

Gestione della Memoria

- ❖ **Memoria Virtuale**
- ❖ **Paginazione**
- ❖ **Segmentazione**

Gestione della memoria

□ **Attività del S.O. in connessione con la gestione della memoria**

❖ **Rilocazione**

- Il S.O. deve associare agli indirizzi logici di un programma gli indirizzi fisici

❖ **Allocazione**

- Il S.O. deve tener traccia di quali parti della memoria sono correntemente usate e da quali processi

❖ **Protezione e Condivisione**

- Il S.O. deve proteggere ogni processo da interferenze indesiderate da parte di altri processi e deve permettere a diversi processi di accedere alla stessa porzione di memoria

❖ **Gestione degli scambi**

- Il S.O. deve gestire gli scambi di informazioni tra memoria centrale e disco quando la prima non è abbastanza grande per contenere i processi (**swapping**)

Gestione della memoria

Rilocazione

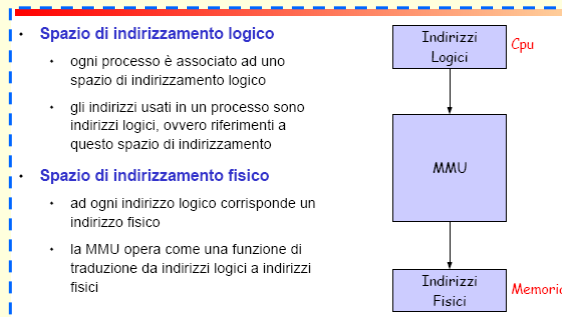
- ❖ **Associazione (Mapping)** tra lo spazio di indirizzi logici del programma e lo spazio di indirizzi fisici della memoria



- ❖ I metodi di associazione degli indirizzi nelle fasi di compilazione e di caricamento di un programma producono indirizzi logici e fisici identici
 - Il processo non può essere ricaricato in posizioni diverse della memoria

Gestione della memoria

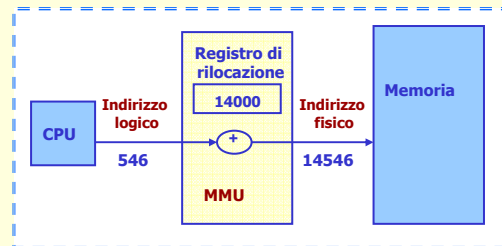
- I metodi di associazione degli indirizzi nella fase di esecuzione producono indirizzi logici e fisici differenti
- L'associazione degli indirizzi logici agli indirizzi fisici è svolta da un dispositivo detto unità di gestione della memoria (**MMU-Memory Management Unit**)
 - La possibilità di caricare un programma a partire da una posizione arbitraria della memoria presuppone che sul codice venga effettuata una rilocazione di tutti gli indirizzi logici presenti nel programma in indirizzi fisici



Gestione della memoria

❖ Rilocalizzazione statica

- Rilocalizzazione effettuata mediante un registro base, **registro di rilocalizzazione**, che contiene l'indirizzo di caricamento del programma
- Quando un processo utente genera un indirizzo, si somma a tale indirizzo il valore contenuto nel registro di rilocalizzazione
 - Il sistema operativo MS-DOS carica ed esegue processi impiegando quattro registri di rilocalizzazione



❖ Rilocalizzazione dinamica

- Le procedure sono caricate solo quando vengono richiamate
- Il collegamento dinamico è necessario nel caso di multiprogrammazione

Gestione della memoria

□ Allocazione della memoria

❖ Allocazione statica

- Un processo conserva la propria area di memoria dal caricamento alla terminazione

❖ Allocazione dinamica

- Un processo può essere spostato all'interno della memoria

❖ Strategie, attuate dal S.O., per consentire un uso ottimale della memoria disponibile garantendo:

- spazio di indirizzamento virtuale assegnato ai vari processi
- meccanismi di protezione che tutelano lo spazio di lavoro assegnato ad ogni processo

