GESTIONE DELLA MEMORIA

La memoria persistente è un disco magnetico, viene chiamata memoria di massa.

La memoria principale viene chiamata RAM.

La memoria cache ha piccole dimensioni ma è molto veloce. Essa però a differenza delle altre due memorie è invisibile al sistema. Il suo scopo è quello di aumentare le prestazioni.

Ogni memoria deve assicurare:

1. Disponibilità dio memoria per il codice di ogni programma.
2. Protezione dei dati e delle istruzioni.
3. Il riutilizzo della memoria dei programmi terminati.
4. Eventuale condivisione di aree di memoria tra processi cooperanti.

**PAGINAZIONE DELLA MEMORIA E TRANSLAZIONE DEGLI INDIRIZZI**

La pagina è un blocco della memoria da 4kb. Ogni memoria è divisa da blocchi da 4kb.

Un metodo semplicissimo consiste nell’allocare a ogni processo un settore di memoria compreso tra un indirizzo iniziale ed uno finale.

In un computer il processore elabora indirizzi logici mentre il bus di interconnessione con la memoria principale sono emessi gli indirizzi fisici traslati dalla MMU.

L’attività della MMU è ora più complessa, per traslare l’indirizzo logico in indirizzo fisico deve prima determinare la pagina logica. La traslazione avviene sottraendo l’indirizzo base della pagina logica e sommando l’indirizzo base della pagina fisica.

La MMU gestisce gli indirizzi logici e li assegna ad indirizzi fisici.

La translazione degli indirizzi è un’operazione completamente hardware ma il gestore della memoria del S.O. ha il compito di impostare la tabella nella MMU.

**MEMORIA VIRTUALE** pag 116

Un sistema operativo ha molti processi attivi. Può capitare che i programmi attivi superano la disponibilità di memoria RAM disponibili. Alcuni processi possono restare in ready per tanto tempo ed occupare RAM. Per questo c’è la memoria virtuale, cioè che se la memoria principale, la RAM, è piena i dati vengono salvati nella memoria persistente temporaneamente.

La SSD ha un o file, che sarebbe la mia memoria virtuale denominata area di swap o di paging. Se la pagina non viene trovata in memoria il programma passa allo stato di wait. Questo viene chiamato page-fault. Deve prendere la pagina dalla Swapp e portarla in memoria.

Usare questa tecnica però rallenta i processi.

Se al page-fault non sono disponibili pagine fisiche allora vanno eliminate pagine e copiate nell’area di swap nella memoria persistente.

Politiche di eliminazione di pagine:

1. FIFO, viene eliminata la prima che è stata caricata.
2. LRU, viene eliminata quella inutilizzata da più tempo.
3. LFU, viene eliminata quella meno utilizzata.

La MMU tiene anche il conto a secondo di queste politiche.

Nel caso di politica FIFO tiene in memoria il l’ora.

La MMU approssima la LRU cambiando solo un bit di referenza per gli accessi da 0 ad 1.

Prima di eliminare una pagina dalla memoria è da salvare nella pagina di swapp. Così assegno un dirty-bit per dire se quella pagina è stata scritta e dopo averla scritta la vado a copiare nell’area di swapp.

L’innalzamento del tasso di page-fault si chiama trashing.

POLITICHE DI LIBERAZIONE DELLO SWAPP

1. Demand-paging: assegnano i frame in base alle richieste.
2. Working-set: il gestore della memoria decide quanti frame assegnare.

**GESTIONE DELLA MEMORIA IN WINDOWS E LINUX**

Entrambi i sistemi usano la memoria virtuale. Entrambi i sistemi hanno una memoria composta da pagine di 4Kb. Pagine di memoria troppo grandi occupano troppa memoria. Quelle invece troppo piccoli diventerebbero troppo numerose.