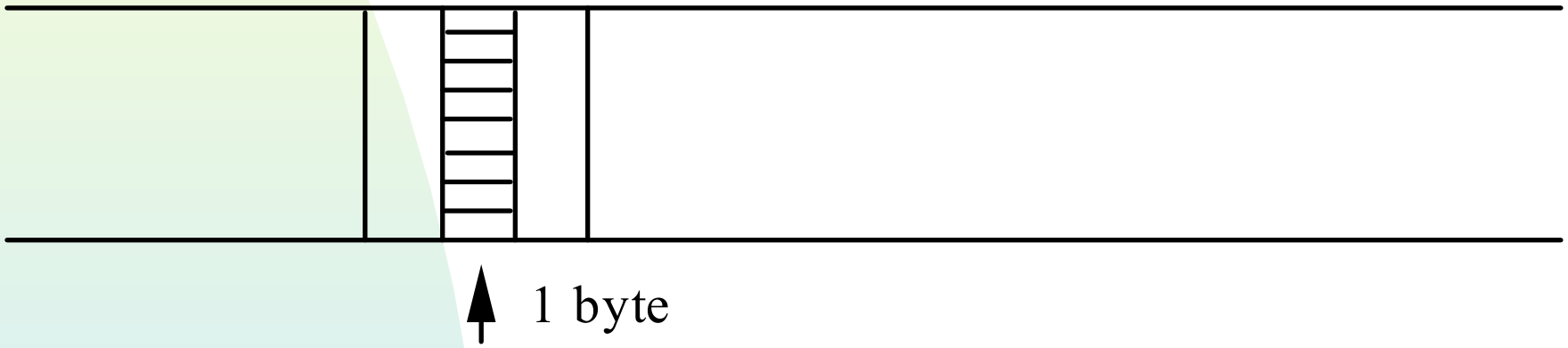
Tipi di memorie secondarie

Raffronto tra diverse memorie secondarie

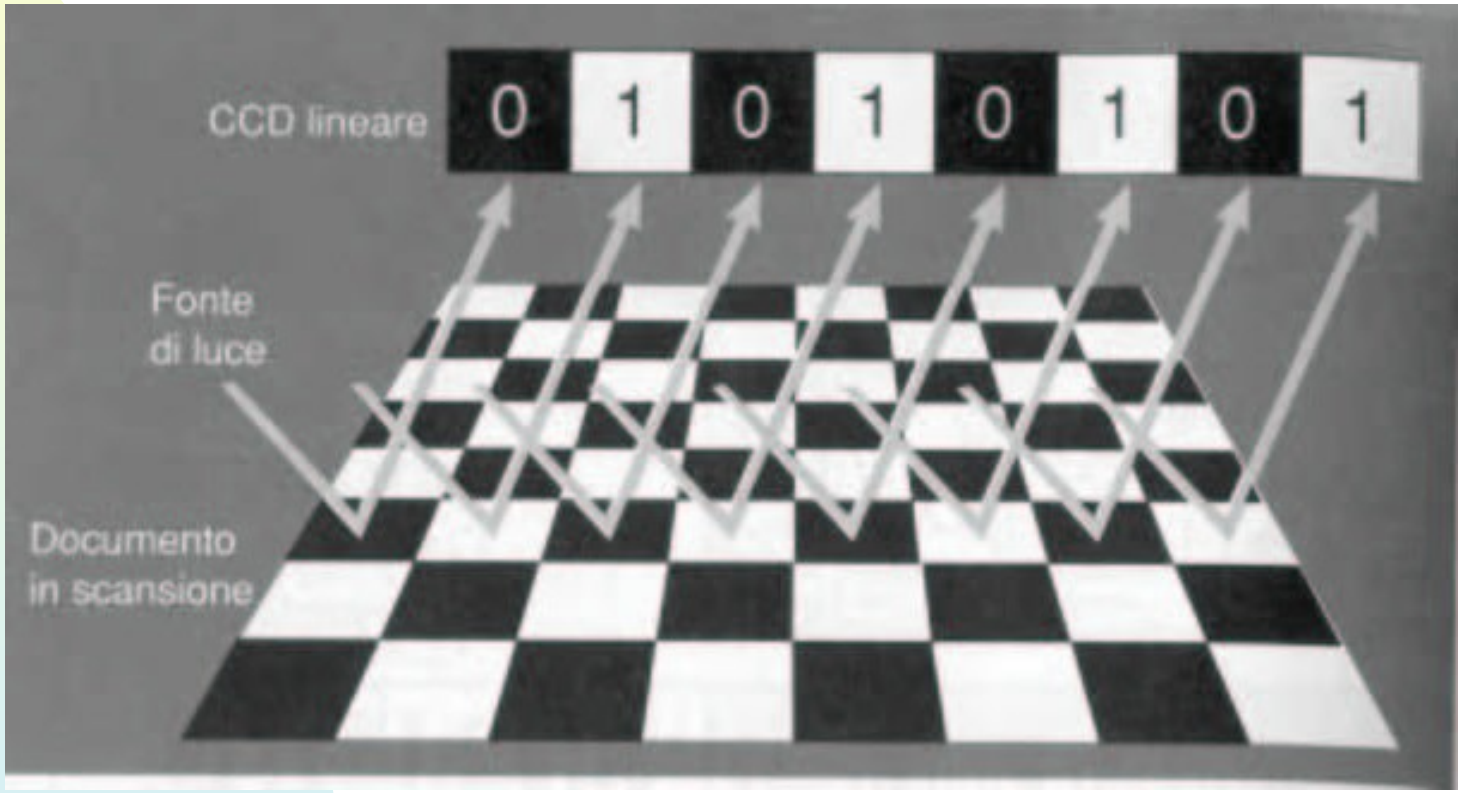
Tipo	Velocità	Capacità	Costo	Registrazione
RAM	Alta	Bassa	Alta	Elettronico
Dischetto	Bassa	Bassa	Medio	Magnetico
Disco rigido	Alta	Alta	Medio	Magnetico
CD ROM	Bassa	Alta	Bassa	Ottico
Disco magneto-ottico	Medio	Alta	Alta	Magneto/Ottico
Nastro da 0,25 pollici	Bassa	Alta	Bassa	Magnetico
Nastro DAT	Bassa	Alta	Bassa	Magnetico

Memoria flash

DAT / DLT / ULTRIUM



Scanner



Bus esterni

PCI Peripheral Component Interconnect	**Attuale standard di espansione, compatto e veloce, è un bus a 32 o 64 bit e supporta il *plug & play*.**
AGP Accelerated Graphics Port	**Dedicato alla scheda video, supporta diverse velocità di comunicazione e viene usato per la grafica tridimensionale.**
PCMCIA Personal Computer Memory Card International Association	**Standard per molte schede di espansione oggi in commercio. Sviluppato originariamente per aggiungere memoria ai computer portatili, questo standard è stato esteso più volte ed è oggi usato per molti dispositivi. Esistono schede PCMCIA che hanno la stessa dimensione rettangolare ma diverso spessore e sono usate per espansioni di memoria (3.3 mm), per schede modem o di rete (5 mm), per hard disk, modem, schede wireless (10.5 mm).**

Video

VGA (**V**ideo **G**raphics **A**rray). 800X600 pixel e 65535 colori (SVGA, XVGA o UVGA)

risoluzioni anche più elevate (1600X1200 e 16 milioni di colori);

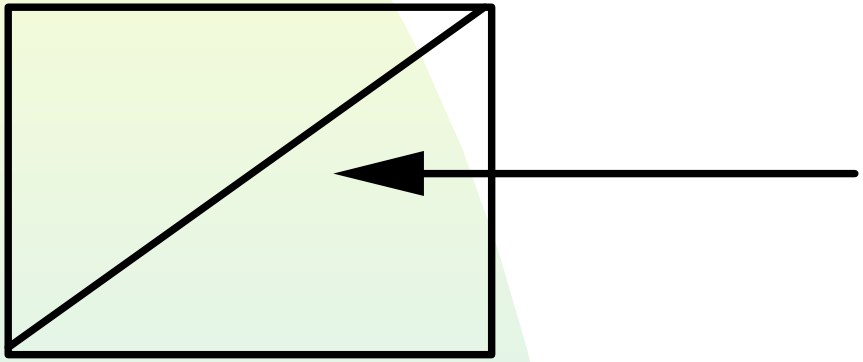
alta risoluzione si arriva anche a 4096X3300 pixel

Video CRT

Display LCD TFT attivo

frequenza di refresh (*refresh rate*). Tanto maggiore è questa frequenza, tanto più l'immagine apparirà stabile. Frequenze a partire da 70 Hz consentono immagini stabili su video con una buona risoluzione.

Misura del video



misura in pollici di un video

Stampanti

Stampanti a getto di inchiostro.

- **Piezoelettrica.**
- **Termica.**
- **Flusso continuo**

Stampanti laser.

led;

Altre periferiche

Scanner - software OCR (Optical Character Recognition)

I lettori di codici a barre

Le schede fax

Le schede audio (sound blaster)

