Sistemi Operativi A

12 Giugno 2006

Esercizio 1

Considerare una variante del problema del produttore e consumatore in cui sia presente un solo produttore che scrive il dato prodotto in una variabile condivisa x, e ci siano n consumatori ognuno dei quali deve utilizzare lo stesso dato memorizzato in x (immaginare, ad esempio, la stampa simultanea su n stampanti). Aggiungere le necessarie sincronizzazioni al codice del produttore e del consumatore i-esimo, utilizzando i semafori. Spiegare il funzionamento della soluzione indicata.

```
// "x" e' una variabile condivisa
// "dato" e' una variabile locale

produttore {                          consumatore(i) {
  while(true) {                          while(true) {
    <produci dato>
                                             dato := x
    x := dato;
                                             <consuma dato>
} }                                       }}
```

Spiegazione:

Esercizio 2

Si consideri il seguente processo

```
process Trasferisci{
    conti.incrementaX; conti.decrementaY; conti.stampasomma;
}

monitor conti {

    integer x=10,y=10;                    void decrementaY() {
    condition completamento;                   y = y - 1;
                                          }
    void incrementaX() {
         while (x+y != 20)                void stampasomma() {
             completamento.wait;              print(x+y);
         x = x + 1;    }                   completamento.notify; }
```

Considerare due istanze concorrenti del processo Trasferisci.

1. Quali sono i possibili output?

   Spiegare:

2. Ipotizzare di rimuovere il codice di sincronizzazione

   ```
   while (x+y != 20)
       completamento.wait;      e       completamento.notify;
   ```

   Quali sono i possibili output in questo caso?

   Spiegare:

Esercizio 3

In un sistema, sono pronti per essere eseguiti i 2 processi P1, P2 con tempi di esecuzione (di CPU) 6 e 8, rispettivamente. È inoltre presente il processo P3 che svolge attività di I/O per 3 unità di tempo per poi utilizzare la CPU per un quanto, tornare in I/O per altre 5 unità di tempo usare ancora la CPU per un quanto e terminare.

1. Quale scheduling minimizza il tempo medio di completamento per processo (*turnaround*)? Spiegare e indicare nella tabella sottostante l'esecuzione dei tre processi (scrivere CPU per il processo in esecuzione P per i processi pronti e T per quelli completati)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1																				
P2																				
P3																				

   Tournaround =

2. Quale scheduling, invece, risulta essere più equo? Motivare e illustrare l'esecuzione.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1																				
P2																				
P3																				

   Tournaround =

Esercizio 4

Nel sistema operativo Linux è possibile richiedere al sistema operativo che un determinato processo real-time (soft) venga gestito con scheduling batch FCFS oppure Round Robin. La priorità real-time prevede che i processi abbiano priorità assoluta su tutti gli altri. Si considerino due processi CPU-bound che eseguono una computazione senza I/O di circa un'ora. Ipotizzando che nessun altro processo nel sistema sia attualmente eseguito a priorità real-time, discutere come varia il comportamento del sistema operativo, anche da un punto di vista dell'interfaccia con l'utente, nelle seguenti situazioni:

1. i due processi vengono eseguiti senza modifiche di priorità e scheduling;

2. i due processi vengono eseguiti a priorità massima real-time (soft) con scheduling Round Robin;

3. i due processi vengono eseguiti a priorità massima real-time (soft) con scheduling FCFS.

Esercizio 5

Considerare una matrice A, 10×10, di elementi da 1 byte memorizzata per righe. Le pagine virtuali sono di dimensione 10 bytes e la matrice è memorizzata sequenzialmente a partire dalla prima pagina virtuale. Considerare una scansione in lettura di A per righe e una scansione in lettura di A per colonne. Tralasciando gli accessi alla memoria per il codice del programma (fetch istruzioni) e variabili ausiliarie (indici della matrice), illustrare le due sequenze di accesso alle pagine virtuali

- Sequenza di accesso nella scansione per righe:

  Spiegazione:

- Sequenza di accesso nella scansione per colonne:

  Spiegazione:

Indicare l'allocazione in memoria fisica per i primi 20 accessi delle due scansioni in una memoria di 3 frame con algoritmo LRU:

1. Scansione per righe:

Sequenza																			
Frame 1																			
Frame 2																			
Frame 3																			
Page Fault																			

2. Scansione per colonne:

Sequenza																			
Frame 1																			
Frame 2																			
Frame 3																			
Page Fault																			

