

SO**20/21**

Università degli Studi della Basilicata
Corso di Laurea in Scienze e Tecnologie Informatiche

2501**Sistemi Operativi - A.A. 2020/2021**

Simulazione di esame del 25/01/2020

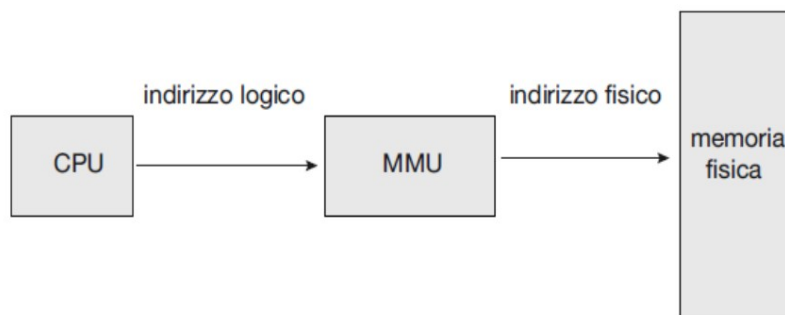
Tempo a disposizione: 2 ore

Domanda 1 (max 5 punti)

Che cos'è la Memory Management Unit (MMU)? Fornire uno schema grafico di funzionamento della MMU.

Risposta

L'unità di gestione della memoria (memory management unit, MMU) svolge l'associazione nella fase d'esecuzione dagli indirizzi virtuali agli indirizzi fisici. Uno schema generale del funzionamento della MMU è riportato nella figura seguente.



Per tradurre un indirizzo logico nel formato <pagina p, offset> in un indirizzo fisico la MMU compie i seguenti passi:

1. Estrae il numero di pagina p e lo utilizza come indice nella tabelle delle pagine
2. Estrae il numero di frame f corrispondente dalla tabella delle pagine
3. Sostituisce il numero di pagina p nell'indirizzo logico con il numero di frame f

Domanda 2 (max 5 punti)

Descrivere che cosa sia una system call. Utilizzare opportuni esempi pratici per integrare la spiegazione.

Risposta

Una system call (o syscall) è una chiamata diretta al sistema operativo da parte di un processo di livello utente (ad esempio, una richiesta di I/O). In seguito alla chiamata di una syscall, verrà generato

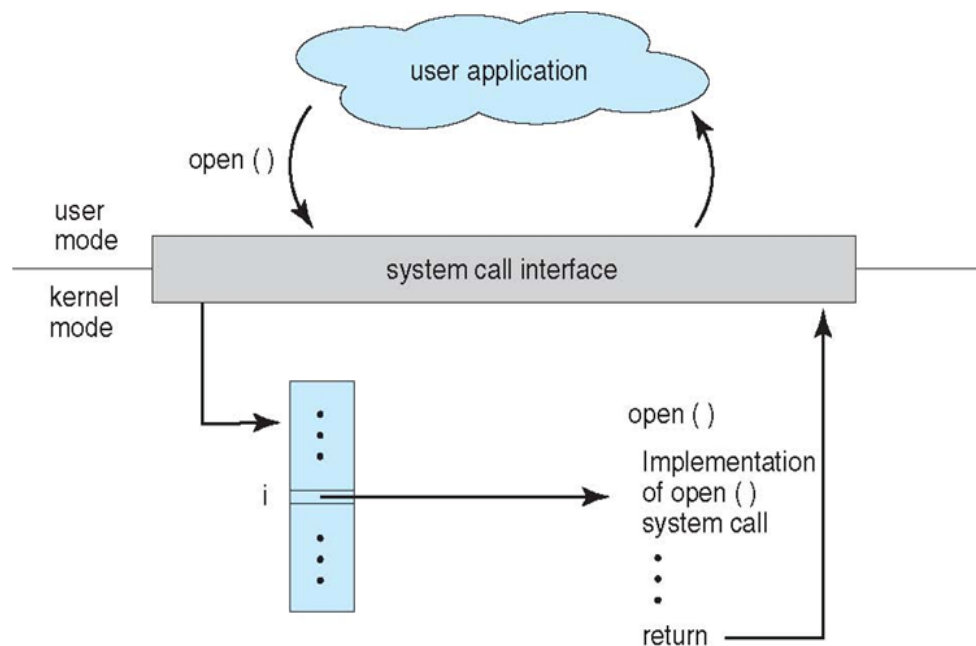
un interrupt software (denominato trap), in modo da poter richiamare l'opportuna funzione associata a tale syscall utilizzando la Syscall Table (ST).