Le schede video

Le schede di rete

I plotter

Fotocamere e telecamere

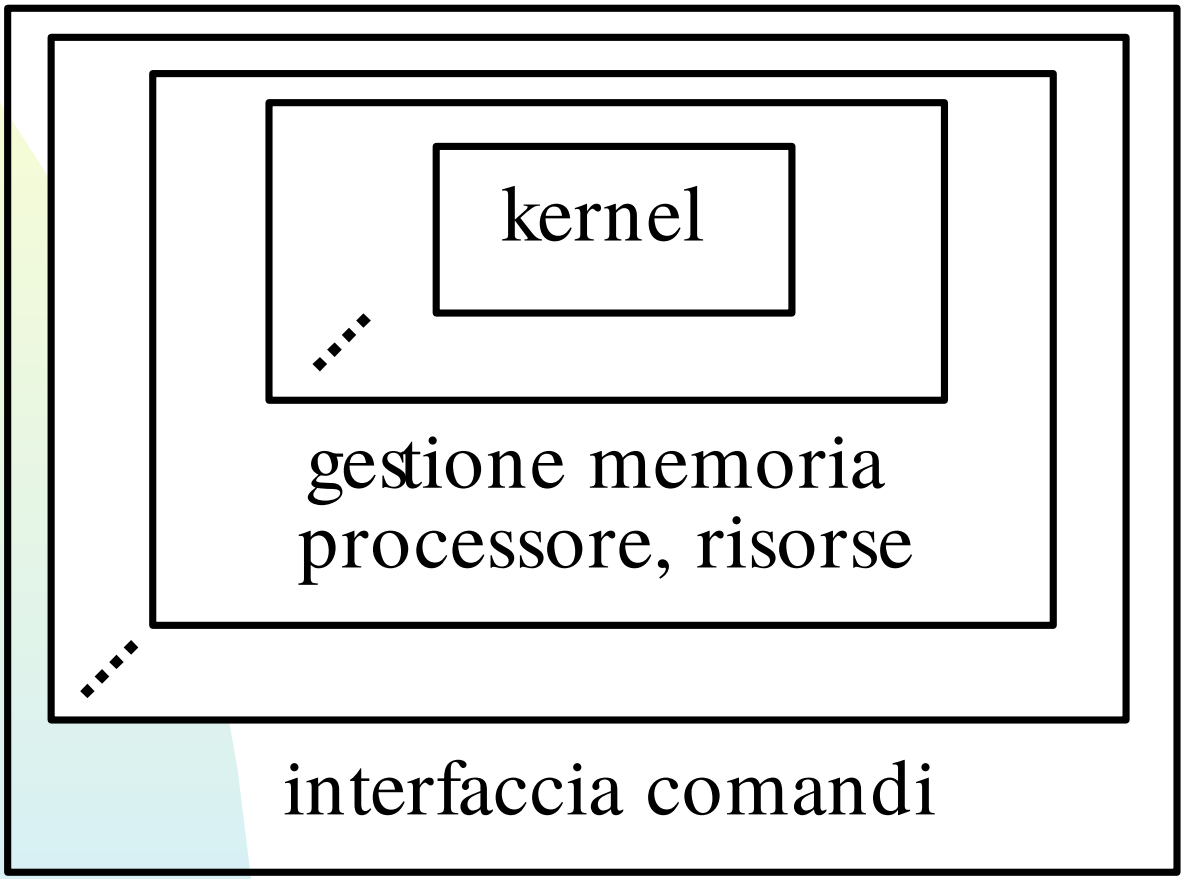
Sistemi operativi

Sistemi operativi 1

-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.

Sistemi operativi 2



Struttura a cipolla

Struttura teorica del S.O.

Sistemi operativi 3

-Avviamento dell'elaboratore e creazione dell'ambiente virtuale;

-Gestione del processore e dei processi;

-Gestione della memoria principale;

-Gestione della memoria virtuale;

-Gestione della memoria secondaria;

-Gestione delle risorse di input/output (periferiche)

-Linguaggio di comandi per l'interazione con l'utente

-Compilazione

Sistemi operativi

Più semplicemente :

- Gestione del processore e dei **processi**
- gestione della memoria principale
- gestione della **memoria virtuale**
- gestione della memoria secondaria (il **file system**)

Avviamento dell'elaboratore

Il Boot del S.O.

- 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, ossia che fornisce all'utente un linguaggio di comandi per l'interazione con il sistema.

Sistemi operativi 4

Il ruolo del processore è quello di *eseguire programmi.*
Più precisamente consideriamo la seguente definizione di processo:

un processo è un programma in esecuzione

Sistemi operativi

- Mono-utente o Multi-utente (Mono/multi-user)
- Mono o Multi-programmati (Mono/multi-tasking)
- Distribuiti (o di rete)

Sistemi Mono task – mono user

- Un solo utente puo' eseguire un programma per volta.

- Il programma viene lanciato, esegue le proprie funzioni e termina

-Un solo processo attivo.

-I diversi processi sono sequenziali

- Ma la CPU viene sfruttata al meglio?

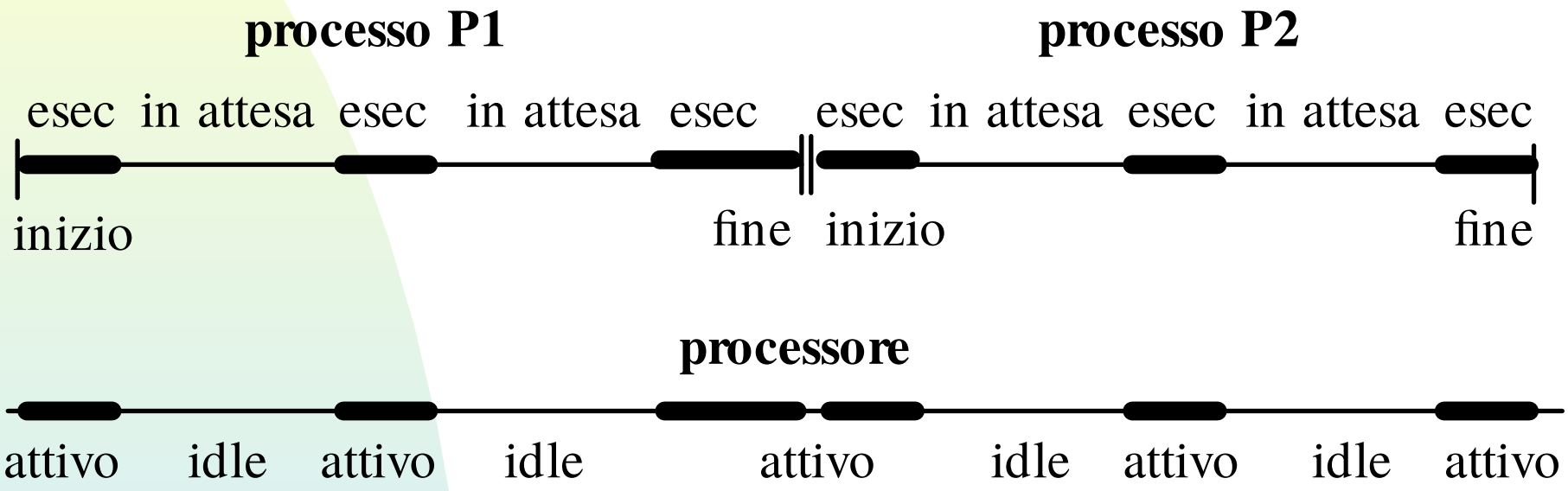
Sistemi Mono task – mono user

- La CPU e' molto piu' veloce dei dischi e delle periferiche, e passa la maggior parte del suo tempo in attesa del completamento delle operazioni di questi **devices**.

-Durante l'attesa si dice che la CPU e' in uno stato inattivo detto **idle**.

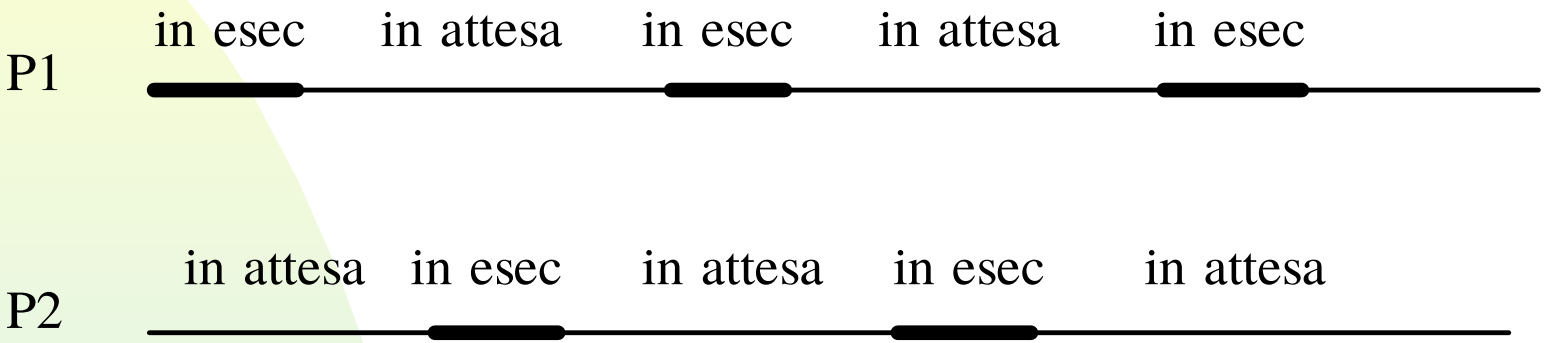
-Quindi la necessità di più processi in parallelo

Processi



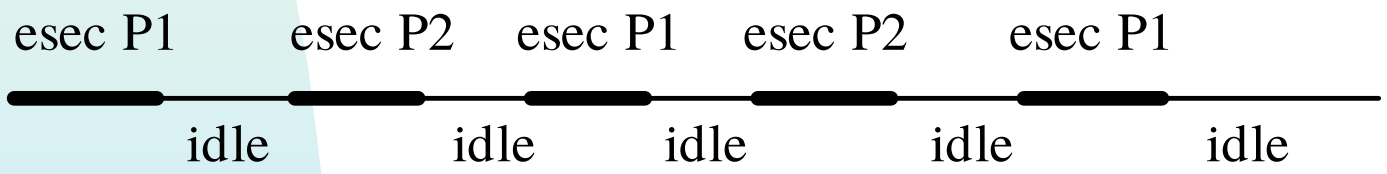
Multitask e multiutenza

Dal punto di vista dei processi:



(a)

Dal punto di vista del processore



(b)

Stati del processo

Un processo può trovarsi in **tre** diversi **stati**:

