

Sistemi Operativi

3 settembre 2013

Compito

Si risponda ai seguenti quesiti, giustificando le risposte.

- Che cos'è la multiprogrammazione? Si può realizzare su sistemi uniprocessore?
 - (*) In un sistema basato su thread di livello utente, c'è uno stack per ogni thread oppure uno stack per processo? Che cosa succede se i thread sono di livello kernel?

Risposta:

- Un sistema multiprogrammato suddivide la memoria in partizioni, dedicando ognuna di queste ultime a contenere l'immagine di un job. In questo modo se il job correntemente in esecuzione si mette in attesa del completamento di un'operazione di I/O, il sistema operativo può far eseguire un altro job tra quelli presenti in memoria. In questo modo non si spreca tempo di CPU per l'attesa del completamento di operazioni di I/O. La multiprogrammazione può essere realizzata anche su un sistema con una singola CPU.
 - (*) In un sistema basato su thread di livello utente c'è un unico stack per processo; infatti i singoli thread sono gestiti per mezzo di apposite librerie in spazio utente e sono completamente invisibili al sistema operativo. Se invece i thread sono a livello kernel, ognuno di essi avrà il proprio stack, essendo gestiti direttamente dal sistema operativo.
- Si discuta ciascuna delle seguenti affermazioni.
 - Un processo può subire una starvation all'entrata di una sezione critica se l'implementazione della sezione critica non soddisfa la condizione di attesa limitata.
 - Attese attive indefinite si possono verificare in un SO che utilizza uno scheduling basato su priorità, ma non possono verificarsi in un SO che utilizza uno scheduling Round Robin.
 - (*) In un sistema lettori-scrittori che favorisce gli scrittori, alcuni processi lettori che vogliono leggere i dati condivisi possono bloccarsi anche quando altri processi lettori stanno leggendo i dati condivisi.
 - (*) Un deadlock non può verificarsi nel problema dei filosofi a cena se un filosofo può mangiare solo con una forchetta.

Risposta:

- Vero: per la definizione stessa di condizione di progresso (stabilisce che nessun processo in esecuzione fuori della sezione critica possa impedire l'accesso di altri processi alla sezione critica).
 - Falso: si possono verificare attese *attive* indefinite anche in un SO che utilizza uno scheduling Round Robin. Infatti, essendo l'attesa attiva, è il processo stesso che si pone in un loop (potenzialmente indefinito) in cui ad ogni iterazione controlla il verificarsi di una certa condizione.
 - (*) Vero: infatti è sufficiente l'arrivo di un singolo processo scrittore per bloccare i processi lettori (anche se vi sono già dei processi lettori che stanno accedendo ai dati condivisi).
 - (*) Vero: sicuramente almeno uno dei filosofi riuscirà in ogni istante ad acquisire la forchetta per mangiare.
- Si consideri un sistema con scheduling a priorità con tre code, A, B, C, di priorità decrescente, con prelazione tra code. Le code A e B sono round robin con quanto di 15 e 20 ms, rispettivamente; la coda C è FCFS. Se un processo nella coda A o B consuma il suo quanto di tempo, viene spostato in fondo alla coda B o C, rispettivamente.

Nelle code A, B, C entrano i seguenti processi:

	coda	arrivo	burst
P_1	A	0	20ms
P_2	B	10	25ms
P_3	C	15	20ms
P_4	A	25	20ms

Si determini:

- il diagramma di GANTT relativo all'esecuzione dei quattro processi;

Sistemi Operativi

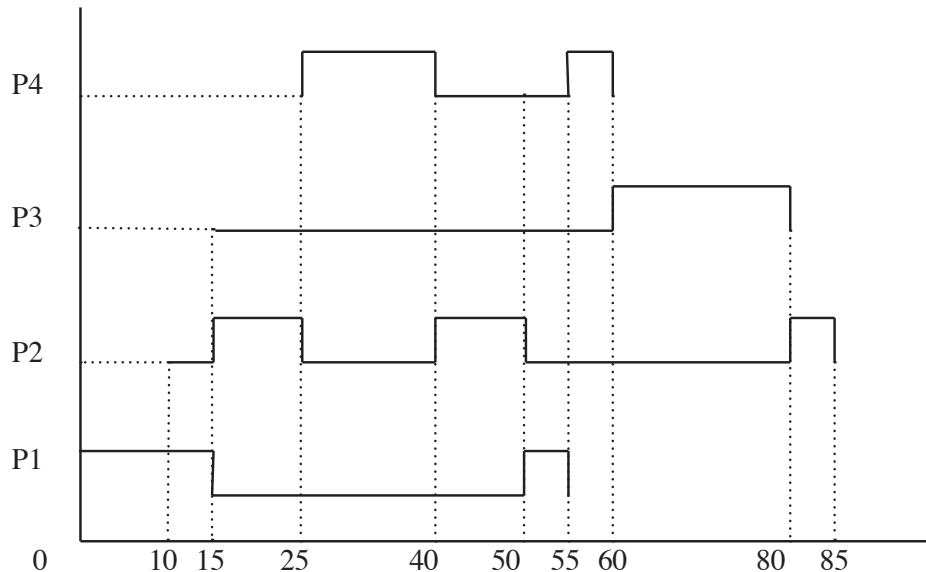
3 settembre 2013

Compito

- (b) il tempo di attesa medio;
 (c) il tempo di turnaround medio.

Risposta:

- (a) il diagramma di GANTT relativo all'esecuzione dei quattro processi è il seguente:



- (b) il tempo di attesa medio è pari a $\frac{35+50+45+15}{4} = \frac{145}{4} = 36,25 \text{ ms}$;
 (c) il tempo di turnaround medio è pari a $\frac{55+75+65+35}{4} = \frac{230}{4} = 57,5 \text{ ms}$.