Sistemi multiprogrammati o multiutenti

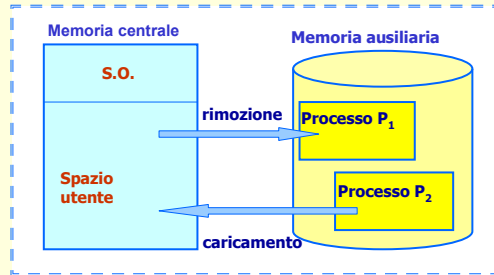
Incompatibilità tra la dimensione della memoria fisica e lo spazio complessivo richiesto dai programmi che devono essere eseguiti in modo concorrente e dai relativi dati

Gestione della memoria

❑ Swapping

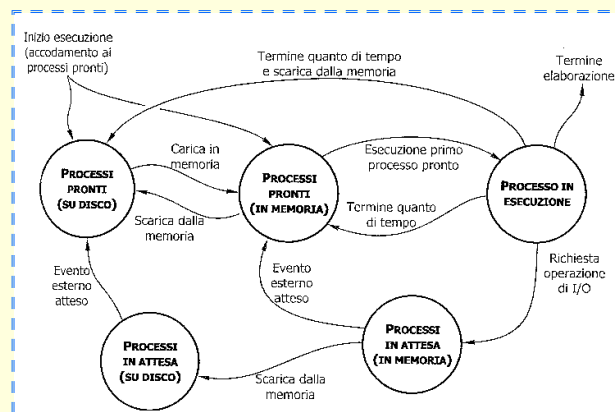
❖ Avvicendamento dei processi

- Il S.O. mantiene in memoria solo alcuni processi, mentre altri processi in stato di pronto o attesa sono trasferiti su memoria ausiliaria



Gestione della memoria

❖ Swapping e stati dei processi



Gestione della memoria

□ Partizionamento della memoria

- ❖ La memoria centrale è normalmente suddivisa in due **partizioni**, una per il sistema operativo e l'altra per i processi utente
 - Più processi devono risiedere contemporaneamente nella memoria centrale
 - La memoria disponibile per un programma e i suoi dati può essere insufficiente
 - Un processo può essere diviso in parti non necessariamente localizzate in memoria in modo contiguo
 - Tutte le parti di un processo non necessariamente localizzate in memoria durante l'esecuzione

□ Memoria Virtuale

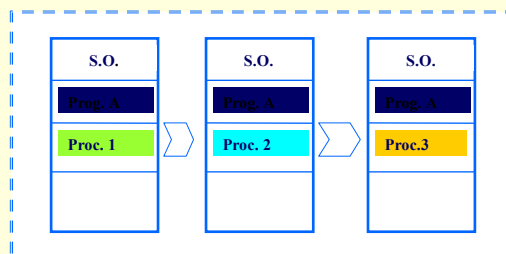
- ❖ Simulazione di una memoria principale di elevata capacità usando una memoria ausiliaria (disco fisso)
 - Le partizioni di un processo non sono tutte in memoria
 - Le partizioni necessarie non presenti in memoria sono caricate successivamente in modo automatico
 - Il programmatore ha l'illusione di disporre di uno spazio di indirizzi fisici di dimensioni pari a quelle dello spazio di indirizzi logici

Gestione della memoria

□ Tecniche di partizionamento

1. Overlay

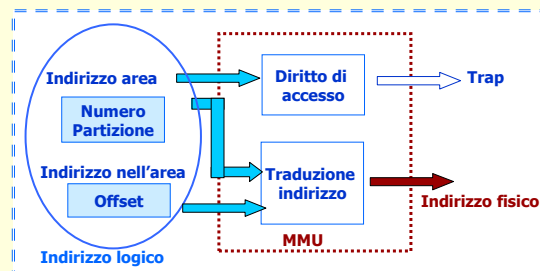
- **Partizionamento del programma in parti logiche**
 - Il programma è suddiviso in parti logiche a sé stanti, ognuna delle quali è caricata in memoria al momento del suo utilizzo e poi rilasciata su disposizione del programmatore



