Sistemi Operativi - Schema generale

Premessa: il seguente schema è incompleto. Portate pazienza lo ho realizzato in un pomeriggio. Il documento non è formattato. Modificatelo, miglioratelo e condividetelo.

Sincronizzazione

Metodo per valutare l’efficacia e l’eleganza di modelli e meccanismi per la sincronizzazione tra processi

* Filosofi a cena : accesso esclusivo a risorse limitate

Filosofo(i) {

while(1) {

<pensa>

if(i == X) {

P(f [(i+1)%N]);

P(f [i]);

} else {

P(f [i]);

P(f [(i+1)%N]);

}

<mangia>

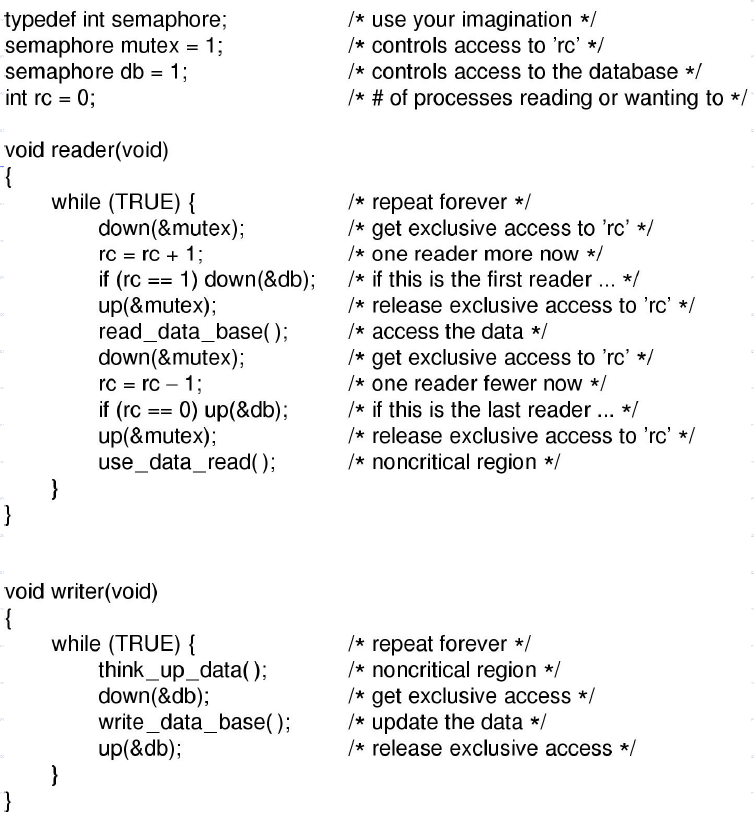
V(f [i]);

V(f [(i+1)%N]);