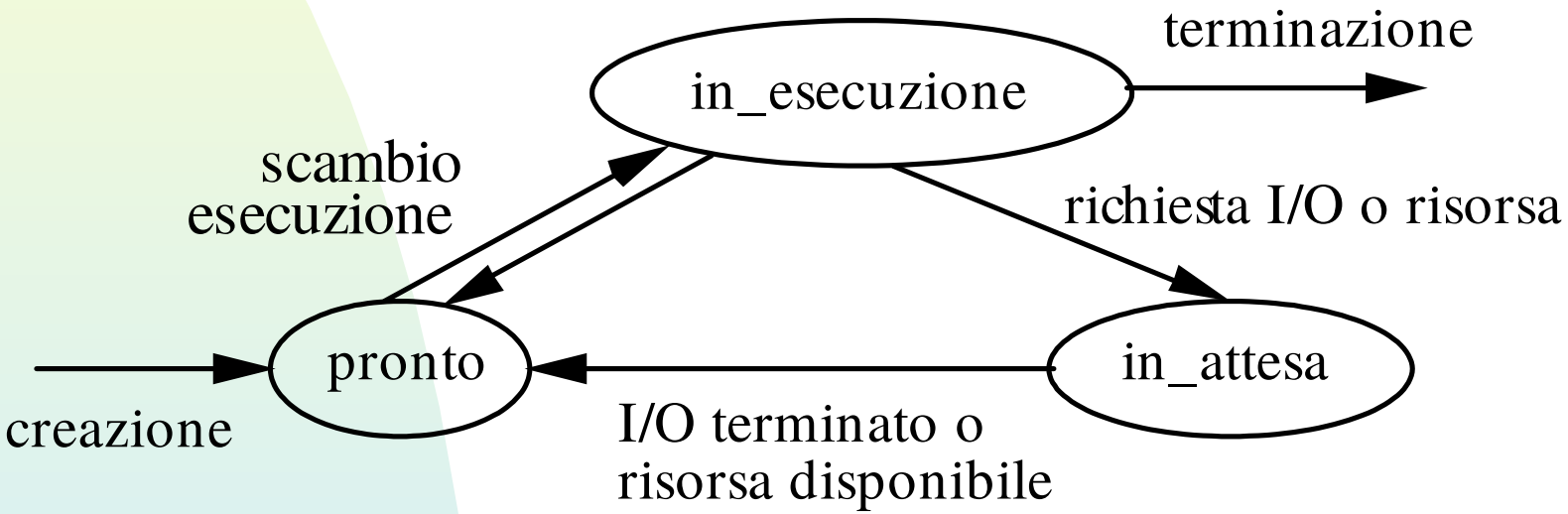
- in **esecuzione**, quando sta utilizzando il processore;
- in **attesa (bloccato)**, quando è in attesa del verificarsi di un evento esterno, ad esempio, la terminazione di un'operazione di input/output o la possibilità di utilizzare una qualche risorsa in uso da parte di altri processi;
- **pronto**, quando è potenzialmente in condizione di poter utilizzare il processore che è occupato da un altro processo.

Abbandono dell'esecuzione

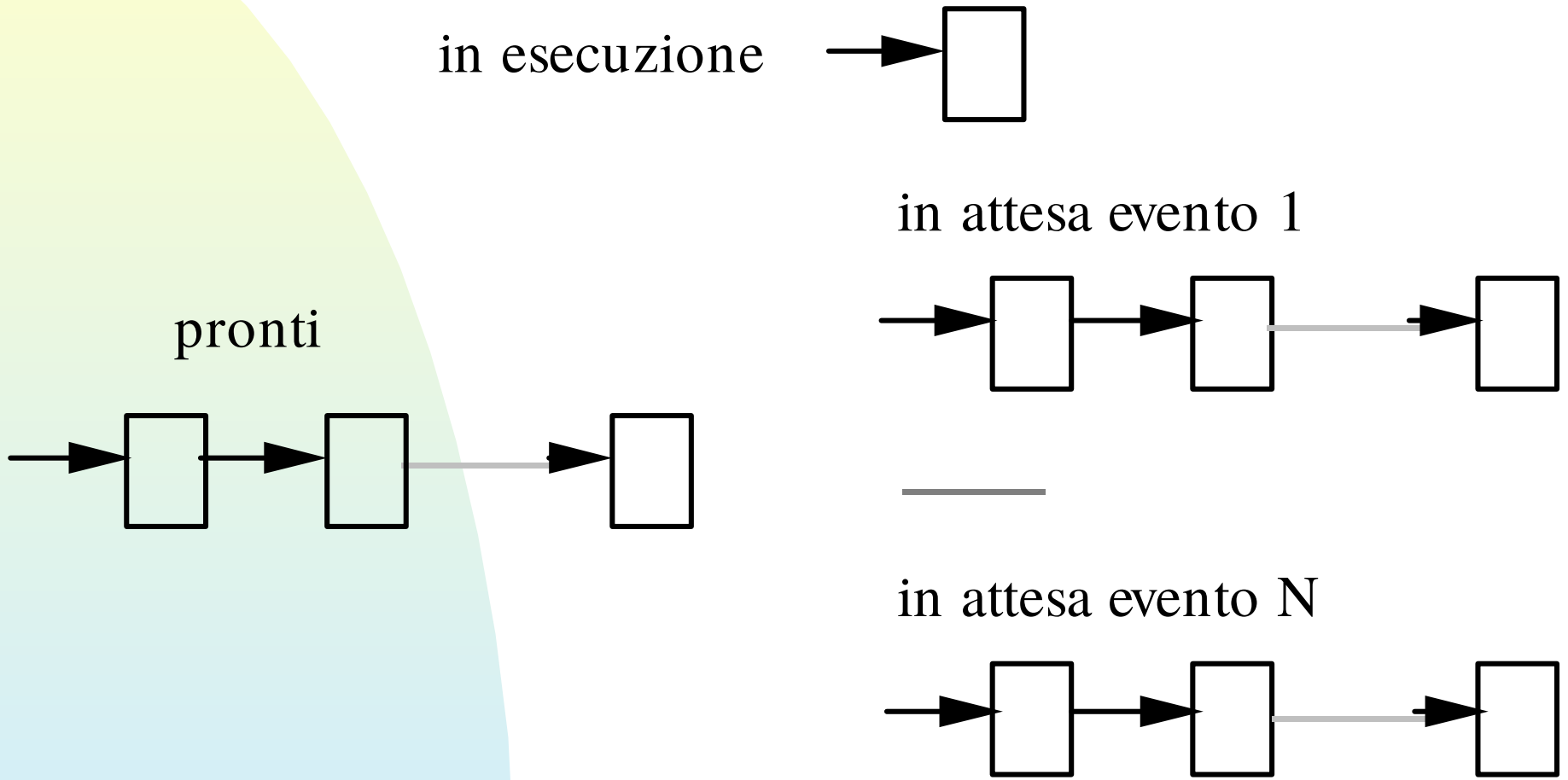
tre diverse ragioni:

- **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

Stati e transazioni lecite



Le code di processo



Famiglie di processi

Due tipologie di classificazione:

Per tipologia progettuale

- processi **compute bound**
- processi **input/output bound (I/O bound)**

Per esigenza elaborativa.

- modello **batch** (a lotti)
- modello **time-sharing** (interattivo)
- modello **real-time** (in tempo reale)

PCBT (Process Control Block Table)

Contiene tutte le informazioni sui processi attivi.

- **l'identificatore del processo:** ogni processo è identificato da un nome (di solito un numero progressivo) univoco;
- **l'identificatore dell'utente proprietario;** ogni processo ha un proprietario e questa informazione è particolarmente importante nel caso di sistemi multi-utente in quanto permette di realizzare meccanismi di protezione,
- lo **stato del processo;**
- il **valore del registro Program Counter**, questo permette di ricordare a che punto è arrivato il processo nel corso dell'elaborazione;
- il ***contenuto di tutti i registri*** (torneremo tra breve sull'utilità di queste informazioni);
- informazioni sui *file* e sulle *risorse hardware* attualmente in uso;
- informazioni sull'utilizzo della *memoria centrale* e della *memoria secondaria*;
- informazioni per lo *scheduling*

Context switch

- Il S.O. salva il contenuto di tutti i registri del processore nel descrittore della tabella dei processi del processo P1 che è stato sospeso;

- Seleziona un processo P2 pronto per l'esecuzione,

- Copia all'interno dei registri del processore i valori dei registri salvati nel descrittore del processo P2, in modo da ripristinare lo stato in cui P2 si trovava al momento della sospensione.

-Se P2 non era mai stato eseguito in precedenza (è un processo nuovo), è sufficiente caricare nel registro PC l'indirizzo della sua prima istruzione.

Politiche di scheduling del processore (1)

Due parametri di riferimento :

- Massimizzare il grado di utilizzazione del processore, ossia fare in modo che il processore rimanga attivo per il maggior tempo possibile;

- Minimizzare il tempo di attesa dei processi o, nel caso di processi interattivi, minimizzare il tempo di risposta agli utenti.

Politiche di scheduling del processore (2)

- Massimizzare il numero di programmi che vengono eseguiti nell'unità di tempo, cioè il **throughput** del sistema, in modo tale che il processore svolga il massimo lavoro possibile

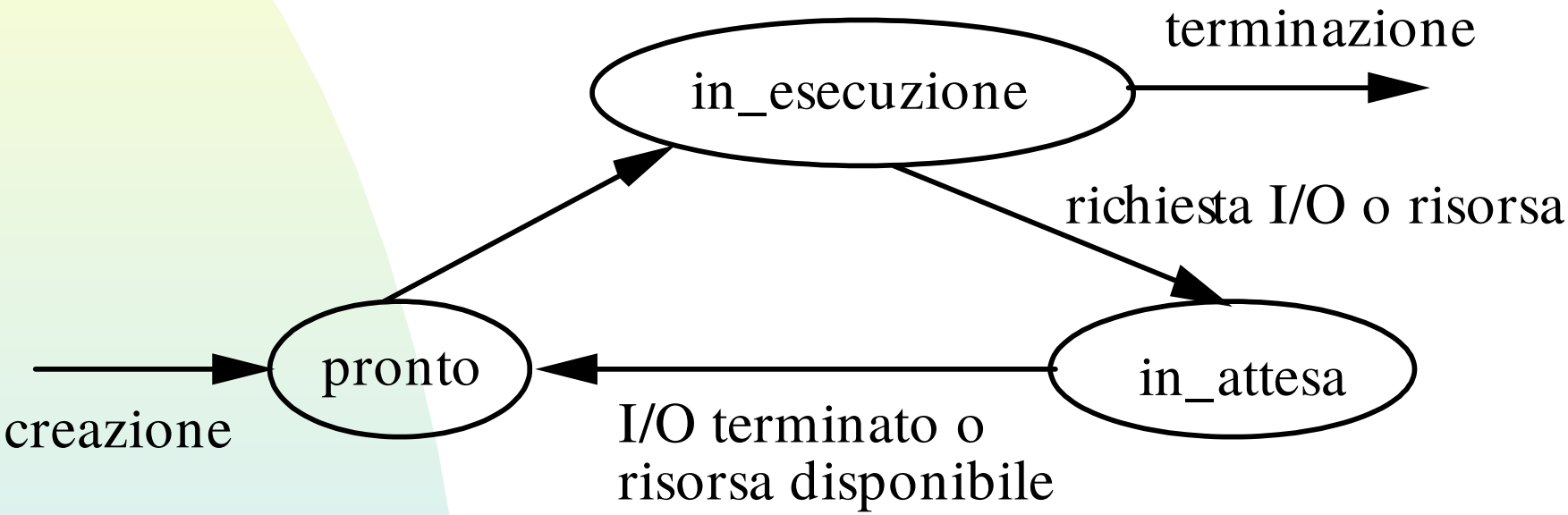
- Minimizzare il tempo di esecuzione dei processi, detto anche tempo di **turnaround**, cioè il tempo che intercorre tra l'istante in cui un processo viene creato e quello in cui esso termina