4. Si descriva brevemente la tecnica di gestione della memoria della paginazione.

Risposta: La paginazione è una tecnica di allocazione della memoria non contigua che prevede:

- la suddivisione della memoria fisica in *frame* (blocchi di dimensione fissa, una potenza di 2, tra 512 e 8192 byte);
- la suddivisione della memoria logica in *pagine* (della stessa dimensione dei frame).

Quindi, per eseguire un programma di n pagine, servono n frame liberi in cui caricare il programma. È compito del sistema operativo tenere traccia dei frame liberi. Non esiste frammentazione esterna e la frammentazione interna è ridotta all'ultima pagina allocata al processo.

Per quanto riguarda la traduzione degli indirizzi logici in indirizzi fisici, ogni indirizzo generato dalla CPU si suddivide in un numero di pagina p e uno spiazzamento (offset) d all'interno della pagina. Il numero di pagina viene utilizzato come indice nella *page table* del processo correntemente in esecuzione: ogni entry della page table contiene l'indirizzo di base del frame fisico f che contiene la pagina in questione. Tale indirizzo di base f viene combinato (giustapposto come prefisso) con lo spiazzamento d per definire l'indirizzo fisico da inviare all'unità di memoria.

5. Si descriva la gestione dell'input/output guidata da interrupt (Interrupt Driven I/O). In tale contesto cos'è e a cosa serve il vettore degli interrupt?

Risposta: Con la gestione dell'input/output guidata da interrupt, una volta inviato il comando di I/O, il processo viene sospeso fintanto che non arriva un interrupt a segnalare il completamento dell'operazione. Durante la sospensione del processo, la CPU può mandare in esecuzione altri processi o thread. Di fondamentale importanza è il vettore delle interruzioni che consente di selezionare la routine di gestione opportuna per ogni tipo di interrupt. Infatti il numero intero che contraddistingue quest'ultimo funge da indice del vettore delle interruzioni ed in tale posizione il sistema operativo può recuperare l'indirizzo di memoria della corrispondente routine di gestione.

6. Si descrivano brevemente i concetti di *Matrice d'Accesso*, *ACL* (ovvero, *Access Control List*) e *Capability Ticket*.

Sistemi Operativi

3 settembre 2013

Compito

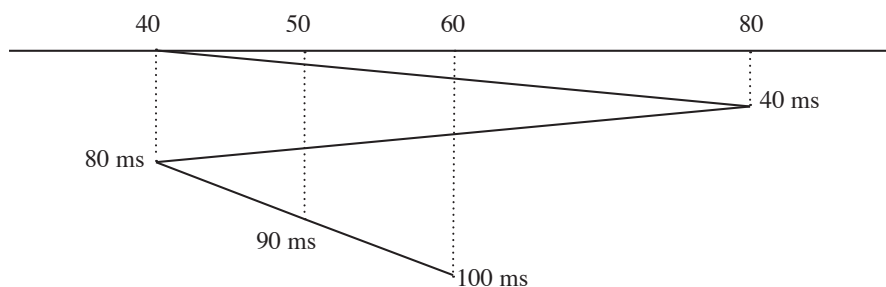
Risposta: Una *matrice di accesso* tiene traccia in ogni riga del dominio di protezione (e.g., utenti, processi) ed in ogni colonna dell'oggetto del filesystem da proteggere; quindi ad ogni coppia (processo, oggetto), associa le operazioni permesse. Dato che le matrici di accesso sono molto sparse, possono essere implementate come access control list: ad ogni oggetto, tali strutture associano chi (dominio) può fare cosa (operazione). Dualmente, i capability ticket invece vengono usati per assegnare ad un processo un insieme di token che indicano che operazioni può compiere e su quali oggetti.

7. Si consideri un disco gestito con politica C-LOOK (con direzione di servizio verso cilindri con numero crescente). Inizialmente la testina è posizionata sul cilindro 40; lo spostamento ad una traccia adiacente richiede 1 ms. Al driver di tale disco arrivano richieste per i cilindri 80, 50, 40, 60, rispettivamente agli istanti 0 ms, 20 ms, 30 ms, 50 ms. Si trascuri il tempo di latenza.

1. In quale ordine vengono servite le richieste?
2. Il tempo di attesa di una richiesta è il tempo che intercorre dal momento in cui è sottoposta al driver a quando viene effettivamente servita. Qual è il tempo di attesa medio per le quattro richieste in oggetto?

Risposta:

1. Le richieste vengono servite nell'ordine 80, 40, 50, 60, come risulta dal seguente diagramma:



2. Il tempo di attesa medio per le quattro richieste in oggetto è dato da $\frac{(40-0)+(80-30)+(90-20)+(100-50)}{4} = \frac{(40+50+70+50)}{4} = \frac{210}{4} = 52,5 \text{ ms}$

Punteggi:

- 9/12 CFU: 1a (2 pt.), 1b (2 pt.), 2a (2 pt.), 2b (2 pt.), 2c (2 pt.), 2d (2 pt.), 3a (3 pt.), 3b (2 pt.), 3c (2 pt.), 4 (3 pt.), 5 (2 pt.), 6 (3 pt.), 7.1 (3 pt.), 7.2 (1 pt.)
- 6 CFU: 1a (3 pt.), 2a (3 pt.), 2b (2 pt.), 3a (4 pt.), 3b (2 pt.), 3c (2 pt.), 4 (3 pt.), 5 (3 pt.), 6 (3 pt.), 7.1 (3 pt.), 7.2 (2 pt.)