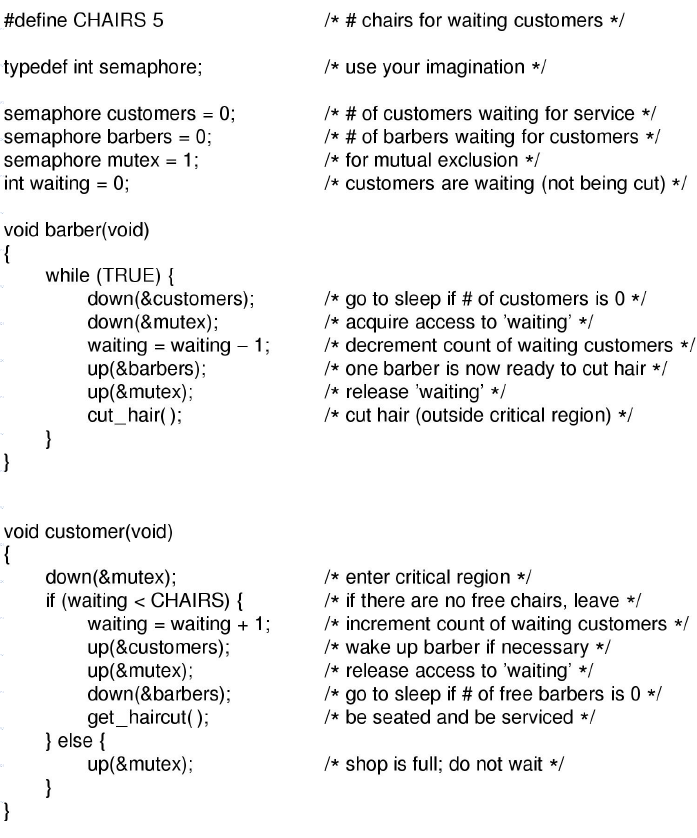
}

}

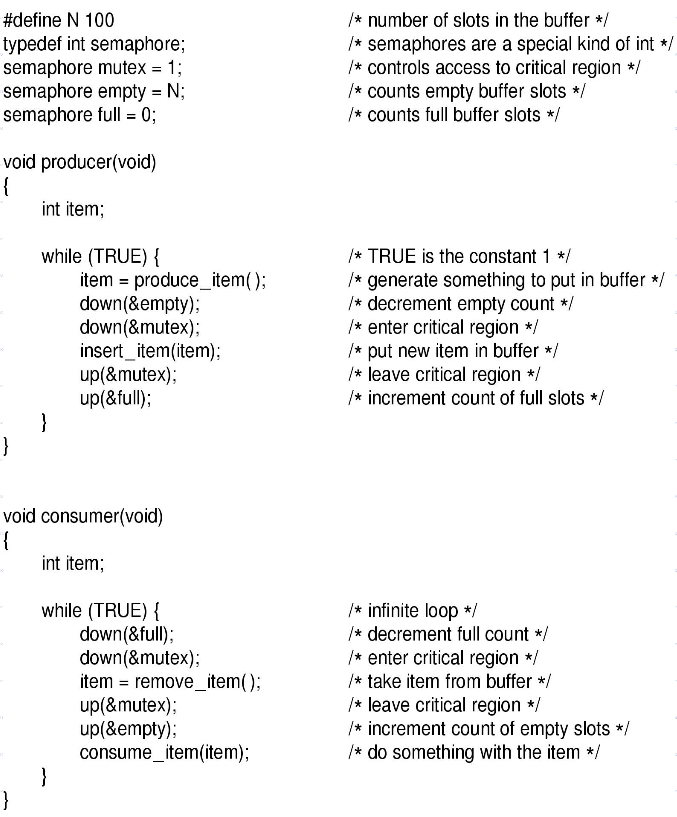
* Lettori e scrittori : accessi concorrenti a basi di dati



* Barbiere che dorme: prevenzione di race condition



* (Produttore e Consumatore)



Stallo condizioni necessarie e sufficienti:

* Accesso esclusivo a risorsa condivisa
* Accumulo di risorse
* Inibizione di prerilascio
* Condizione di attesa circolare

Ordinamento di processi

Caratteristiche desiderabili per le politiche di ordinamento:

* Equità (fairness)
* Coerenza (enforcement)
* Bilanciamento

Politiche scambio processi:

* FCFS (First-Come-First-Served) penalizza i processi i/o bound
* Round-Robin = FCFS + Time-sharing
* SJF (Shortest job first)
* SRTN (Shortest remaining time next)

Tempo di attesa = tempo che intercorre dall’arrivo del processo alla sua prima esecuzione sommati tutti i tempi in cui il processo è pronto ma non in esecuzione

Tempo di Turn Around = Tempo che impiega il processo a terminare da quando arriva (cioè si mette in coda)

Tempo di risposta = tempo che intercorre dall’arrivo del processo alla sua prima esecuzione.

Gestione della memoria

Uso stimato CPU = 1-P^N (P= percentuale di tempo in i/o . N processi in memoria.)