3. Commentare brevemente i risultati ottenuti:

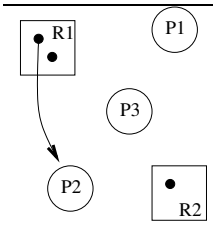
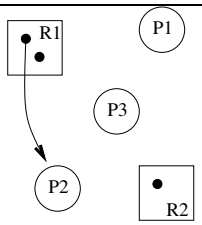
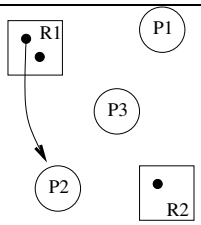
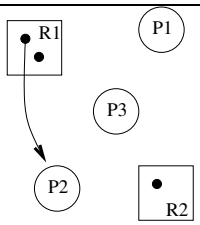
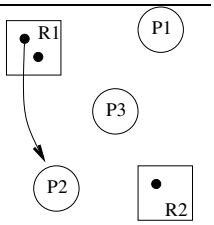
Esercizio 6

Cos'è la tecnica di paginazione su richiesta e perchè è importante nei sistemi multiprogrammati. Illustrare le attività principali necessarie a realizzare tale tecnica, evidenziando quali di queste attività sono svolte dal hardware e quali dal Sistema Operativo.

Esercizio 7

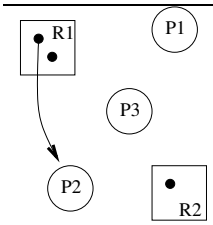
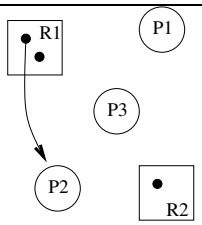
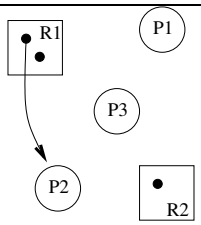
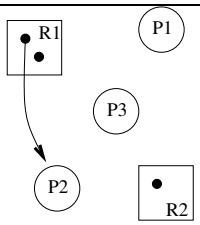
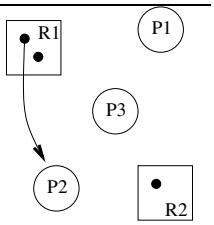
Si consideri un sistema con due risorse di tipo R1 e una risorsa di tipo R2. Sono presenti tre processi: P1, P2 e P3 e il processo P2 ha già assegnata una risorsa di tipo R1. Completare con le opportune frecce il grafo di assegnazione dopo ogni richiesta nei tre casi sotto menzionati:

1. Nessuna gestione del deadlock da parte del sistema operativo

				
P1 chiede 1 R1	P3 chiede 1 R2	P1 chiede 1 R2	P2 chiede 1 R2	P3 chiede 1 R1
Spiegazione:	Spiegazione:	Spiegazione:	Spiegazione:	Spiegazione:

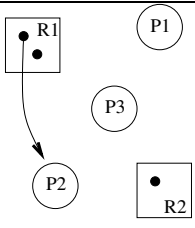
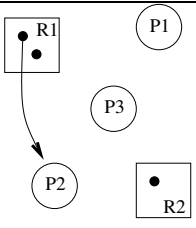
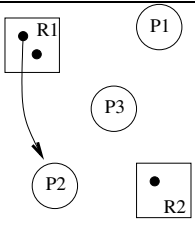
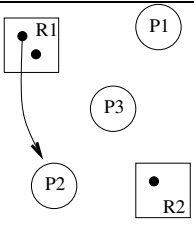
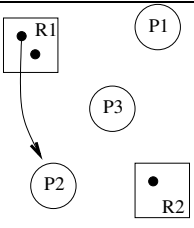
Si verifica una situazione di deadlock? Perchè?

2. Gestione gerarchica delle risorse: R1 < R2

				
P1 chiede 1 R1	P3 chiede 1 R2	P1 chiede 1 R2	P2 chiede 1 R2	P3 chiede 1 R1
Spiegazione:	Spiegazione:	Spiegazione:	Spiegazione:	Spiegazione:

Si verifica una situazione di deadlock? Perchè?

3. Algoritmo del banchiere (le richieste corrispondono a quelle massime dichiarate dai processi)

				
P1 chiede 1 R1	P3 chiede 1 R2	P1 chiede 1 R2	P2 chiede 1 R2	P3 chiede 1 R1
Spiegazione:	Spiegazione:	Spiegazione:	Spiegazione:	Spiegazione:

Si verifica una situazione di deadlock? Perchè?