

- Che cos'è un sistema multiprogrammato? Si può realizzare la multiprogrammazione su un sistema con una sola CPU?

**Risposta:** Un sistema multiprogrammato suddivide la memoria in parti-zioni, 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.

- Quali sono i vantaggi di un sistema operativo basato su thread rispetto ad uno basato su processi?

**Risposta:** I vantaggi di un sistema operativo basato su thread rispetto ad uno basato sui processi sono i seguenti:

- creare e cancellare thread è circa 100-1000 volte più veloce che creare e cancellare processi in quanto c'è meno informazione da duplicare/creare/cancellare;
- lo scheduling fra thread di uno stesso processo è molto più veloce di quello tra processi;
- la cooperazione di più thread nello stesso task ha come conseguenza un maggior throughput e performance.

- In coda ready arrivano i processi  $P_1, P_2, P_3, P_4$ , con CPU burst e istanti di arrivo specificati in tabella:

|       | arrivo | burst |
|-------|--------|-------|
| $P_1$ | 0      | 15ms  |
| $P_2$ | 10     | 15ms  |
| $P_3$ | 15     | 10ms  |
| $P_4$ | 30     | 10ms  |

1. Se i processi terminano nell'ordine  $P_1, P_3, P_2, P_4$ , quale può essere l'algoritmo di scheduling? (Si trascuri il tempo di latenza del kernel.)
  - RR q=10ms
  - Scheduling non-preemptive
  - SJF
  - RR q=15ms
  - Nessuno dei precedenti
2. (\*) Si calcoli il tempo d'attesa medio per i processi  $P_1, P_2, P_3, P_4$  della tabella sopra nel caso di algoritmo RR con q=10ms.

**Risposta:**

1. a), b), c).

2. Considerando un algoritmo di scheduling RR con  $q=10\text{ms}$  senza tempo di latenza del kernel, abbiamo il seguente diagramma di GANTT:

| P1 | P2 | P1 | P3 | P2 | P4 |
|----|----|----|----|----|----|
| 0  | 10 | 20 | 25 | 35 | 40 |

Quindi il tempo di attesa medio è  $\frac{(20-10)+(35-20)+(25-15)+(40-30)}{4} = \frac{10+15+10+10}{4} = 45/4 = 11,25\text{ms}.$

- Si consideri la seguente situazione, dove  $P_0, P_1, P_2, P_3, P_4$  sono cinque processi in esecuzione, C è la matrice delle risorse correntemente allocate, Max è la matrice del numero massimo di risorse assegnabili ad ogni processo e A è il vettore delle risorse disponibili:

|       | C |   |   |   | Max |   |   |   | Available (A) |
|-------|---|---|---|---|-----|---|---|---|---------------|
|       | A | B | C | D | A   | B | C | D |               |
| $P_0$ | 0 | 2 | 0 | 2 | 0   | 3 | 1 | 2 |               |
| $P_1$ | 0 | 0 | 0 | 0 | 2   | 7 | 5 | 0 |               |
| $P_2$ | 2 | 3 | 5 | 4 | 2   | 3 | 5 | 6 |               |
| $P_3$ | 0 | 4 | 3 | 2 | 0   | 4 | 5 | 2 |               |
| $P_4$ | 0 | 0 | 1 | 5 | 0   | 6 | 5 | 5 |               |

- Calcolare la matrice R delle richieste.
- Il sistema è in uno stato sicuro (safe)?

**Risposta:**

- La matrice R delle richieste è data dalla differenza  $Max - C$ :

|   | R |   |   |   |
|---|---|---|---|---|
|   | A | B | C | D |
| 0 | 1 | 1 | 0 |   |
| 2 | 7 | 5 | 0 |   |
| 0 | 0 | 0 | 2 |   |
| 0 | 0 | 2 | 0 |   |
| 0 | 6 | 4 | 0 |   |

- Sì il sistema è in uno stato sicuro in quanto esiste la sequenza sicura  $\langle P_0, P_2, P_1, P_3, P_4 \rangle$ . Infatti, dapprima si esegue  $P_0$  in quanto  $R_0 \leq A$  ed A diventa quindi  $(1, 7, 3, 2)$ . A questo punto  $R_2 \leq A$  e quindi si esegue  $P_2$  generando il nuovo valore di A:  $(3, 10, 8, 6)$ . Quindi si può mandare in esecuzione  $P_1$  dato che  $R_1 \leq A$  lasciando inalterato A, dato che  $C_1 = (0, 0, 0, 0)$ . In seguito può essere eseguito  $P_3$  ( $R_3 \leq A$ ) aggiornando A al valore  $(3, 14, 11, 8)$ . Infine viene eseguito  $P_4$  ( $R_4 \leq A$ ) generando il valore finale di A =  $(3, 14, 12, 13)$ .