**Swapping**: Tecnica rudimentale per alternare processi in memoria. Rischio frammentazione **esterna**. La partizione si assegna con margine.

Per tenere traccia delle partizioni libere si usano due tecniche: 1) Mappe di **Bit** e 2) **Liste concatenate**

Strategie di allocazione delle liste concatenate:

* First fit : il primo segmento libero ampio abbastanza
* Next fit : come First fit ma cercando sempre avanti
* Best fit : il segmento libero più adatto
* Worst fit : sempre il segmento libero più ampio
* Quick fit : liste diverse di ricerca per ampiezze “tipiche”

**Memoria virtuale**: Ogni processo ha una memoria maggiore della ram (grazie all’hdd).

Tecniche di gestione

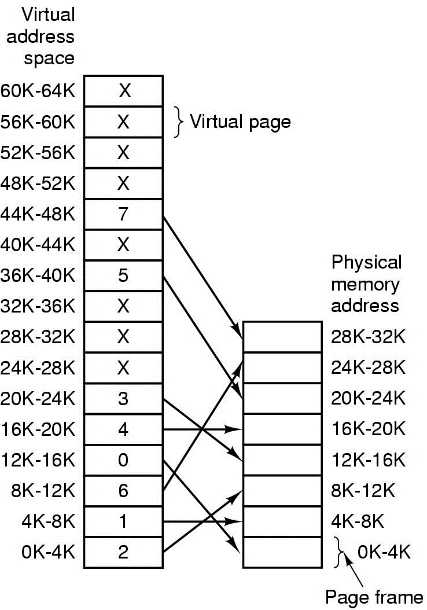
1)**Paginazione:**

La MMU traduce gli indirizzi virtuali in indirizzi logici

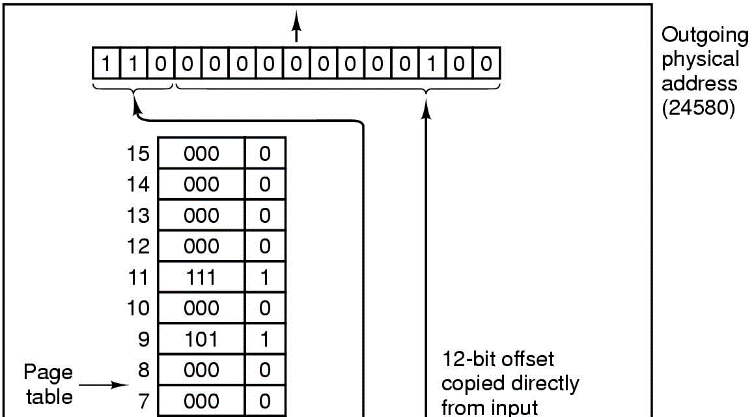
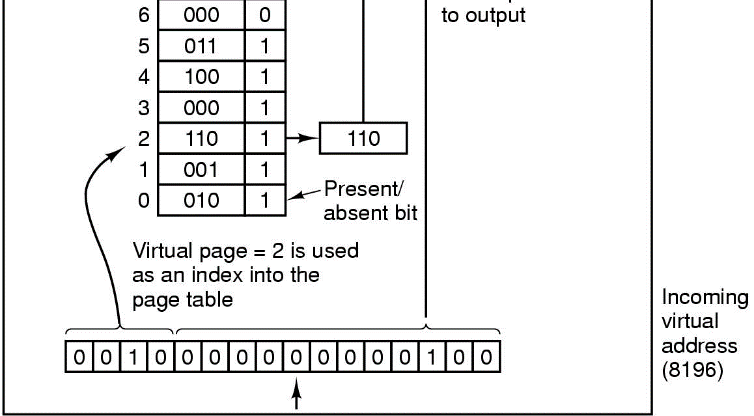
La memoria virtuale è divisa in **Pagine**.

La RAM e divisa in **Page Frame**. Dimensione Page Frame = Dimensione Pagine

Per la traduzione si usa la tabella delle pagine.