Preemption

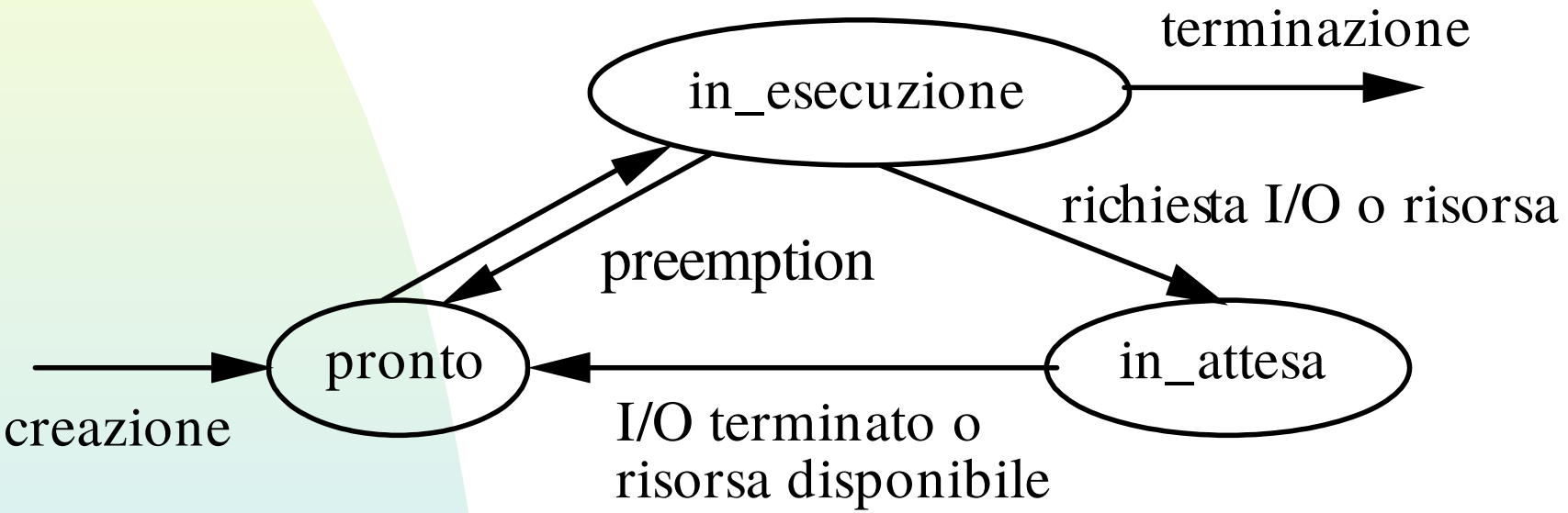
- **Politiche non-preemptive**, in cui il processo in esecuzione può essere sostituito solo se si ferma volontariamente, magari in attesa di un'operazione di I/O;

- **Politiche preemptive**, in cui il sistema operativo può decidere di bloccare un processo per mandarne in esecuzione un altro.

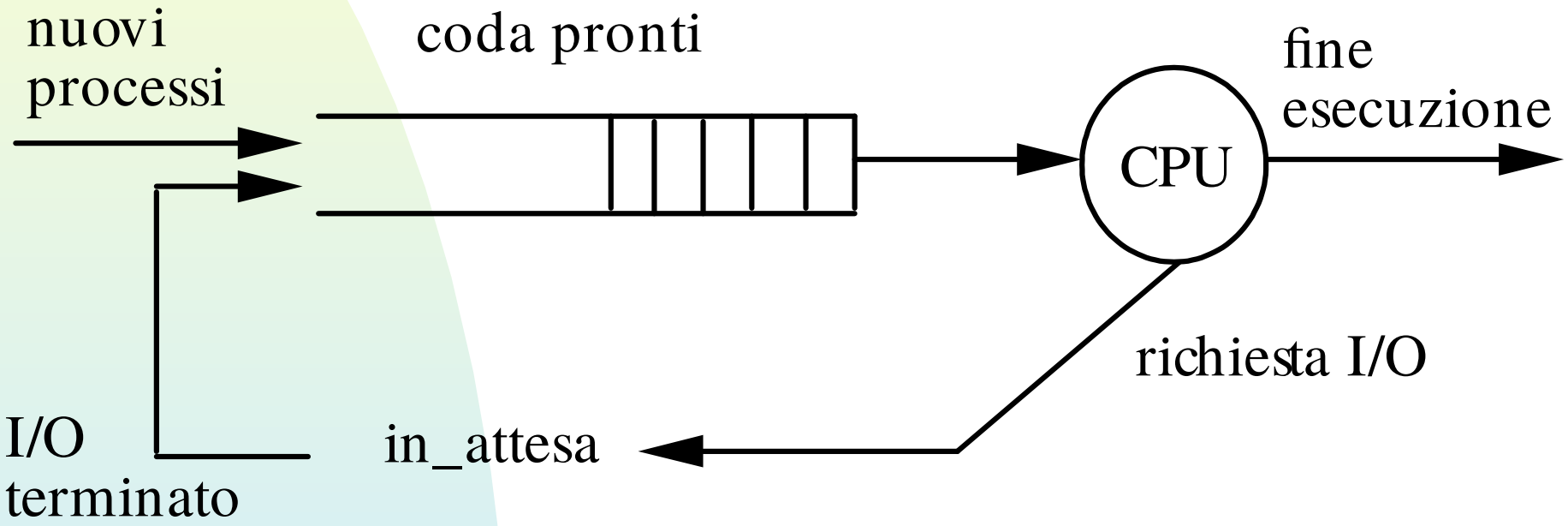
*Gli stati **non-preemptive***



Gli stati nel ***preemptive***



Politica di scheduling FCFS (FIFO)



First Come First Served (FCFS) o First In First Out (FIFO)

Si tratta di una politica non-preemptive in cui i processi sono eseguiti nell'ordine in cui vengono sottomessi al sistema.

La coda dei processi pronti viene gestita selezionando il prossimo processo da mandare in esecuzione dall'inizio della coda, e inserendo in fondo alla coda i processi che diventano pronti.

Quindi, ogni volta che il processore è libero, viene selezionato e mandato in esecuzione il primo processo della coda.

Shortest Job First (SJF)

È una politica non-preemptive. Lo schema è simile a quello della politica FCFS ma, invece di selezionare il primo processo della coda dei pronti, viene selezionato quello che richiede meno tempo per terminare o per fermarsi sulla prossima operazione di attesa.

Shortest Remaining Time First (SRTF)

Si tratta di una politica *preemptive* che costituisce una variante della politica SJF: si manda in esecuzione il processo più breve, ma in ogni istante esso può essere interrotto se, nel frattempo, è diventato pronto un processo che richiede meno tempo per terminare.

Il confronto deve essere fatto tra il tempo di esecuzione del nuovo processo e quello che manca al processo in esecuzione per terminare.

Selezione con priorità

Due varianti di questa politica

- Preemptive se, durante l'esecuzione di un processo P1, diventa pronto un altro processo P2, prioritario rispetto a P1, viene effettuato il cambio di contesto;
- Non-preemptive il processo rimane in esecuzione fino a quando non va in attesa, anche se diventano pronti dei processi con priorità maggiore.

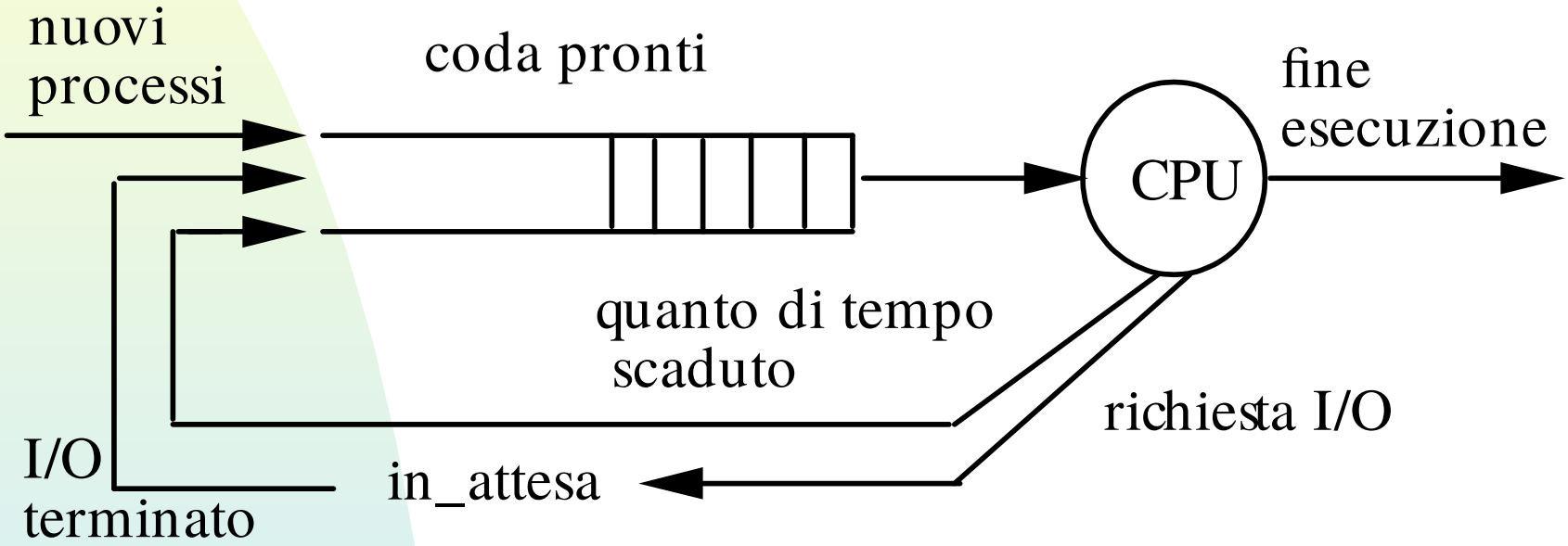
Criteri diversi:

- in base al tipo di programa: i programmi interattivi hanno una priorità più alta di quelli batch;
- in base all'utente proprietario del processo
- in base alla lunghezza del programma
- in base al tempo già trascorso in esecuzione.

