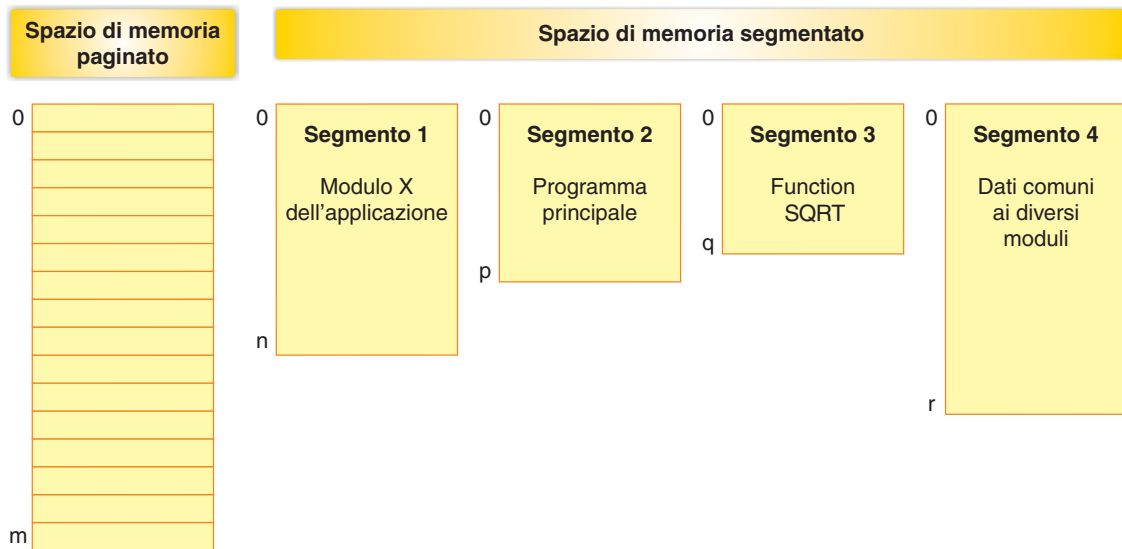


# La segmentazione

La **segmentazione** è una tecnica di allocazione non contigua dei processi in memoria che differisce dalla paginazione perché i processi sono suddivisi in blocchi, detti **segmenti**, di differente dimensione. Analogamente a quanto succede alle pagine con la paginazione, un processo segmentato è allocato in memoria per segmenti non necessariamente adiacenti.

Con la segmentazione un processo è suddiviso in blocchi secondo criteri logici che riflettono l'organizzazione del software. Per esempio un programma potrebbe essere composto da un certo numero di segmenti secondo la strutturazione dei moduli che lo compongono: un segmento per il programma principale, uno per ogni sottoprogramma, un altro ancora per un blocco di dati e così via. La segmentazione riflette quindi la visione che il programmatore ha del proprio programma.

Nel caso della paginazione la suddivisione del programma in blocchi avviene secondo criteri geometrici, la singola pagina non ha alcun significato per il programmatore e non c'è nessun collegamento logico tra il contenuto di un blocco e quello del blocco successivo. Quindi la paginazione è invisibile al programmatore.



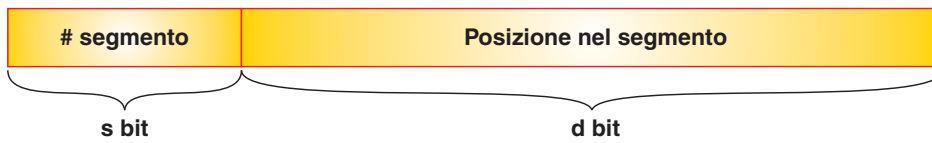
Nel caso della paginazione un processo è formato da un unico spazio di memoria che è suddiviso artificialmente in un certo numero di blocchi. Con la segmentazione un processo è suddiviso in segmenti, ognuno dei quali possiede una precisa identità logica e funzionale e ha un proprio spazio di indirizzamento.

I vantaggi della segmentazione rispetto alla paginazione sono:

- È semplificata la gestione di spazi memoria condivisi: i dati (o il codice) da condividere possono essere inseriti in un unico segmento che viene condiviso tra i processi.
- La protezione della memoria può essere effettuata in modo puntuale in base alle esigenze e alle caratteristiche funzionali del singolo segmento.
- È più semplice gestire strutture dati dinamiche allocandole in un segmento. Se nell'esecuzione del processo lo spazio occupato dai dati aumenta, basta aumentare la dimensione di quello specifico segmento. Nel caso della paginazione è invece impossibile aumentare la dimensione delle pagine ed è più complicato inserire nuove pagine tra quelle assegnate al processo per allocare i dati che possono crescere sino a superare lo spazio inizialmente previsto.

