GESTIONE DELLA MEMORIA

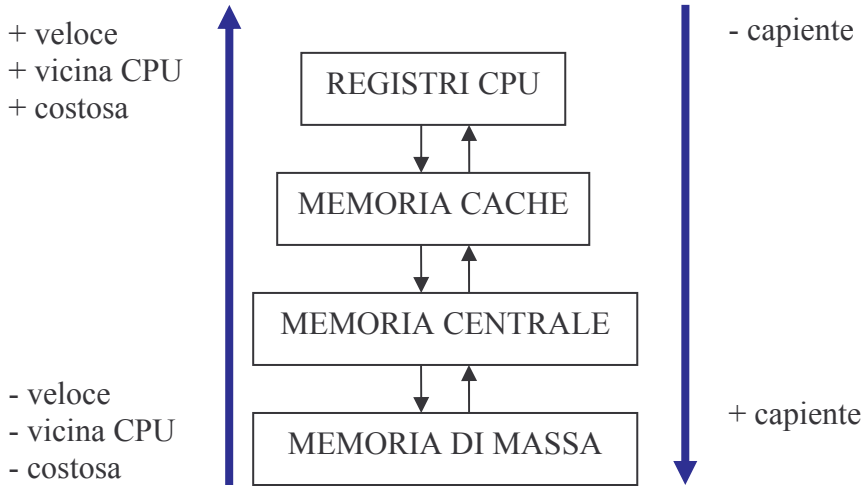
La gestione della memoria da parte di un Sistema Operativo è strettamente legata alla struttura Hardware dell'elaboratore considerato.

Le **funzioni** a cui questo modulo deve sottostare sono quattro:

- Tenete aggiornato lo stato di ogni locazione della memoria centrale (sapere, ad esempio, se è libera oppure occupata).
- Decidere, in base a determinate politiche di allocazione, a quale processo debba essere assegnata la memoria, con che priorità e per quanto tempo.
- Selezionare le locazioni da assegnare ad un processo e provvedere alla memorizzazione reale delle informazioni associate.
- Gestire la politica del recupero do una certa area di memoria, decidendo a quale processo togliere la risorsa per assegnarlo ad un altro con maggiore priorità.

È utile fare la distinzione tra **Spazio Fisico**, cioè l'insieme degli indirizzi delle locazioni di memoria centrale accessibili a livello della macchina assembler (tutti gli indirizzi della *memoria fisica*) e **Spazio Logico**, un insieme di indirizzi più astratto su cui opera un certo processo (*memoria logica*). In tal modo, ogni processore non genera **indirizzi fisici**, ma **indirizzi logici**, anche se occorrerà far corrispondere ad ogni indirizzo logico un determinato indirizzo fisico. Questa corrispondenza è effettuata dalla **MMU (Memory Management Unit)**, che si occupa della gestione delle interazioni tra memoria centrale e memoria cache.

In tutti i computer di ultima generazione si riconosce un'organizzazione gerarchica della memoria: è possibile individuare la memoria a più alto livello gerarchico, cioè quella "più vicina" alla CPU, più veloce (in termini di accesso), più costosa, ma meno capiente. Ci sono poi memorie intermedie sino ad arrivare al livello più basso, quella "più lontana" dalla CPU, più lenta, meno costosa, ma più capiente.



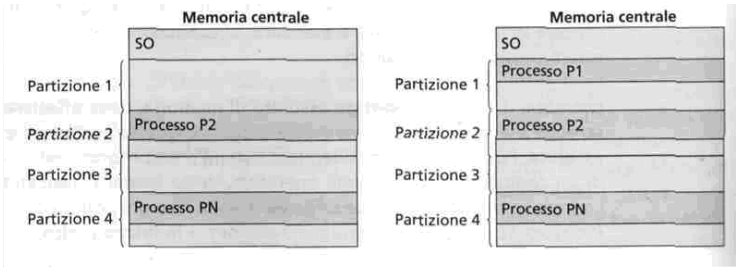
I vari modi in cui è possibile organizzare la memoria sono:

- A partizione (Fisse o Variabili)
- A memori virtuale (Paginata o Segmentata)

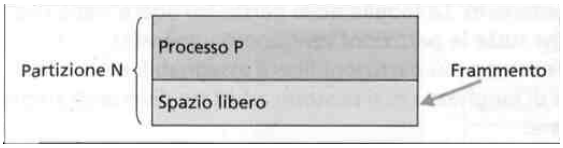
ORGANIZZAZIONE A PARTIZIONI

Possiamo distinguere due divisioni:

- **Partizioni fisse** o **statiche.** Si prevede in fase di configurazione del sistema un numero prefissato di partizioni. Si suddivide la memoria centrale in un numero prefissato di porzioni, di uguale dimensione oppure di dimensioni diverse e a ogni processori assegnata, in modo esclusivo, una partizione.