Round Robin (RR)

È una politica preemptive basata sul concetto di **quanto di tempo** (**time-slice**). Ad ogni processo che viene mandato in esecuzione è assegnato lo stesso quanto di tempo prefissato - di solito circa una decina di millisecondi - entro il quale esso dovrà abbandonare lo stato di esecuzione.

Round Robin (RR)



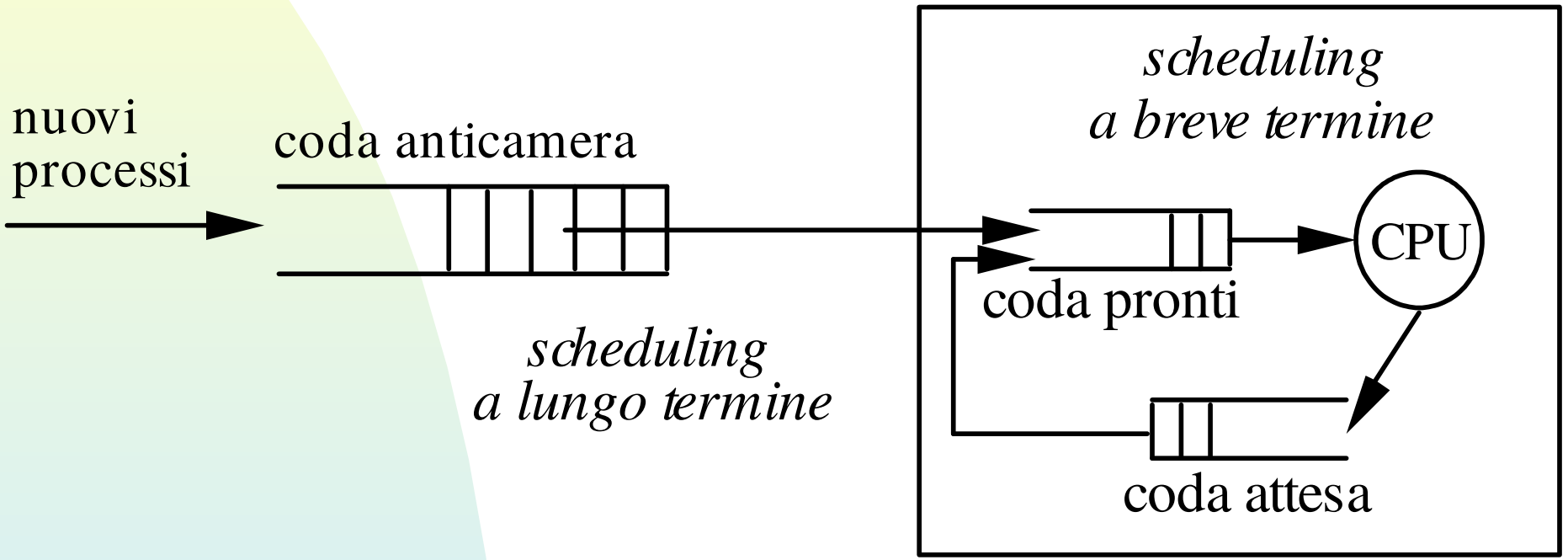
Fair e unfair

Distinzione tra politiche **eque** (**fair**) e politiche **non eque** (**unfair**).

Una politica si dice equa se garantisce che ogni processo prima o poi verrà selezionato e andrà in esecuzione; le politiche FCFS e Round Robin sono esempi di politiche eque.

Nel caso di politiche non eque, invece, c'è il rischio che un processo pronto sia costretto ad aspettare in eterno perché non arriva mai il suo turno per l'esecuzione.

Scheduling a lungo termine



INTERRUPT

- Segnali hardware
- Collegati al verificarsi di un evento
- Provocano l'esecuzione immediata di un programma del SO (vettore di interrupt) che serve a gestire l'evento
- Una CPU puo' gestire piu' interrupt
- Interrupt anche per fine di quanto di tempo (generato dal timer)

Tipologie di INTERRUPT

- Rilevazione malfunzionamento hardware
- Fine del quanto di tempo
- Arrivo di dati in input
- Rilevazione di segnali da sensori
- Rilevazione di errori software

Ogni interruzione ha la sua priorita', cosi' il processo che la gestisce puo' interrompere solo processi con priorita' inferiore

Conflitto e competizione

*La risorsa è usabile solo con il **semaforo verde;***

*Il processo che vuole usare la risorsa deve richiederla al sistema verificando **il valore del semaforo**; se è verde, lo mette a rosso e usa la risorsa; se è rosso, aspetta che torni verde e nel frattempo lascia il processore ad altri; in questo caso il processo va in stato di attesa*

*Il processo che stava occupando la risorsa ed ha finito di usarla, **rimette il semaforo a verde** e avverte chi sta aspettando della possibilità di ripartire. Il sistema operativo fa in modo che uno tra i processi in attesa della risorsa venga rimesso nello stato di pronto.*

Gestione della memoria principale

Gestione della memoria principale

Un programma per essere eseguito deve risiedere all'interno della memoria principale.

- molti processi devono essere eseguiti *contemporaneamente*, devono condividere l'uso della memoria principale.

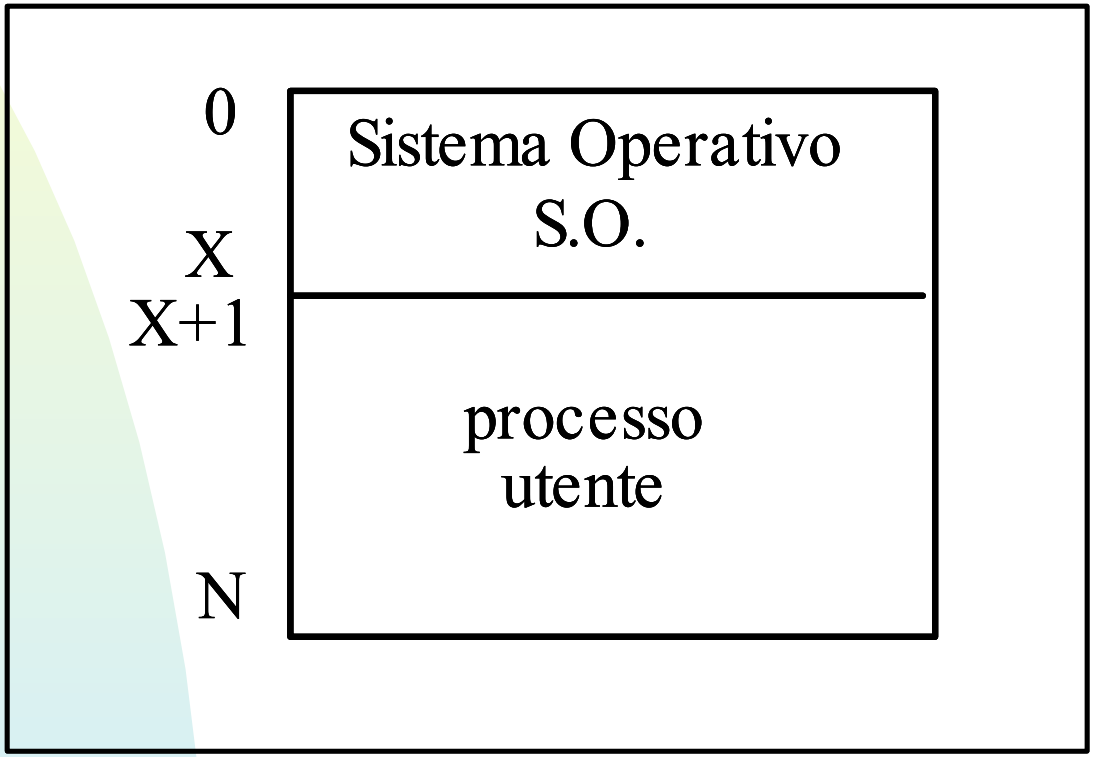
- la gestione della **memoria reale**;
- la gestione della **memoria virtuale**.

Nel primo caso si suddivide la memoria principale tra i processi; nel secondo caso si realizza una visione astratta della memoria principale che consente di estenderne le potenzialità.

Allocazione della memoria

-Quante partizioni creare e di quali dimensioni?
-Le partizioni sono create una volta per tutte al momento della configurazione del sistema o possono essere modificate a seconda delle necessità correnti?
-Come si sceglie la partizione in cui caricare l'immagine di un processo?
-Come si tiene traccia di dove sono stati caricati i processi?
-Come si proteggono i programmi tra di loro?

Memoria in sistema monotask



X = Indirizzo FENCE

Allocazione della memoria

-**Sistemi a partizioni multiple con allocazione contigua**. La memoria viene suddivisa in partizioni che devono contenere le immagini dei processi. Ciascuna partizione deve contenere interamente l'immagine di un processo