Serve per capire se la pagina è presente in Ram e la sua

Posizione.

 Ogni processo ha la sua.

**Esempio di traduzione**

La tabella delle pagine è così grande che non può stare nei registri ma deve stare in RAM. Oneroso riferirla ad ogni indirizzo emesso. È stata creata la TLB (translation lookside buffer, consente la scansione parallela, HW interna alla MMU). Oggi sono software (MMU più semplice).

Con le architetture a 64bit la tabella delle pagine è enorme quindi si adotta una tabella delle pagine invertita. La tabella specifica quali righe della ram sono occupate oppure no dalle pagine del processo. Difetto nell’efficienza perché bisogna scandire tutta la tabella (la TLB aiuta).

In caso di page fault le politiche di rimpiazzo sono:

* Rimpiazzo ottimale (optimal replacement): Rimpiazza la pagina che non verrà usata per maggior tempo.
* NRU (Not Recently Used): Ogni page frame ci sono due bit M (modificato) e R (referenziato, posto periodicamente a 0 dal S/O). Sostituisce una pagina dalla classe non vuota più bassa.

Quattro classi:

* + Classe 0: non riferita, non modificata
  + Classe 1: non riferita, modificata
  + Classe 2: riferita, non modificata
  + Classe 3: riferita, modificata
* FIFO: Rimuove la pagina ad ingresso più antico.
* Second chance: Come la FIFO ma utilizza un bit R per dare una seconda chance.
* Orologio: come Second chance ma procede in senso circolare riprendendo ogni volta da dove si era fermata.
* LRU (Least Recently Used): Rimpiazza quello usato meno di recente cioè quello che non è stato usato da parecchio tempo.
* NFU (Not Frequently Used): Rimpiazza quello che è stato utilizzato meno. Utilizza un contatore e ha memoria.
* Aging (not frequently used modificato): DA FARE!
* Working set (WS): DA FARE!
* Working set con orologio (WS): DA FARE!

**2) Segmentazione:**

DA FARE!

**File system**

**File system** servizio progettato dal s/o per soddisfare:

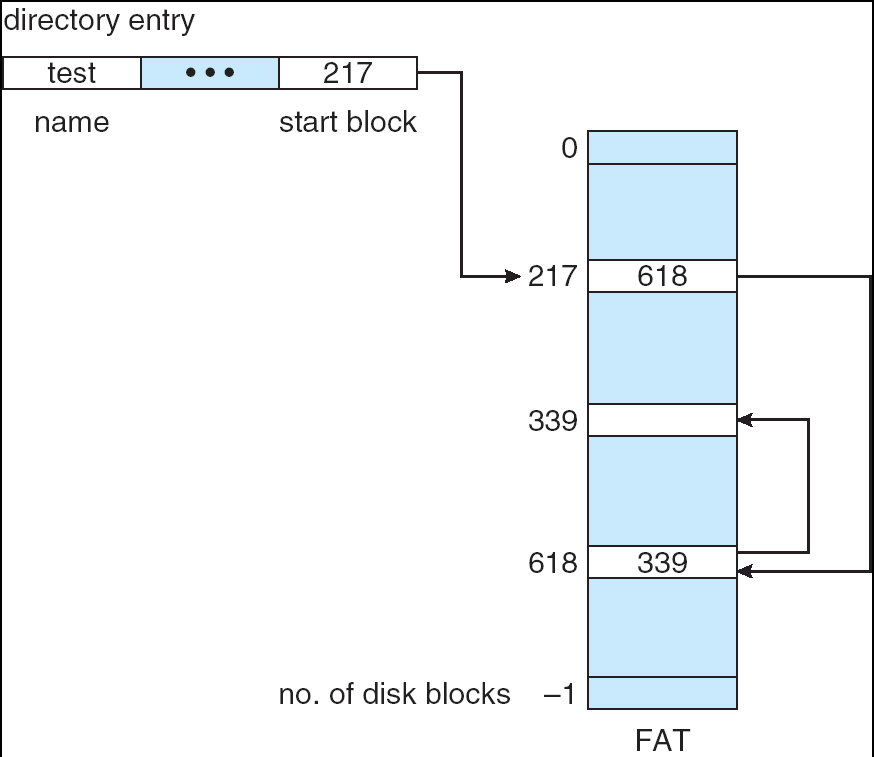
* Persistenza dei dati
* Possibilità di condivisione dei dati tra applicazioni distinte
* Nessun limite di dimensione fissato a priori

