

# Sistemi Operativi

## 4 febbraio 2013

### Compitino I A

#### IMPORTANTE:

- Si risponda ai seguenti quesiti, giustificando le risposte.
  - Gli studenti del corso di laurea in Informatica risolvano gli esercizi 1–6.
  - Gli studenti del corso di laurea in TWM, che hanno in piano di studi Sistemi Operativi come corso da 6 crediti, risolvano gli esercizi 1–5.
  - Gli studenti del corso di laurea in TWM, che hanno in piano di studi Sistemi Operativi come corso da 9 crediti, risolvano gli esercizi 1–6.
- (a) Che cosa si intende per processi CPU-bound e processi I/O-bound? Quali effetti si possono avere se viene data la precedenza a processi CPU-bound?  
(b) Si diano esempi di algoritmi di scheduling della CPU che privilegiano i processi I/O-bound.

#### Risposta:

- (a) I processi possono essere classificati come I/O-bound quando presentano lunghi periodi di I/O e brevi periodi di calcolo. Al contrario si parla di processi CPU-bound quando si hanno lunghi periodi di intensiva computazione e pochi (possibilmente lunghi) cicli di I/O. Privilegiando i processi CPU-bound si allungano i tempi di risposta percepiti dagli utenti che stanno eseguendo dei processi interattivi (I/O-bound).  
(b) In generale una classe di algoritmi di scheduling della CPU che privilegiano i processi I/O-bound è costituita dagli scheduling con code multiple e con retroazione (feedback) in cui la coda a maggior priorità viene riservata ai processi interattivi, mentre quelli CPU-bound vengono posizionati nelle code a minor priorità. Fra i sistemi operativi di uso comune l'algoritmo di scheduling di Linux è un esempio di algoritmo che favorisce i processi interattivi (I/O-bound) dato che assegna un bonus di priorità dinamica a questi ultimi. Anche lo scheduling di Windows favorisce i processi I/O-bound dato che sceglie sempre dalla coda a priorità maggiore, ma la priorità di un thread utente può essere temporaneamente maggiore di quella base (per effetto delle "spinte") per thread che attendono dati di I/O (spinte fino a +8) e per dare maggiore reattività a processi interattivi (+2).
2. Quali delle seguenti transizioni di stato per un processo possono causare la transizione di stato *waiting* → *ready* per uno o più degli altri processi?
  - (a) Un processo avvia un'operazione di I/O e passa nello stato *waiting*.
  - (b) Un processo termina.
  - (c) Un processo effettua la richiesta per una risorsa e passa nello stato *waiting*.
  - (d) Un processo invia un messaggio.
  - (e) Un processo effettua una transizione di stato *waiting* → *waiting swapped*.

#### Risposta:

- (a) No, non influenza il passaggio da *waiting* a *ready* di un altro processo.
  - (b) Sì se, per esempio, si tratta della terminazione di un processo figlio di cui il padre era in attesa o se un altro processo era in attesa di una risorsa detenuta dal processo terminato.
  - (c) No, non influenza il passaggio da *waiting* a *ready* di un altro processo.
  - (d) Sì: un altro processo poteva essere in attesa di ricevere un messaggio.
  - (e) No, lo swapping su disco di un processo in stato *waiting* non può influenzare il passaggio da *waiting* a *ready* di un altro processo.
3. Si consideri un sistema con scheduling della CPU 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 10 e 15 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:

# Sistemi Operativi

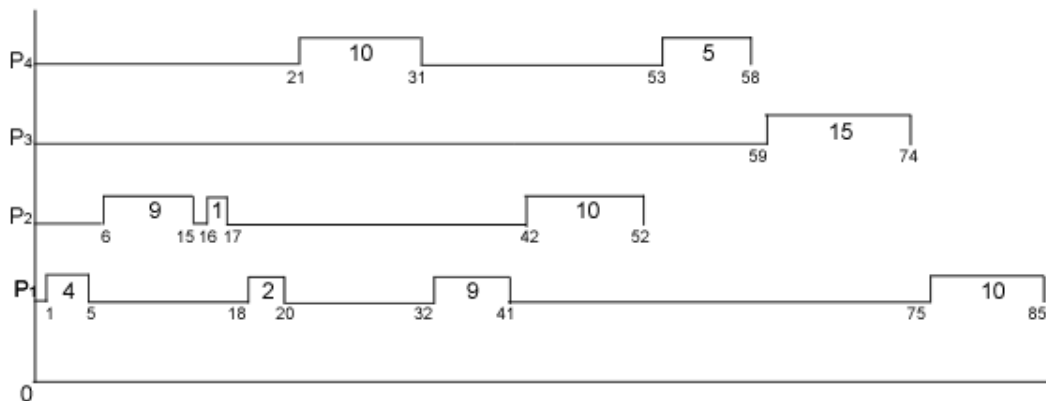
## 4 febbraio 2013

### Compitino I A

	coda	arrivo	burst
$P_1$	B	0	25ms
$P_2$	A	5	20ms
$P_3$	C	15	15ms
$P_4$	A	20	15ms

Si determini il diagramma di GANTT relativo all'esecuzione dei quattro processi, assumendo che il tempo di latenza del kernel sia pari a 1 ms.