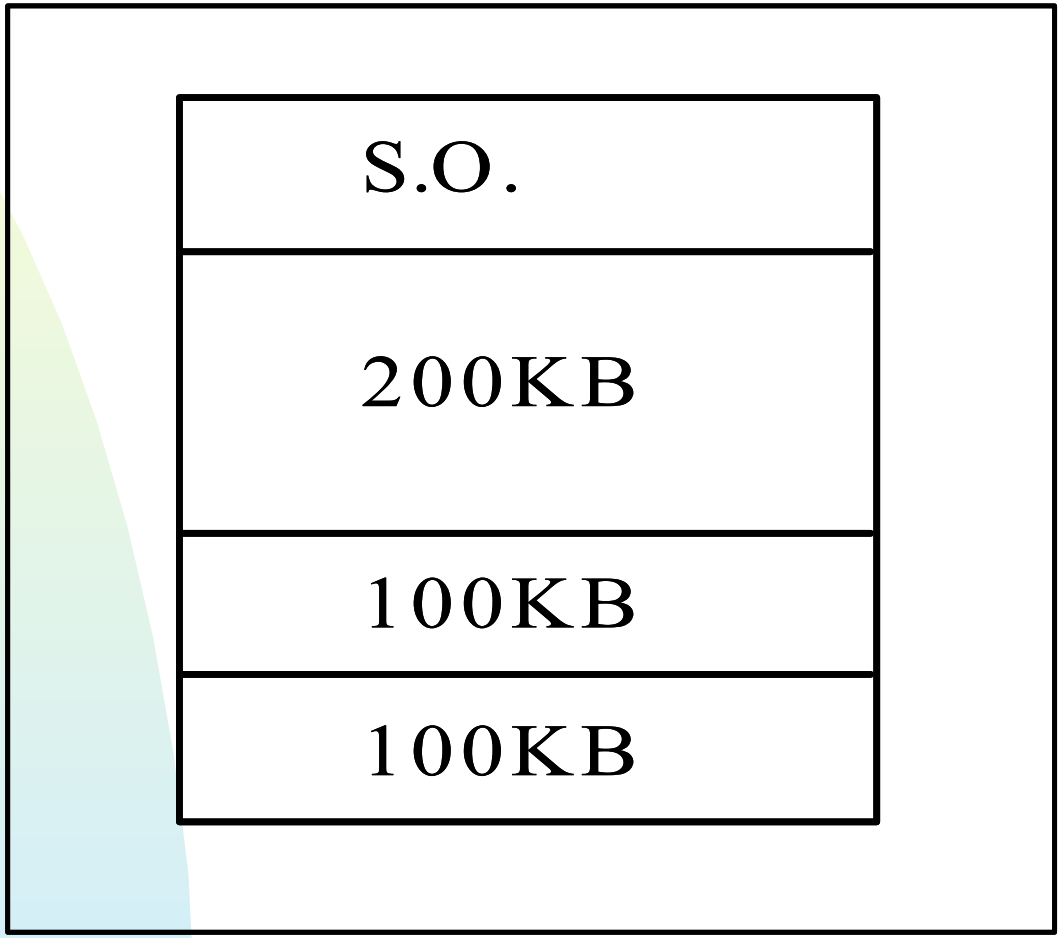
- **Sistemi ad allocazione non contigua**. Si divide l'immagine di un processo in più parti che possono essere caricate in memoria separatamente.

Allocazione della memoria

Due le tecniche :

Partizionamento Fisso
Partizionamento variabile

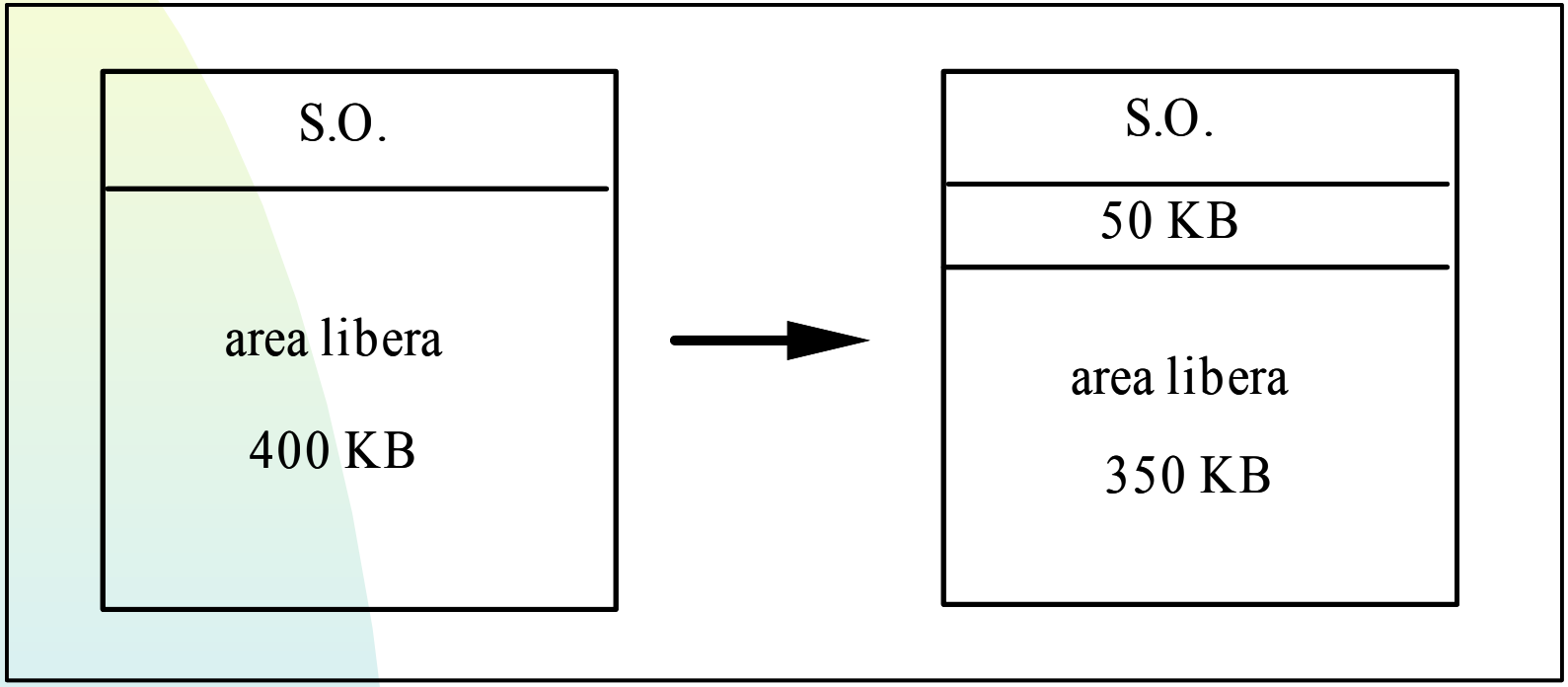
Partizionamento fisso



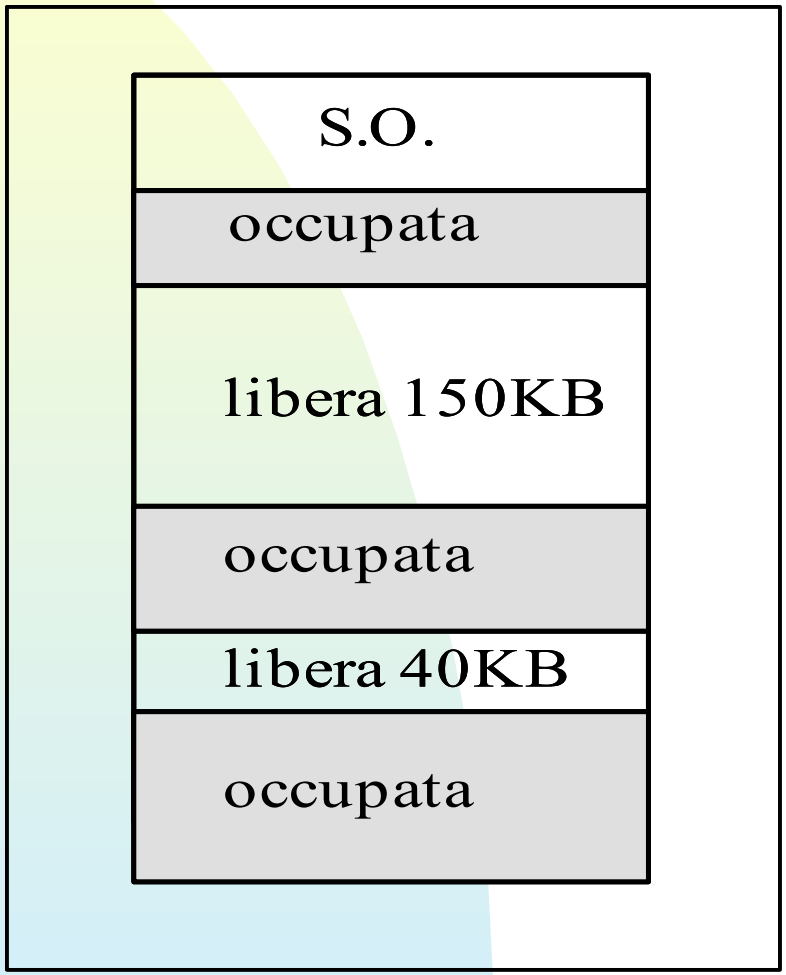
Frammentazione

- la **frammentazione interna**, ovvero lo spreco di spazio all'interno di una partizione, dovuto al fatto che la dimensione dell'immagine del processo è minore della dimensione della partizione;
- la **frammentazione esterna**, ovvero lo spreco di spazio esterno alle partizioni, che si verifica, ad esempio, quando un'intera partizione viene sprecata perché è troppo piccola per contenere l'immagine di un qualunque processo.

memoria in partizioni variabili

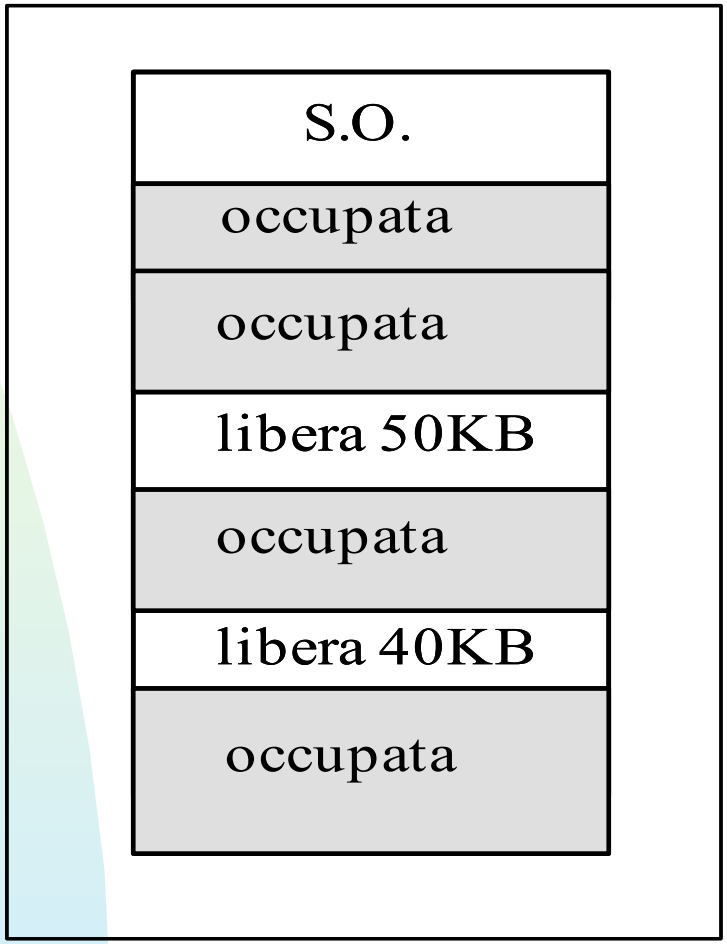


Partizioni variabili



Se devo inserire un processo da 100 KB ?