Nello spazio di memoria segmentato i bit che compongono l'indirizzo di una cella di memoria sono scomposti in due parti:

- gli **s** bit più significativi identificano il numero di segmento;
- i rimanenti **d** bit meno significativi indicano la posizione della parola nel segmento.



Poiché nell'indirizzo si usano **s** bit per il numero di segmento e **d** bit per l'*offset*, cioè per la posizione dentro il segmento, si sta operando con un'architettura di processore che permette di definire fino a  $2^s$  segmenti della dimensione massima di  $2^d$  parole di memoria.

Per esempio, supponendo di operare con indirizzi di 32 bit, dei quali 10 precisano il numero di segmento e 22 sono l'offset nel segmento, allora un processo può essere composto al massimo da

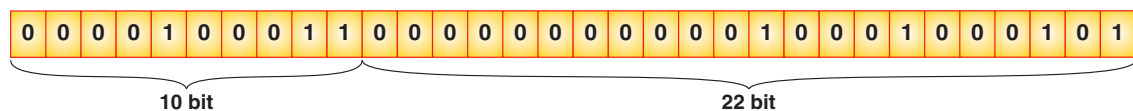
$$2^{10} = 1024 \text{ segmenti}$$

della dimensione massima di:

$$2^{22} \text{ celle} = 2^2 \cdot 2^{20}$$

cioè circa 4.000.000 di celle.

Per esempio l'indirizzo:



referenzia una parola di memoria posta nel segmento:

$$35 = 2^5 + 2 + 1$$

con offset:

$$1093 = 2^{10} + 2^6 + 2^2 + 2^0$$

## La tabella dei segmenti

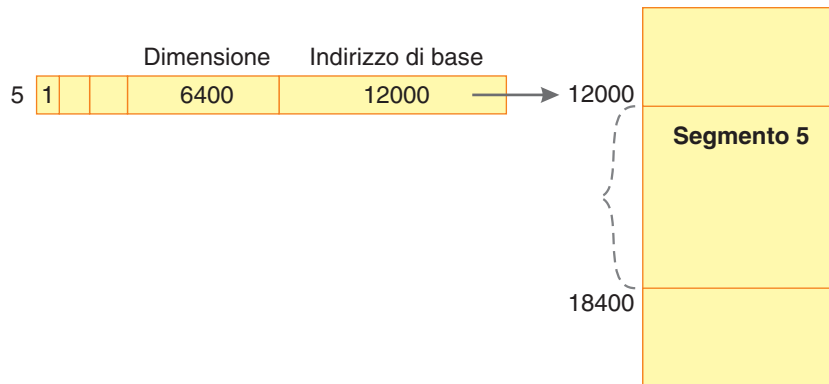
Con la segmentazione un processo è allocato in memoria per segmenti. Per collocare un segmento in memoria il sistema operativo deve trovare una zona libera della memoria di dimensione uguale a quella del segmento. Le informazioni sul posizionamento dei segmenti di un processo in memoria sono inserite in una struttura dati analoga alla tabella delle pagine, detta **tabella dei segmenti**, che è usata dalla MMU per costruire l'indirizzo fisico di una cella di memoria.

La segmentazione può essere usata con o senza la memoria virtuale. Nel caso di memoria virtuale segmentata, il formato di una riga della tabella dei segmenti è il seguente:

V	M	...	Dimensione segmento	Indirizzo base del segmento
---	---	-----	---------------------	-----------------------------

Ogni riga della tabella dei segmenti contiene, oltre ai bit di controllo e all'indirizzo di base del segmento, anche la sua dimensione. L'indirizzo di base e la dimensione del segmento identificano lo spazio di memoria fisica occupato dal segmento.

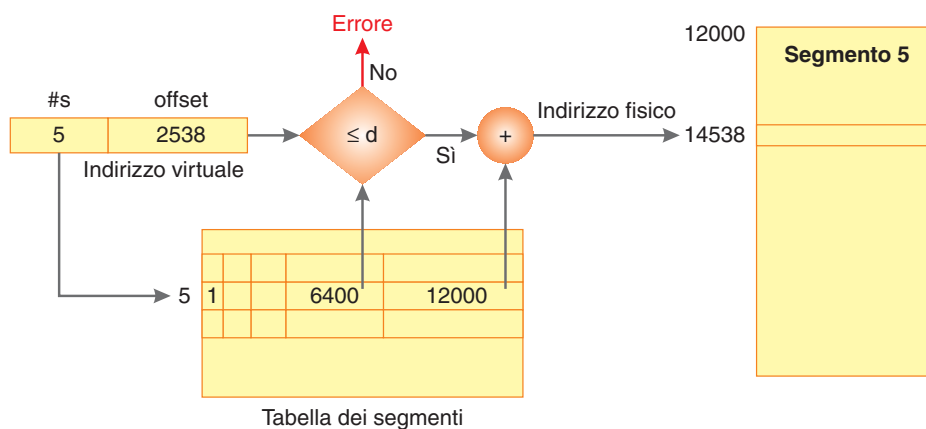
La figura seguente mostra un esempio di elemento di tabella dei segmenti: il segmento 5 del processo è presente in memoria a partire dall'indirizzo 12000 sino alla locazione 18400.