Gestione della memoria

2. Paginazione e Segmentazione

- **Partizionamento del programma e della memoria in aree**
- Supporti necessari affinché il **partizionamento della memoria** avvenga in modo automatico
- **Hardware** che deve supportare tecniche di partizionamento della memoria, quali paginazione e/o segmentazione
- **S.O.** che deve essere abile a gestire il movimento di pagine e/o segmenti fra memoria ausiliaria e memoria principale



Gestione della memoria

□ Partizionamento della memoria

❖ Partizioni fisse

- La memoria è divisa in partizioni (**pagine**) la cui dimensione è fissata a priori
- **Frammentazione interna** dello spazio di memoria, ovvero spazio di memoria sprecato all'interno di alcune pagine

❖ Partizioni variabili

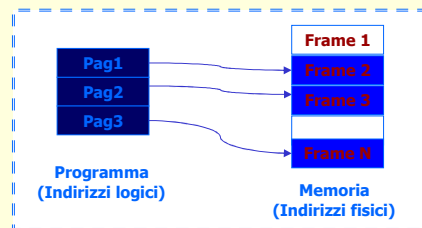
- La memoria è divisa in partizioni (**segmenti**) la cui dimensione non è fissata a priori ma in base alle richieste dei processi
- **Frammentazione esterna** dello spazio di memoria, ovvero spazio di memoria sprecato tra i segmenti (all'esterno dei segmenti)

Paginazione

□ Paginazione

❖ Metodo di gestione della memoria che permette che lo **spazio degli indirizzi fisici di un processo non sia contiguo**

- La memoria (spazio di indirizzi fisici) è divisa in aree di uguale dimensione dette **'page frame'**
- Il programma (spazio di indirizzi logici) è diviso in **'pagine'** della stessa dimensione di una page frame, dal compilatore o dal gestore della memoria
- Ogni pagina del programma può essere contenuta in un page frame



Paginazione

❖ Page table

- Il S.O. fa uso, per ogni processo, di una **tabella delle pagine** che mostra in che frame si trova la pagina e di una lista dei frame liberi (**free frame list**)

0	0	0	—	0	7	0	4	13
1	1	1	—	1	8	1	5	14
2	2	2	—	2	9	2	6	
3	3			3	10	3	11	
						4	12	

Process A page table Process B page table Process C page table Process D page table Free frame list

Struttura della page table

Page	Flag	Page frame	Start address su disco	
1	1	1	3C4T8S	100
2	0	5	1C2T8S	110
3	1	3	2C2T3S	001

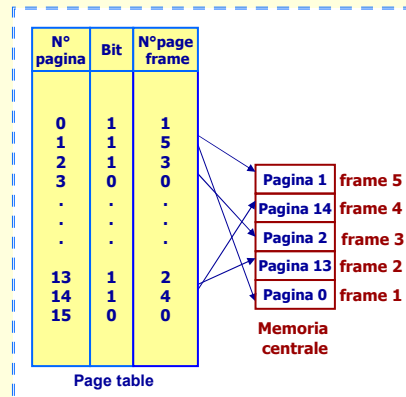
1: in mem. Cent.
0: non in mem.cent.

r w m

Paginazione

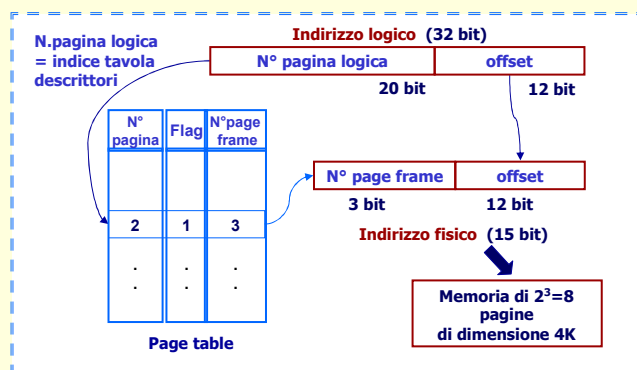
❖ Traduzione da indirizzo logico ad indirizzo fisico

- Effettuata dal dispositivo **MMU (Memory Management Unit)** utilizzando la tavola delle pagine (**page table**)
 - Ad ogni pagina è associato un bit che indica la presenza o l'assenza di tale pagina in memoria