Partizioni variabili



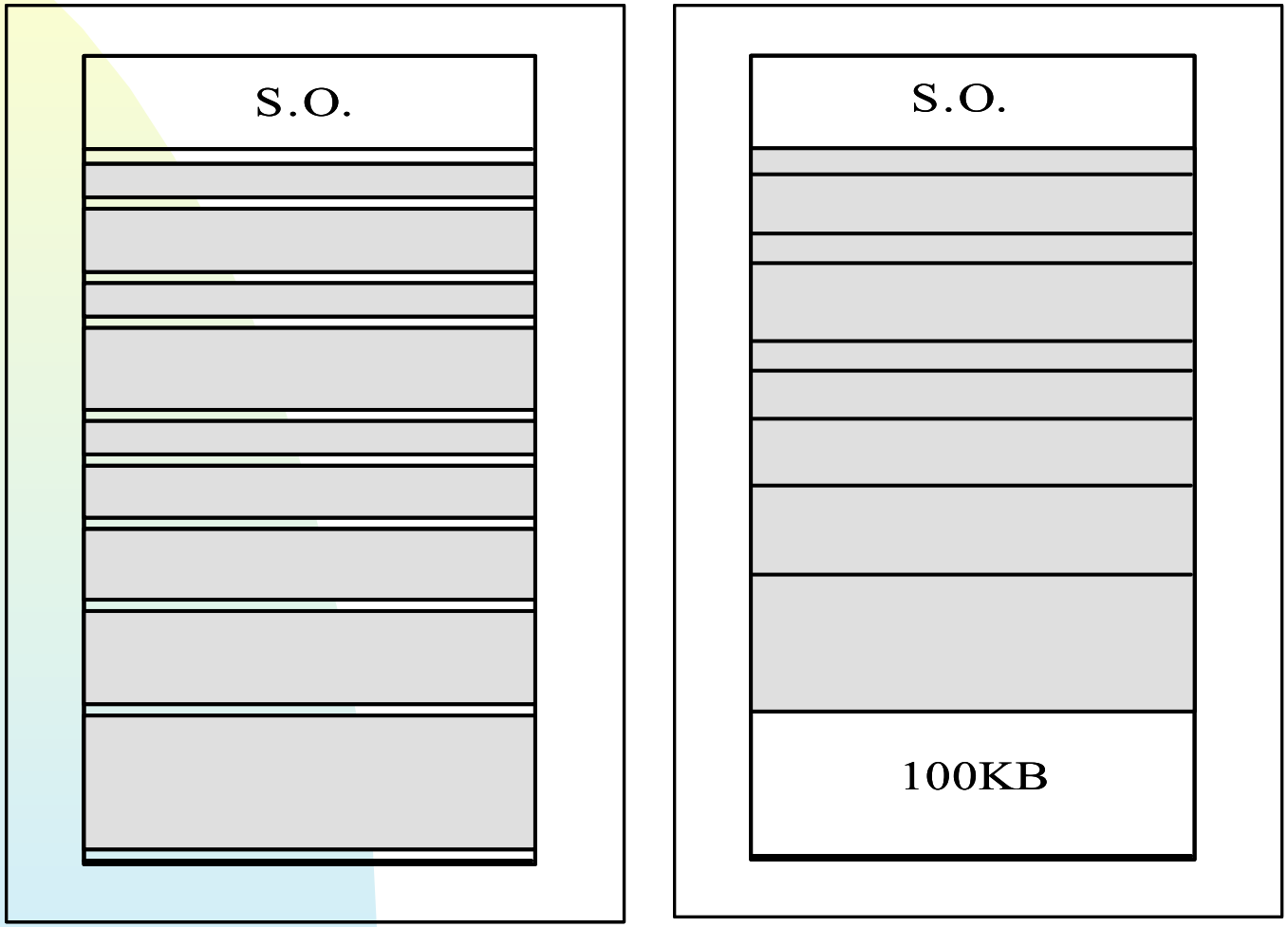
Criteri di scelta dell'allocazione

Best-fit, prevede di caricare il processo nella partizione più piccola che lo contiene; questo permette di minimizzare localmente la frammentazione esterna. Il problema di questa scelta è dovuto al fatto che gli spazi che avanzano sono piccoli e quindi difficilmente utilizzabili.

Worst-fit prevede di caricare un processo sempre nella partizione più grande; in questo modo gli spazi avanzati sono grandi ed hanno maggiore probabilità di poter essere utilizzati per altri processi.

La partizione per un processo da 30 KB verrebbe creata in quella da 150 KB, lasciando liberi 120 KB.

Compattazione



Segmentazione (all. non contigua)

Suddivisione dell'immagine di un processo in n parti: ad esempio 2 - una che contiene il codice che deve essere eseguito (programma) e l'altra che contiene i dati su cui il programma lavora.

Queste parti potrebbero essere caricate in memoria indipendentemente l'una dall'altra; si noti che, comunque, entrambe sono necessarie quando il processo va in esecuzione. Diremo allora che l'immagine del processo è stata divisa in due **segmenti** che vengono caricati in memoria separatamente e denoteremo questa tecnica con il nome di **segmentazione**.

Paginazione

La memoria viene divisa in un insieme di **blocchi** (detti **frame**), tutti della stessa dimensione (solitamente tra 1 KB e 4 KB);

Le immagini dei processi vengono suddivise in **pagine** uguali tra loro e con la stessa dimensione dei blocchi di memoria;

Le pagine che costituiscono l'immagine di un processo vengono caricate in memoria in modo sparso;

Il sistema operativo tiene traccia, nella tabella dei processi, di dove sono caricate le pagine di ciascun processo.

Gestione della memoria virtuale

Visione astratta della memoria che prende il nome di **memoria virtuale** e i programmi di sistema che la realizzano prendono il nome di **gestori della memoria virtuale**.

Due le tecniche di gestione non sono mutuamente esclusive:

Primo caso: le immagini dei processi vengono alternate all'interno della memoria centrale, nello stesso modo in cui il processore viene alternato tra i processi;

Secondo caso le immagini dei processi vengono caricate nella memoria centrale *a pezzi.*

Swapping

Caricare in memoria principale l'intera immagine di un processo al momento in cui questa è necessaria.

- Quando si crea un processo, la sua immagine viene posta in memoria secondaria; non appena vi sarà uno spazio libero in memoria principale, l'immagine verrà caricata e il processo messo nello stato di pronto.

- Quando un processo in esecuzione si ferma, oltre a perdere l'uso del processore può anche essere tolto dalla memoria e messo nello stato di attesa. Lo spazio lasciato libero viene preso da un altro processo che può così diventare pronto.

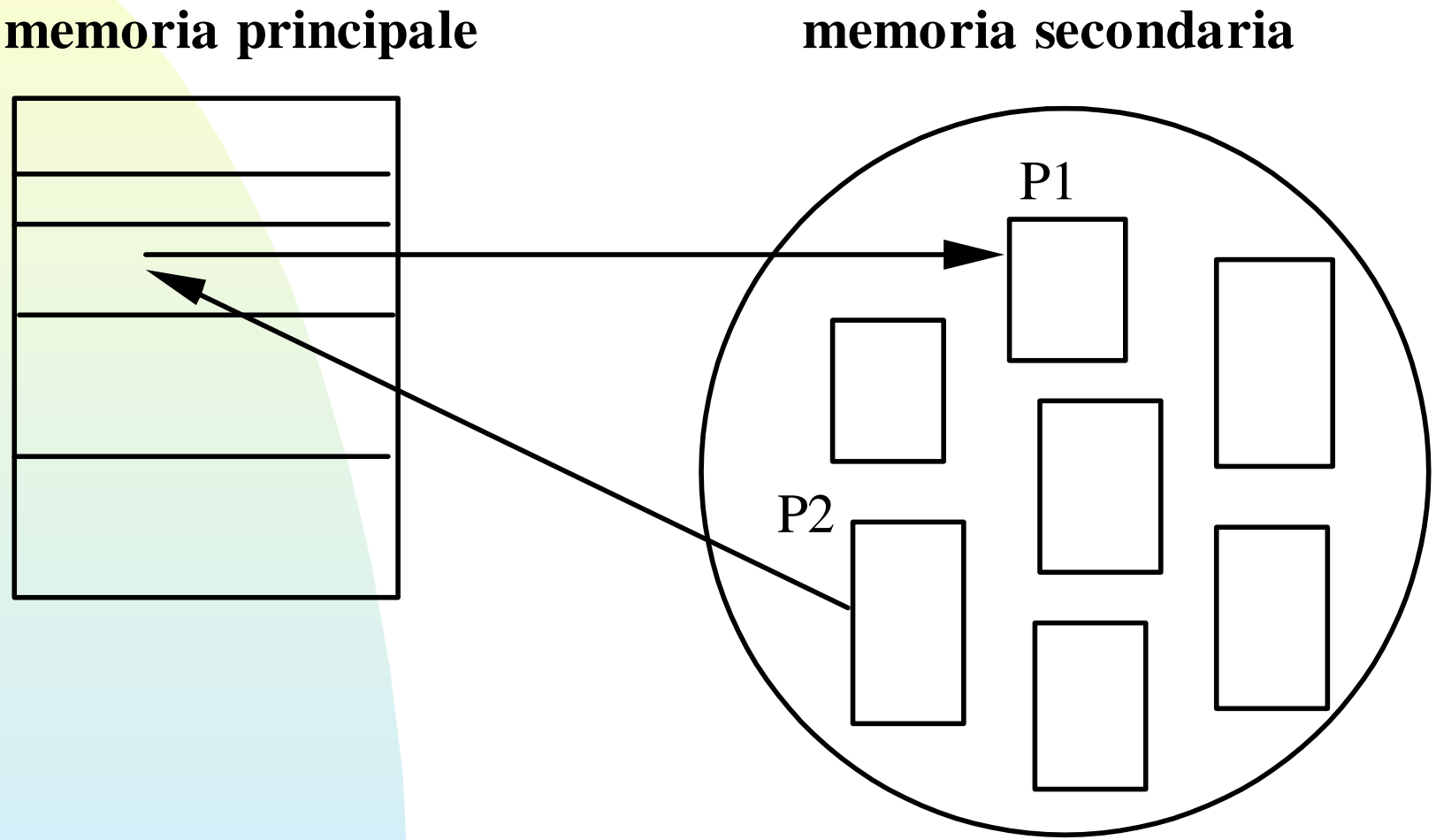
- Quando un processo esce dallo stato di attesa non può essere immediatamente messo nello stato di pronto perché deve attendere che si liberi dello spazio per essere caricato in memoria.