Un processo, nel corso dell'esecuzione, riferenzia la memoria con indirizzi virtuali secondo il formato sopra specificato. La MMU costruisce il *mapping* tra indirizzi virtuali e indirizzi fisici usando le informazioni contenute nella tabella dei segmenti.

L'indirizzo di base della tabella dei segmenti è usato come *registro base* per la rilocazione dinamica del segmento, mentre la dimensione del segmento è usata per proteggere gli accessi alla memoria: un indirizzo virtuale che riferenzia un certo segmento deve avere un offset minore o uguale alla dimensione del segmento stesso. Se invece l'offset è maggiore della dimensione del segmento, significa che il processo ha generato un indirizzo di una cella di memoria esterna al segmento.

Per calcolare l'indirizzo corrispondente all'indirizzo virtuale 05002538, la MMU segue il seguente schema di funzionamento. In esso sono ipotizzati indirizzi virtuali con 2 cifre decimali per il segmento e 6 cifre decimali per la posizione nel segmento.



Il numero di segmento è usato per accedere alla corrispondente linea della tabella dei segmenti. Il bit di validità **V** vale 1 e quindi il segmento è in memoria. L'offset è minore della dimensione del segmento e l'indirizzo riferenzia una posizione della memoria fisica, 14538, ottenuta sommando all'offset, 2538, l'indirizzo di base del segmento 5, 12000.

Nella costruzione dell'indirizzo fisico ci sono due possibili situazioni anomale:

- un offset di valore maggiore della dimensione del segmento
- l'assenza del segmento in memoria.

La MMU, rilevata la condizione di errore, la segnala con un'interruzione che provoca l'intervento del sistema operativo per portare il segmento mancante in memoria o per interrompere l'esecuzione del programma per violazione della memoria.

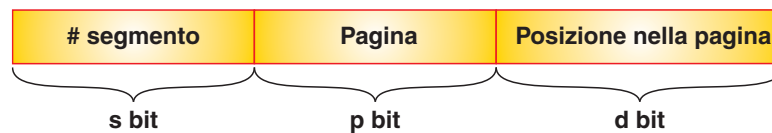
Dal punto di vista dell'allocazione fisica dei segmenti in memoria, si può pensare che un processo sia stato suddiviso in un insieme di parti che sono allocate in memoria come avviene con le partizioni a dimensioni variabili e, quindi, si possono avere gli stessi problemi causati dalla frammentazione esterna.

In ogni caso la memoria virtuale segmentata si differenzia dalla gestione della memoria a partizioni dinamiche in quanto:

- i segmenti sono più piccoli dei processi, migliorando l'utilizzazione della memoria e riducendo gli effetti della frammentazione;
- i segmenti permettono l'allocazione non contigua del processo in memoria;
- non è necessario che tutti i segmenti siano presenti in memoria durante l'esecuzione del processo.

Sia la segmentazione che la paginazione hanno pregi e difetti. Per utilizzare i pregi delle due tecniche, sono stati realizzati sistemi di indirizzamento che combinano segmentazione e paginazione. In tal caso la gestione della memoria consiste nella paginazione dei segmenti che compongono un processo.

Un indirizzo virtuale ha il seguente formato:



Un indirizzo con **s** bit per il numero di segmento, **p** bit per il numero di pagina, e **d** bit per l'offset, permette di definire segmenti composti da  $2^p$  pagine di  $2^d$  parole di memoria.

Il calcolo dell'indirizzo fisico richiede che per ogni processo sia definita una tabella dei segmenti. Ogni riga della tabella dei segmenti punta a una tabella delle pagine del segmento che realizza il *mapping* tra pagina virtuale del segmento e frame di memoria fisica. La parola di memoria referenziata si trova nel frame così individuato, nella posizione precisata dall'offset.

La segmentazione e paginazione combinate richiedono una MMU più sofisticata e una maggior complessità delle strutture dati gestite dal sistema operativo.