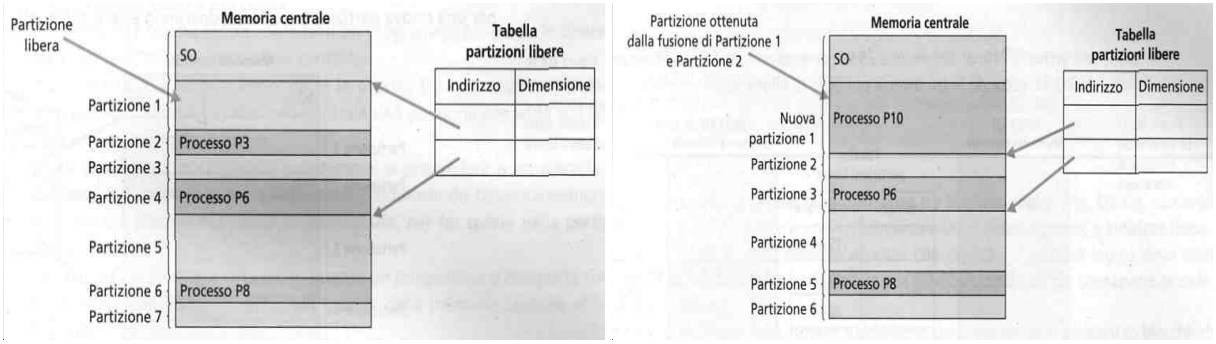
In questo tipo di organizzazione, lo spazio della partizione non utilizzato dal processo non può essere utilizzato da un altro processo (un solo processo, infatti, può essere allocato in una partizione), generando così un problema di spreco di spazio all'interno delle partizioni. Allo stesso tempo i processi che necessitano di più spazio di quanto ciascuna partizione offre non possono essere eseguiti.



Nel caso di partizioni di dimensioni diverse, il sistema operativo gestisce una tabella con lo stato di ogni partizione e con le sue dimensioni. A ogni processo in coda è assegnata una partizione in grado dì accogliere il programma e i dati corrispondenti. Se un processo richiede molta memoria e non esiste una partizione di adeguata dimensione disponibile (cioè non ancora occupata da un processo in esecuzione), dovrà attendere in coda fino a quando si sia liberata una partizione di dimensioni opportune. Il miglioramento rispetto al caso precedente non è però decisivo.

- **Partizioni variabili** o **dinamiche**. Le partizioni non sono determinate a priori, ma dipendono dai processi in attesa di esecuzione. Nell'organizzazione a partizioni dinamiche non esistono partizioni di dimensione prefissata; nuove partizioni vengono create in base alla richiesta dei processi, prelevandole da una tabella delle partizioni di memoria libere. Se esistono due partizioni libere attigue esse vengono fuse in un'unica partizione libera.

  La situazione è schematizzata nelle figure seguenti, che mostrano una memoria, composta inizialmente da sette partizioni, nella quale si vede ciò che accade prima e dopo l'arrivo di un nuovo processo P10 (eseguito alla terminazione di P3). Si noti che la tabella delle partizioni libere viene aggiornata: infatti tutte le partizioni vengono rinumerate.



Supponendo che esistano più partizioni libere assegnabili a un nuovo processo che richiede una partizione di lunghezza *n*, si possono adottare diverse strategie per scegliere quale partizione occupare:

- **strategia first fit**. Consiste nell'assegnare la prima partizione libera (presente nella tabella delle partizioni libere) di dimensione maggiore o uguale a n;
- **strategia best fit**. Consiste nell'assegnare la partizione di dimensione più vicina ad n (ovvero quella con meno spazio libero residuo);
- **strategia worst fit**. Consiste nel l'assegnare la partizione che lascia il maggiore spazio residuo.

ORGANIZZAZIONE A MEMORIA VIRTUALE

Finora abbiamo implicitamente supposto che, quando carichiamo un programma in memoria, venga caricalo tutto il suo codice.
Possiamo ora far cadere questa ipotesi. Se infatti un programma superasse le dimensioni della memoria centrale, sarebbe impossibile caricarlo.
Ricorrendo alla tecnica di funzionamento, detta **in overlay** (*sovrapposizione*), si divide il programma in porzioni, si carica in memoria centrale una porzione per volta e si lascia il resto delle porzioni in memoria di massa.
Quando servirà un'ulteriore porzione del programma, si provvedere a recuperarla dalla memoria di massa e a caricarla in memoria centrale, rimuo-vendo temporaneamente da essa una porzione del programma stesso non necessaria, per far spazio nella partizione destinata a quel processo. Si parla in questo caso di **memoria virtuale,** in quanto un programma si comporta come se avesse a disposizione una memoria molto più grande della memoria centrale effettivamente disponibile.

La memoria virtuale si basa sulla differenza tra **indirizzo logico**, contenuto all'interno di un programma come riferimento a un qualsiasi oggetto, e **indirizzo fisico** di memoria, in cui è effettivamente allocato tale oggetto. L'indirizzo logico deve essere tradotto in indirizzo fisico per accedere alla memoria centrale; tale operazione prende il nome di **mapping**.

Esistono due differenti tecniche di suddivisione del programma in porzioni (o **blocchi**) che permettono di mettere in corrispondenza l'indirizzo logico con quello fisico:

- ***tecnica della paginazione*.** Attraverso questa tecnica un programma viene suddiviso logicamente in blocchi di uguali dimensioni detti *pagine logiche* e contemporaneamente la memoria centrale viene divisa in blocchi delle stesse dimensioni, detti *pagine fisiche*). Per riferirci ad un oggetto all'interno del programma (cioè per definire l'indirizzo logico di un oggetto X), indicheremo il numero di pagina e l'indirizzo all'interno della pagina (Offset).

- ***tecnica della segmentazione***. presuppone una suddivisione logica, effettuata a priori, del programma che andrà in esecuzione. Tale suddivisione prevede l'individuazione di unità di dimensioni diverse, chiamate **segmenti**, sulla base di criteri logici stabiliti anche da parte del programmatore. Un programma può essere suddiviso in un segmento contenente il codice relativo alle procedure di uso più frequente, in un altro contenente le variabili globali e cosi via. Per riferirci a un oggetto all'interno del programma occorre conoscere il numero di segmento in cui è contenuto e l'indirizzo all'interno del segmento, detto Offset.