Swapping

La tecnica dello swapping può essere realizzata sia con partizioni fisse che con partizioni variabili.

Non sempre si caricano i processi nella stessa partizione: se non si impone questo vincolo si parla di **swapping con rilocazione**

Swapping



Demand paging

Si possono tenere in memoria solo alcune parti delle immagini dei processi, purché si sia sempre in grado di caricare le altre parti quando servono.

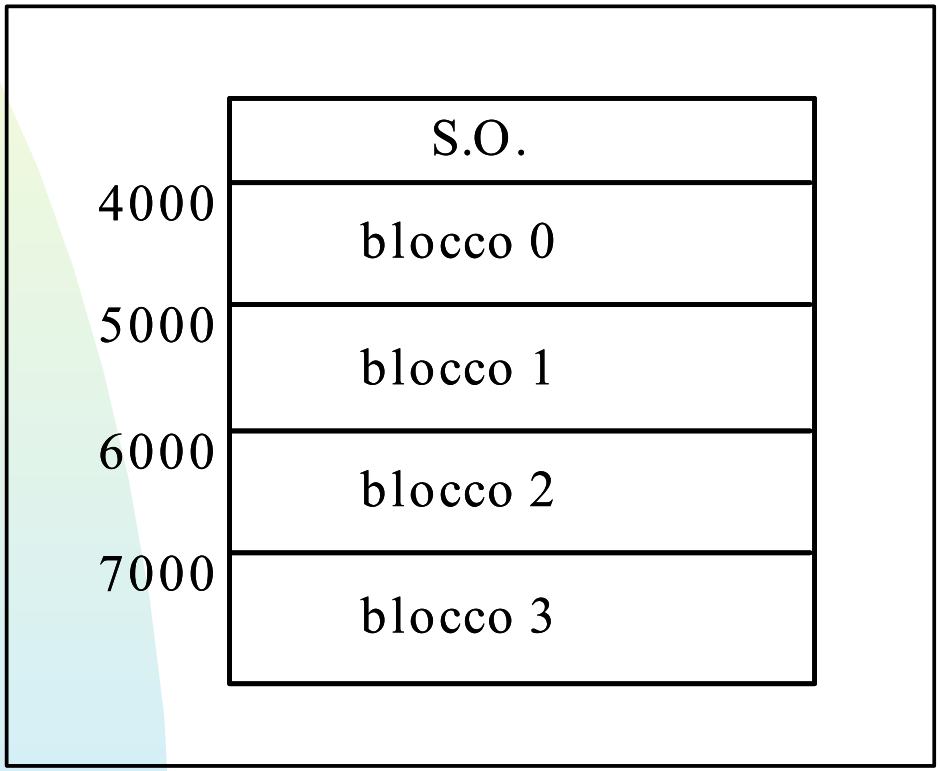
Questa è proprio l'idea che sta alla base delle tecniche di gestione a pezzi e, in particolare, della tecnica di **paginazione a richiesta (demand paging)**

Frame table

Il sistema operativo tiene traccia dei blocchi liberi e di quelli occupati in una tabella detta **tabella dei blocchi (frame table)**.

Frame table

Suddivisione della memoria principale in blocchi tutti uguali



Frame table

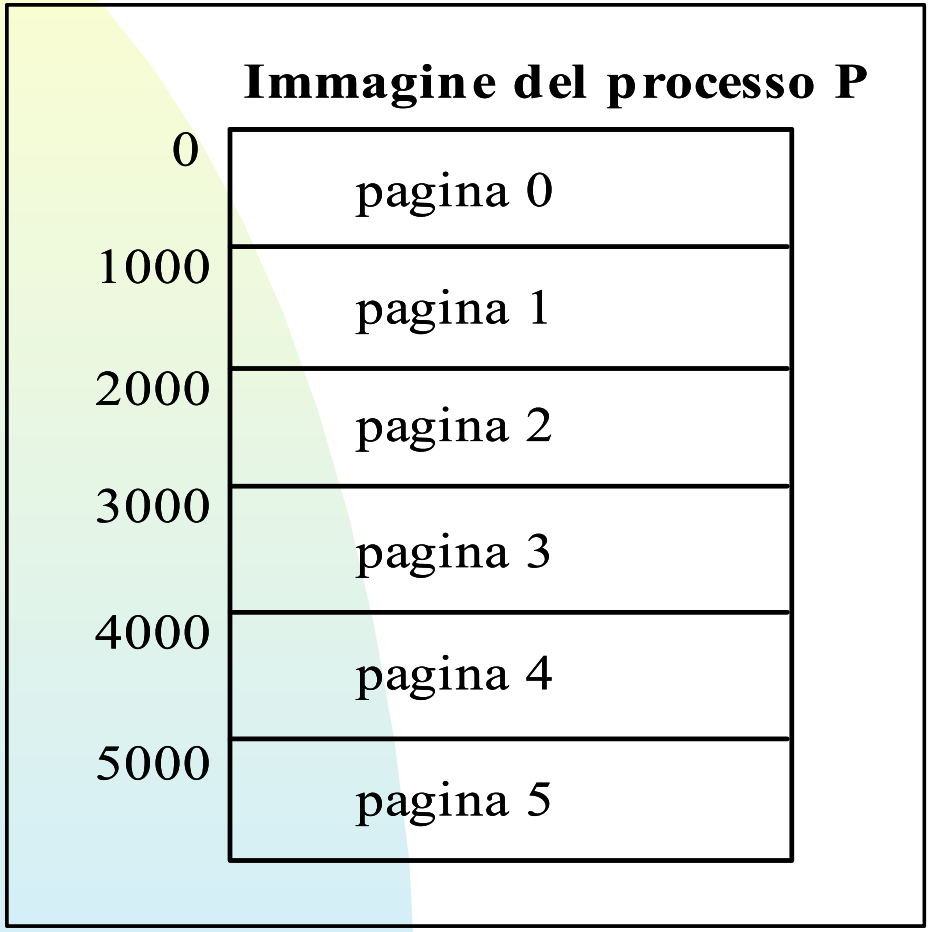
Ogni cella della memoria virtuale del processo P avrà un indirizzo detto **indirizzo virtuale** e le istruzioni del codice di P faranno riferimento agli indirizzi virtuali.

Possiamo pensare di rappresentare gli indirizzi virtuali mediante una coppia di numeri:

- il **numero della pagina** in cui si trova un'informazione;
- l'**indirizzo relativo** all'interno di tale pagina (**offset**).

La coppia (4,130) identifica blocco e indirizzo

Frame table



Qual è l'indirizzo assoluto di (3,45)

Page fault

Non appena l'esecuzione fa riferimento ad un indirizzo che si trova in un'altra pagina si ha un **page fault**, cioè un indirizzamento ad una pagina che non è correntemente caricata in memoria.

Il processo P viene bloccato e messo nello stato di attesa.

Non appena la pagina richiesta viene caricata in memoria il processo P può ritornare nello stato di pronto.

Page table (tabella delle pagine)

Esiste una tabella delle pagine **per ogni processo**,

La tabella ha tante righe quante sono le pagine del processo.

- Ogni riga contiene i campi *In Mem?*, *Blocco* e *Ind Mem Sec.*

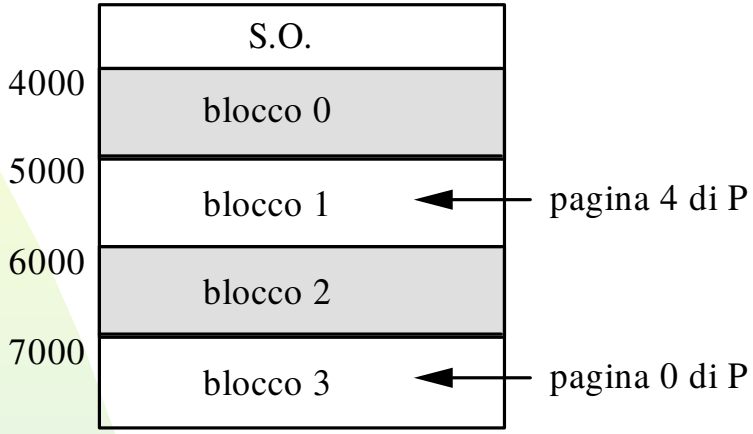
- Se la pagina si trova in memoria principale il campo *In Mem?* contiene il valore SI (espresso in codifica binaria), e il campo *Blocco* contiene l'indirizzo del blocco in cui è caricata.

- Se invece la pagina non è in memoria principale il campo *In Mem?* contiene il valore NO e il campo *Ind Mem Sec* il suo indirizzo in memoria secondaria.

Page table

pagina	In Mem?	Blocco	Ind Mem Sec
0	SI	7000	α
1	NO		β
2	NO		γ
3	NO		δ
4	NO		ε
5	NO		ϕ

Frame/page table



pagina	In Mem?	Blocco	Ind Mem Sec
0	SI	7000	α
1	NO		β
2	NO		γ
3	NO		δ
4	SI	5000	ε
5	NO		ϕ

Processo di caricamento pagina

- **si guarda se la pagina è già in memoria principale** (campo InMem? della tabella delle pagine);

- **se SI**, si prende l'indirizzo del blocco e gli si somma l'indirizzo relativo all'interno della pagina;

- **se NO**, la pagina deve essere caricata e si procede come segue:
  - viene richiesto il caricamento della pagina dal disco in memoria principale;
  - il processo P va in attesa; il sistema esegue quindi uno switch di contesto passando ad eseguire un altro processo;
  - P tornerà pronto quando la pagina sarà stata caricata.

Trashing

Supponiamo venga scelta una pagina che servirà entro breve; appena un processo farà riferimento a quella pagina, si genererà una nuova richiesta di caricamento per cui la pagina appena tolta dalla memoria dovrà subito esservi rimessa.