Il **File System** designa la parte di s/o che si occupa di:

* Organizzazione
* Gestione
* Realizzazione
* Accesso

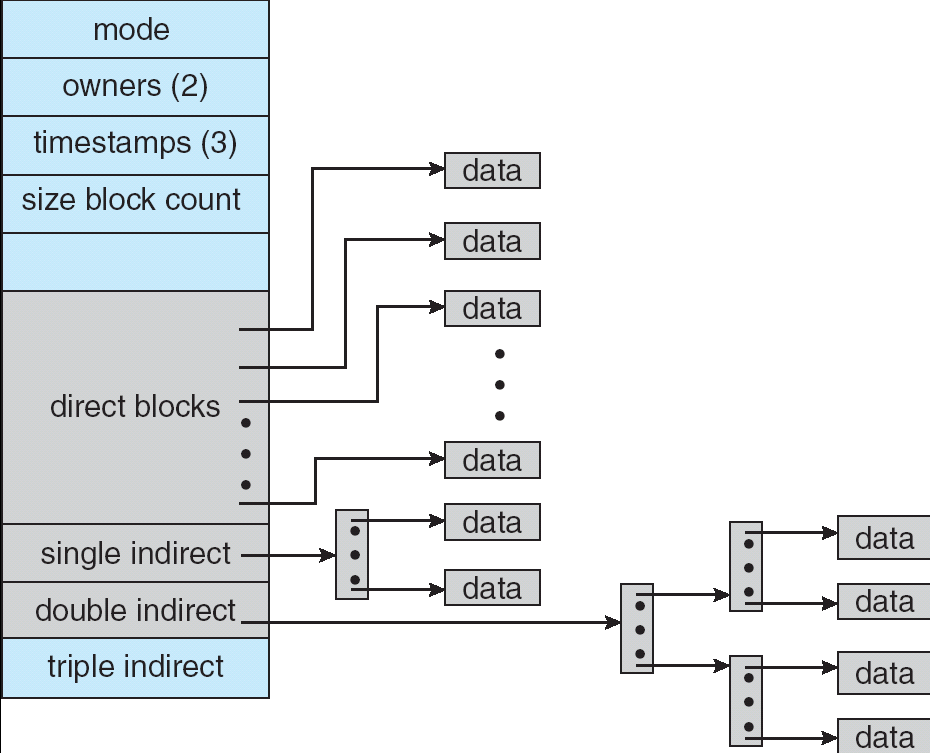
Allocazione file:

* Allocazione contigua: Memorizza i file su blocchi consecutivi quindi basta definire l’inizio e la lunghezza. Usato su CD-Rom. Modifica comporta frammentazione.
* Allocazione a lista concatenata: Coppia di puntatori all’inizio e alla fine del file. Ogni blocco contiene il puntatore al blocco successivo.
* Allocazione a lista indicizzata: Struttura di puntatori a blocchi. Ci sono due strateggie:
  + Forma Tabulare (FAT, File Allocation Table)
  + Forma indicizzata (nodo indice, i-node)

Non c’è frammentazione esterna. Accesso sequenziale e diretto.

****  **Nodi indice**

**FAT**



**FS in Unix v7**

Gestione dei blocchi liberi:

* Vettore di bit (bitmap)
* Lista concatenata di blocchi (sfrutta i campi liberi). Usata dalla FAT

NTFS (New Technology File System):

DA FARE!