

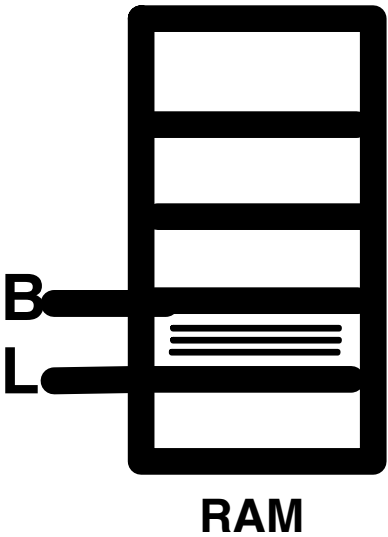
Partizionamento statico a partizioni multiple
(sistemi multiprogrammati)

Ma se siamo in uno scenario in cui dobbiamo gestire più di un'applicazione attiva (gli scenari in cui avevamo un'applicazione attiva per volta come esecuzione seriale e sistemi batch monoprogrammati, sono scenari risolvibili in maniera semplice in tecniche di binding basilari, come binding a tempo di compilazione e caricamento, non abbiamo bisogno neanche di una mmu) quindi abbiamo binding a tempo d'esecuzione, e soprattutto abbiamo la possibilità di mantenere più applicazioni attive, se guardiamo l'altro lato dell'architettura, quindi NON CPU VERSO MEMORIA ma MEMORIA VERSO CPU e in particolare memoria verso applicazioni che noi vogliamo ospitare in memoria:

La memoria com'era realmente utilizzata?

Uno schema classico era il cosiddetto partizionamento statico a partizioni multiple. Questo è per sistemi Multi-Programmati, quindi sistemi batch multiprogrammati o sistemi time sharing.

Come funziona?
È la modalità con la quale decidiamo di voler ospitare le applicazioni all'interno della RAM.
Quindi la regola con la quale utilizziamo la RAM per ospitare gli address space delle applicazioni.
Lo spazio di indirizzamento fisico (RAM) è suddiviso in partizioni fisse di taglia uguale o diversa. Ogni partizione è dedicata per ospitare un singolo processo.
Quindi nello schema di partizionamento statico da cui si è partiti nel momento in cui avevamo il binding a tempo d'esecuzione e l'idea di dover gestire multiprogrammazione, quello che è successo è esattamente questa cosa qua:

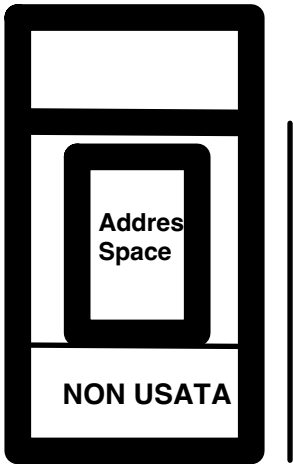


Questa è la RAM. La mia RAM la vedo come logicamente suddivisa in zone chiamate "partizioni". E quando devo attivare un'applicazione - se c'è una di queste zone libere - allora l'applicazione la carico all'interno della zona libera.

Supponiamo che sia libera la penultima zona, inseriamo l'applicazione Lì, e sarà evidente che la base diventerà B (Che indica la posizione di quella partizione della RAM) e il limite sarà L (ovvero il punto finale di quella partizione all'interno della RAM).
Tipicamente le partizioni che abbiamo lì possono essere anche di taglia differente, il che implica dire che se io devo attivare applicazioni che hanno un address space piccolo, posso caricarle in zone più piccole della RAM. Considerando sempre che la taglia massima di un address space è funzione del numero di bit che si utilizzano per esprimere gli indirizzi logici.

Questo schema ha evidenti problemi.

Ogni processo occupa comunque un'intera partizione della RAM: un processo P può occupare, all'interno di una partizione grande, solo una ZONA, e quindi la zona di sotto non viene utilizzata.



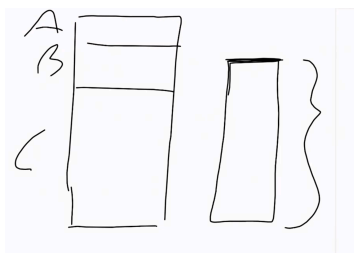
Il non utilizzo di una zona di memoria occupata, si chiama FRAMMENTAZIONE INTERNA.

Unica Partizion
e
Ci sono delle zone interne alle zone della RAM, che non possono essere realmente utilizzate.

RAM

E poi un altro problema è che partizionare la RAM in partizioni statiche, va contro la possibilità di lanciare applicazioni che potrebbero essere troppo grandi per essere ospitate su una singola partizione.

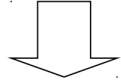
Questo address
space non entra
in una singola



partizione.
Perché
chiaramente
queste
partizioni erano
statiche.

Problemi

- ogni processo occupa una intera partizione indipendentemente dalla sua taglia (**frammentazione interna**)
- un programma potrebbe essere troppo grande per essere contenuto in una singola partizione



Entrambi i problemi possono essere alleviati nel caso di partizioni di taglia diversa (molte di taglia piccola, poche di taglia grande)

Il grado di multiprogrammazione è "limitato" dal numero di partizioni se non abbiamo lo SWAPPING. Se non è vero che possiamo liberare una partizione temporaneamente per caricare qualcos'altro per poi ricaricare in memoria l'applicazione che abbiamo swappato fuori, il numero massimo delle applicazioni che possiamo mantenere attive è pari solo al numero delle partizioni che STATICAMENTE erano gestite dal software del sistema operativo.

Entrambi i problemi li ALLEVIAMO (Non risolviamo) con partizioni di taglia diversa: Tra tutte le partizioni di taglia grande qualcuna sufficientemente grande per ospitare una specifica applicazione con un address space ampio poteva esserci, mentre le partizioni piccole possono essere utilizzate per applicazioni più piccole che, se caricate in una partizione grande, genererebbero una frammentazione interna molto ampia.