- 1. Un sistema di gestione della memoria con paginazione di quale tipo di frammentazione può soffrire? E un sistema con segmentazione?
- (\*) Supponendo di avere un sistema con quattro frame e sette pagine, addottando una politica di rimpiazzamento basata sul working set model, quanti page fault si verificheranno con la reference string seguente, assumendo  $\Delta = 4$  e di mantenere in memoria esattamente il solo working set?

0 1 6 2 3 2 6 1 2 3 1

(Si assume che i quattro frame siano inizialmente vuoti.)

**Risposta:**

1. Un sistema di gestione della memoria con paginazione può soffrire del problema di frammentazione interna; infatti è difficile che le immagini dei processi in memoria abbiano dimensioni che siano multipli esatti della dimensione delle pagine. Quindi l'ultima pagina sarà soltanto parzialmente utilizzata.

Un sistema con segmentazione invece può soffrire del problema di frammentazione esterna; infatti i segmenti hanno dimensioni variabili e, caricando e rimuovendo processi dalla memoria, può capitare che si formino delle zone di memoria libera tra un segmento e l'altro così piccole da non consentire l'allocazione di un nuovo segmento.

2. Simuliamo il funzionamento dell'algoritmo basato sul working set model:

|   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 6 | 2 | 3 | 2 | 6 | 1 | 2 | 3 | 1 |
| 0 | 1 | 6 | 2 | 3 | 2 | 6 | 1 | 2 | 3 | 1 |
| 0 | 1 | 6 | 2 | 3 | 2 | 6 | 1 | 2 | 3 |   |
| 0 | 1 | 6 | 6 | 3 | 2 | 6 | 1 | 2 |   |   |
| 0 | 1 |   |   |   | 3 |   |   | 6 |   |   |
| P | P | P | P | P |   | P |   | P |   |   |

Si verificano quindi sette page fault.

- Elencare i passi eseguiti dai driver dei dispositivi per servire le richieste.

**Risposta:** I passi fondamentali sono i seguenti:

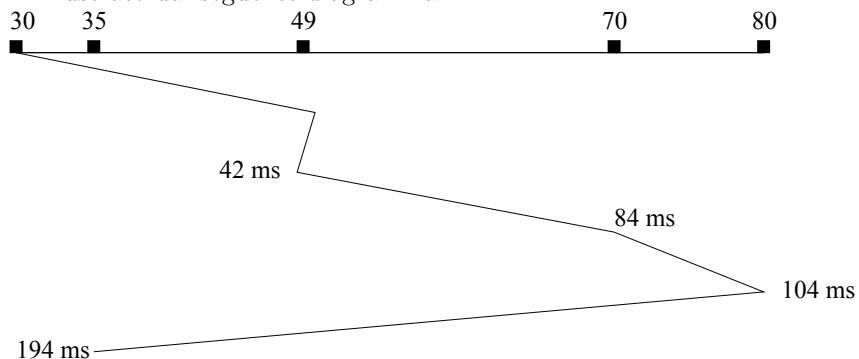
1. controllare i parametri passati,
2. organizzare le richieste in una coda di operazioni che sono soggette a scheduling,
3. eseguire le operazioni, accedendo al controller del dispositivo,
4. mettere il processo in modalità wait (nel caso di I/O interrupt-driven) o attendere la fine dell'operazione in busy-wait,
5. controllare lo stato dell'operazione tramite il controller del dispositivo,
6. restituire il risultato.

- Descrivere cosa si intende per sistema distribuito.

**Risposta:** Un sistema distribuito è un insieme di unità di elaborazione debolmente accoppiate, ovvero, che non condividono clock e memoria e che comunicano tramite diverse linee di comunicazione (es.: reti locali o geografiche).

- (\*) Si consideri un disco gestito con politica SSTF. Inizialmente la testina è posizionata sul cilindro 30, ascendente; lo spostamento ad una traccia adiacente richiede 2 ms. Al driver di tale disco arrivano richieste per i cilindri 70, 49, 35, 80, rispettivamente agli istanti 0 ms, 40 ms 50 ms, 70 ms. Si trascuri il tempo di latenza. In quale ordine vengono servite le richieste?

**Risposta:** L'ordine in cui vengono servite le richieste è 49, 70, 80, 35 come illustrato dal seguente diagramma:



Il punteggio attribuito ai quesiti è il seguente: 3, 3, 3, 3, 2, 2, 2, 5, 3, 3, 4 (totale: 33).