Prendiamo, ad esempio, la system call `open()` nei sistemi UNIX-based, definita con la seguente segnatura

```
int open(const char *pathname, int flags);
```

La system call `open()` viene utilizzata per aprire il file specificato da `pathname`. Se il file non esiste, esso potrà essere creato da `open()` se `O_CREAT` è stato specificato in `flags`.

La figura sotto mostra uno schema di esecuzione per `open()`



I passaggi nella figura sopra sono i seguenti.

1. Il programma utente deve specificare gli argomenti per la system call
2. Una volta specificati, il programma esegue la system call
3. Il controllo passa al sistema operativo, che estrae dalla tabella delle system call la routine adatta a gestire l'interrupt software ricevuto e la esegue.
4. Una volta terminata l'esecuzione della routine, il sistema operativo restituisce il controllo al programma utente.

Domanda 3 (max 5 punti)

Si descrivano le principali tipologie di memorie di massa evidenziandone vantaggi e svantaggi.

Risposta

Le due principali tipologie di memoria di massa (o secondaria) attualmente utilizzate sono

1. dischi rigidi (hard disk)
2. dispositivi NVM (nonvolatile memory)

I principali vantaggi degli hard disk sono:

- Grande capacità
- Basso costo
- Prestazioni di lettura/scrittura costanti nel periodo di vita del dispositivo

I principali svantaggi degli hard disk sono:

- Velocità di accesso ai dati considerevole
- Presenza di parti meccaniche che possono danneggiarsi a causa di movimenti bruschi o cadute
- Peso non trascurabile
- Rumorosità durante il funzionamento

I principali vantaggi dei dispositivi NVM sono

- Elevata velocità
- Buona affidabilità
- Dimensioni ridotte
- Ridotto consumo energetico

I principali svantaggi dei dispositivi NVM sono

- Costo elevato
- Capacità ridotta
- Prestazioni in scrittura variabili nel tempo

In generale, gli hard disk sono da preferire quando il costo è più importante delle performance. I dispositivi NVM sono utili quando sono più importanti le performance, la durata della batteria e l'affidabilità.

Esercizio 1 (max 7,5 punti)

Sia data la seguente successione di riferimenti alle pagine di memoria:

4, 2, 7, 3, 7, 1, 2, 4, 7, 3, 5, 1, 1, 3, 7

Si assuma

- di avere una memoria di 3 frame, gestita con politica LRU (least recently used)
- che T_{ma} e T_{pf} siano rispettivamente i tempi di accesso in memoria e di gestione del page fault

1. Qual è il tempo di accesso effettivo (T_{EAT}) in memoria per la situazione descritta?
2. Qual è la probabilità di avere un page fault?

Risposta

Il calcolo del numero di page fault che si ottengono applicando una politica LRU (least recently used) rispetto alla successione di riferimenti alle pagine di memoria è illustrato nella figura seguente.

4	4	4	3	3	3	2	2	2	3	3	3	3	3	3
	2	2	2	2	1	1	1	7	7	7	1	1	1	1
		7	7	7	7	7	4	4	4	5	5	5	5	7

Dalla figura possiamo notare che i page fault (in rosso) sono 12, mentre i page hit (in verde) sono 3. Si avrà quindi:

$$T_{EAT} = 3 \times T_{ma} + 12 \times T_{pf}$$

La probabilità di page fault sarà dell'80%, ottenuta come

$$ppf = 12 / 15 = 0.8$$

Esercizio 2 (max 7,5 punti)

Si assuma di avere la memoria nella situazione illustrata a lato, con la seguente lista delle allocazioni disponibili:

200 KB, 600 KB, 400 KB, 300 KB.

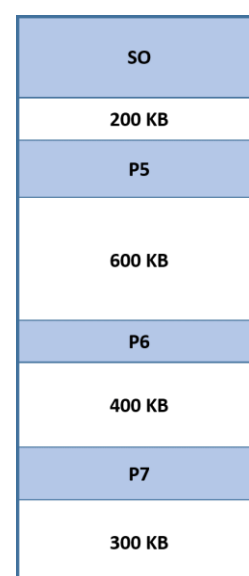
Volendo allocare per intero in memoria i seguenti processi

P1 (250 KB), P2 (120 KB), P3(280 KB), P4 (50 KB)

e adottando un approccio di allocazione worst-fit con una politica

di scheduling FCFS, come verranno allocati P1, P2, P3 e P4?

Motivare la risposta, mostrando graficamente l'evoluzione dell'occupazione della memoria con l'allocazione dei processi sopra elencati.



Risposta

La figura seguente mostra l'evoluzione dell'occupazione della memoria man mano che vengono allocati i processi P1, P2, P3 e P4. Al termine delle allocazioni, la memoria libera, disponibile per le allocazioni sarà: 200 KB, 70 KB, 280 KB, 250 KB.