Paginazione

❖ Creazione di un indirizzo fisico da un indirizzo logico



Paginazione

❖ Page fault

- Riferimento ad una pagina che non si trova in memoria

Page fault
Eccezione che provoca un page demand con diverse strategie possibili

- Il SO, grazie anche all'intervento della MMU,
 - libera lo spazio in memoria
 - trasferisce la pagina richiesta dal disco in memoria
 - aggiorna la page table con l'indirizzo fisico della pagina appena trasferita

❖ Algoritmi di rimpiazzamento delle pagine

- **Least Recently Used (LRU)**
 - Rimozione della pagina usata meno recentemente, perché la probabilità che non sia utilizzata al momento è alta
- **First In First Out (FIFO)**
 - Rimozione della pagina caricata in memoria meno recentemente, indipendentemente dall'ultimo utilizzo della pagina

Segmentazione

□ Segmentazione

❖ Metodo di gestione della memoria che permette che lo spazio degli indirizzi fisici di un processo sia contiguo

- La memoria non è partizionata a priori
- Il programma (spazio di indirizzi logici) è suddiviso in spazi di indirizzamento completamente indipendenti, detti **segmenti**, dal programmatore o dal compilatore
- Il SO fa uso di una tabella dei segmenti (**segment table**) per ogni processo
 - La tabella memorizza l'indirizzo iniziale e la lunghezza di ogni segmento e la lista di spazi liberi nella memoria principale

❖ Segmento

- Sequenza lineare di indirizzi
 - Segmenti diversi possono avere dimensioni diverse
 - La dimensione di un segmento può variare durante l'esecuzione del programma
- Gli indirizzi fisici sono calcolati in base al numero di segmento e all'offset

Un segmento è un'entità logica, nota al programmatore, che contiene un blocco logico di informazioni (es. codice, stack, dati, ...)

Segmentazione

❖ Segment table

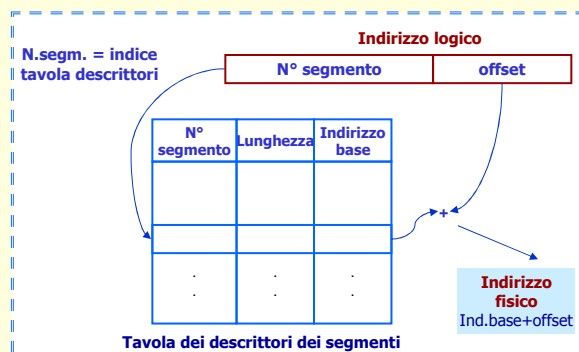
- Il S.O. fa uso, per ogni processo, di una **tabella dei segmenti** che mostra l'indirizzo di base del segmento e la sua lunghezza

Struttura della segment table

Segmento	Indirizzo base	Lunghezza segm.	Modalità di accesso
1	1500	3KB	100
2	4600	8KB	110
3	14000	200KB	001
.....

Segmentazione

❖ Creazione di un indirizzo fisico da un indirizzo logico



Paginazione e Segmentazione

❖ **Paginazione**

- Non visibile al programmatore
- Elimina **frammentazione esterna** e consente un uso efficiente della memoria



Lo spazio di indirizzi logici è monodimensionale, ossia gli indirizzi sono consecutivi

❖ **Segmentazione**

- Visibile al programmatore.
- Elimina la **frammentazione interna** e consente la modularità (segmento codice e segmento dati)



Lo spazio di indirizzi logici è bidimensionale, ossia formato da più spazi lineari non necessariamente contigui

Paginazione e Segmentazione

❖ **Segmentazione con Paginazione**

- Ogni segmento è diviso in pagine di dimensione fissa ed è quindi dotato della page table
 - Intel 80386

Indirizzo logico

N. segmento	N. pagina	Offset
-------------	-----------	--------

- Il numero del segmento permette l'accesso alla tabella del segmento
- Dalla tabella dei segmenti si ricava l'indirizzo base della page table del segmento stesso