**Risposta:** Il diagramma di GANTT relativo all'esecuzione dei quattro processi è il seguente:



4. Che cosa significa “corsa critica”? Si diano esempi in cui si può verificare.

**Risposta:** Per “corsa critica” si intende la situazione in cui più processi accedono concorrentemente agli stessi dati ed il risultato finale dipende dall'ordine di interleaving dei processi. Si tratta di eventi frequenti nei sistemi operativi multitasking, sia per dati in user space sia per le strutture del kernel. Un tipico esempio è l'accesso concorrente al contenuto di un file su disco, di un database ecc.

5. Si vuole evitare che all'interno di un locale ci siano contemporaneamente più di  $N$  utenti. Si completi il seguente monitor, che gestisce l'accesso e l'uscita dal locale.

```
monitor Locale
condition notfull, notempty;
integer spaces, capacity;

procedure enter();
begin
    ...
    ...
    ...
end;

procedure exit();
begin
    ...
    ...
    ...
end;

spaces := N;
capacity := N;
end monitor;
```

**Risposta:**

# Sistemi Operativi

## 4 febbraio 2013

### Compitino I A

```

monitor Locale
  condition notfull, notempty;
  integer spaces, capacity;

  procedure enter();
  begin
    while (spaces = 0) do wait (notfull);
    spaces := spaces - 1;
    signal (notempty);
  end;

  procedure exit();
  begin
    while (spaces = capacity) do wait (notempty);
    spaces := spaces + 1;
    signal (notfull);
  end;

  spaces := N;
  capacity := N;
end monitor;

```

6. 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 da assegnare ad ogni processo e  $A$  è il vettore delle risorse disponibili:

	<u>C</u>				<u>Max</u>			
	A	B	C	D	A	B	C	D
$P_0$	0	0	1	2	0	0	1	2
$P_1$	1	0	0	0	1	7	5	0
$P_2$	1	3	5	4	2	3	5	6
$P_3$	0	6	3	2	0	6	5	2
$P_4$	0	0	1	4	0	6	5	6

<u>Available (A)</u>			
A	B	C	D
1	5	2	0

- (a) Calcolare la matrice  $R$  delle richieste.
- (b) Il sistema è in uno stato sicuro?
- (c) La richiesta  $(0\ 4\ 2\ 0)$  da parte del processo  $P_1$  può essere soddisfatta immediatamente?

**Risposta:**

- (a)  $R = MAX - C$ :

<u>R</u>			
A	B	C	D
0	0	0	0
0	7	5	0
1	0	0	2
0	0	2	0
0	6	4	2

- (b) Il sistema è in uno stato sicuro in quanto esiste la sequenza di esecuzione sicura  $P_0$  (nuovo valore di  $A = (1\ 5\ 3\ 2)$ ),  $P_2$  (nuovo valore di  $A = (2\ 8\ 8\ 6)$ ),  $P_1$  (nuovo valore di  $A = (3\ 8\ 8\ 6)$ ),  $P_3$  (nuovo valore di  $A = (3\ 14\ 11\ 8)$ ) e  $P_4$  (nuovo valore di  $A = (3\ 14\ 12\ 12)$ ).

**Sistemi Operativi**  
**4 febbraio 2013**  
**Compitino I A**

- (c) Se la richiesta (0 4 2 0) da parte del processo  $P_1$  viene soddisfatta immediatamente, si hanno i seguenti nuovi valori per le matrici  $C$ ,  $R$  e  $A$ :

		<u>C</u>				<u>R</u>		
$A$	$B$	$C$	$D$	$A$	$B$	$C$	$D$	
0	0	1	2	0	0	0	0	
1	4	2	0	0	3	3	0	
1	3	5	4	1	0	0	2	
0	6	3	2	0	0	2	0	
0	0	1	4	0	6	4	2	

$$A = (1 \ 1 \ 0 \ 0)$$

La richiesta può essere accettata in quanto esiste la sequenza di esecuzione sicura  $P_0$  (nuovo valore di  $A = (1 \ 1 \ 1 \ 2)$ ),  $P_2$  (nuovo valore di  $A = (2 \ 4 \ 6 \ 6)$ ),  $P_1$  (nuovo valore di  $A = (3 \ 4 \ 6 \ 6)$ ),  $P_3$  (nuovo valore di  $A = (3 \ 10 \ 9 \ 8)$ ) e  $P_4$  (nuovo valore di  $A = (3 \ 10 \ 10 \ 12)$ ).