

Università Ca' Foscari Venezia - Corso di Laurea in Informatica
Sistemi Operativi – prof. Augusto Celentano

Lo scheduling della CPU – Esercizi risolti

1) In un sistema time-sharing con politica di scheduling *round-robin statico senza priorità* sono presenti quattro processi P1-P4 nel seguente stato:

P1 in esecuzione, P2 e P3 pronti (P2 in testa alla coda davanti a P3), P4 in attesa di una operazione di I/O

Descrivere come cambia lo stato del sistema (cioè come cambiano di stato i processi) se a partire dalla situazione data si verificano nell'ordine tutti e soli i seguenti eventi:

Caso 1:

- a) trascorre un quanto di tempo
- b) termina l'operazione di I/O del processo in attesa
- c) il processo in esecuzione chiede una operazione di I/O
- d) trascorre un quanto di tempo
- e) termina l'operazione di I/O per il processo in attesa
- f) il processo in esecuzione termina
- g) trascorre un quanto di tempo

Caso 2:

- a) termina l'operazione di I/O del processo in attesa
- b) trascorre un quanto di tempo
- c) trascorre un quanto di tempo
- d) il processo in esecuzione chiede una operazione di I/O
- e) il processo in esecuzione termina
- f) termina l'operazione di I/O per il processo in attesa
- g) trascorre un quanto di tempo

Soluzione

Nel primo caso il sistema evolve in questo modo:

- a) P1 va in stato di pronto in coda a P3, P2 va in esecuzione
- b) P2 rimane in esecuzione, P4 va in stato di pronto in coda a P1
- c) P2 va in stato di attesa, P3 è il primo processo pronto e va in esecuzione
- d) P3 torna in stato di pronto in coda a P4, P1 va in esecuzione
- e) in attesa c'è il processo P2, che va quindi in stato di pronto in coda a P3
- f) il processo che termina è P1, va in esecuzione P4
- g) P4 va in stato di pronto in coda a P2, P3 va in esecuzione.

Nel secondo caso il sistema evolve così:

- a) P4 va in stato di pronto in coda a P3
- b) P1 va in stato di pronto in coda a P4, P2 va in esecuzione
- c) P2 va in stato di pronto in coda a P1, P3 va in esecuzione
- d) P3 è in esecuzione e va in stato di attesa. P4 è il primo processo pronto e va in esecuzione
- e) P4 termina, il primo processo pronto è P1, che va in esecuzione
- f) in attesa c'è P3, che va in stato di pronto in coda a P2
- g) va in esecuzione P2, mentre P1 va in stato di pronto.

2) In un sistema time-sharing con politica di scheduling *round robin con priorità statiche* sono presenti quattro processi P1-P4 nel seguente stato:

P1 in esecuzione, P2 e P3 pronti, P4 in attesa di un'operazione di I/O.

Le priorità dei processi sono in questa relazione:

$$P1 = P2 > P4 > P3$$

Descrivere, motivando la risposta, come cambia lo stato del sistema (cioè come cambia lo stato dei processi) se a partire dalla situazione data si verificano nell'ordine tutti e soli i seguenti eventi:

- a) trascorre un quanto di tempo
- b) termina l'operazione di I/O del processo in attesa
- c) il processo in esecuzione chiede una operazione di I/O
- d) trascorre un quanto di tempo
- e) il processo in esecuzione chiede una operazione di I/O
- f) trascorre un quanto di tempo
- g) il processo in esecuzione termina

Soluzione

Il sistema evolve in questo modo:

- a) Essendo P1 e P2 della stessa priorità, P2 va in esecuzione al posto di P1, che va in stato di pronto; P1 si posiziona nella coda dei processi pronto davanti a P3 perché ha priorità maggiore.
- b) Il processo P4 va in stato di pronto, posizionandosi davanti a P3 perché meno prioritario. In esecuzione rimane il processo P2.
- c) Il processo P2 dallo stato di esecuzione va in stato di attesa; P1 è il primo processo pronto per priorità e va in esecuzione.
- d) Il processo P1 rimane in esecuzione perché gli altri processi pronti, P3 e P4, hanno priorità minore.
- e) Il processo P1 dallo stato di esecuzione va in stato di attesa; nella coda dei processi pronti ci sono i processi P3 e P4; il più prioritario è il processo P4, che va in esecuzione.
- f) Il processo P4 rimane in esecuzione perché l'altro processo pronto, P3, ha priorità minore.
- g) Il processo P4 termina e va in esecuzione P3, unico processo nella coda dei processi pronti.

L'evoluzione del sistema è riassunta in questa tabella:

Evento	Esecuzione	Pronto	Attesa
Stato iniziale	P1	P2 > P3	P4
a	P2	P1 > P3	P4
b	P2	P1 > P4 > P3	–
c	P1	P4 > P3	P2
d	P1	P4 > P3	P2
e	P4	P3	P2, P1
f	P4	P3	P2, P1
g	P3	–	P2, P1

3) In un sistema time-sharing con politica di scheduling round robin a priorità statiche sono presenti quattro processi P1 - P4, nel seguente stato:

P1, P3 pronti; P2 in esecuzione; P4 in attesa su operazione di I/O

I processi P1, P2 e P4 hanno priorità uguale, e più alta del processo P3. Descrivere come cambia lo stato del sistema se, a partire da questa situazione, si verificano, nell'ordine, gli eventi seguenti:

- a. trascorre un quanto di tempo
- b. si completa l'operazione di I/O per P4
- c. trascorre un quanto di tempo
- d. il processo in esecuzione chiede una operazione di I/O
- e. il processo in esecuzione termina

Soluzione

Lo stato di modifica come segue:

- a. P2 va in stato di pronto davanti a P3 (P3 è meno prioritario), P1 va in esecuzione
- b. P4 va in stato di pronto in coda tra P2 e P3
- c. P1 va in pronto in coda tra P4 e P3, P2 va in esecuzione
- d. P2 va in stato di attesa, P4 (primo processo pronto) va in esecuzione
- e. P4 termina, P1 va in esecuzione, P3 rimane in stato di pronto, P2 è in attesa di I/O

4) In un sistema real-time (con priorità) sono presenti quattro processi P1 - P4, nel seguente stato:

P1 in esecuzione; P2, P3 pronti; P4 in attesa su operazione di I/O

I processi P1 e P4 hanno priorità uguale, e più alta della priorità dei processi P2 e P3. Descrivere come cambia lo stato del sistema se, a partire da questa situazione, si verificano, nell'ordine, gli eventi seguenti:

- a. trascorre un quanto di tempo
- b. Il processo in esecuzione termina
- c. si completa l'operazione di I/O per il processo P4
- d. il processo in esecuzione chiede una operazione di I/O
- e. trascorre un quanto di tempo

Soluzione

Lo stato di modifica come segue:

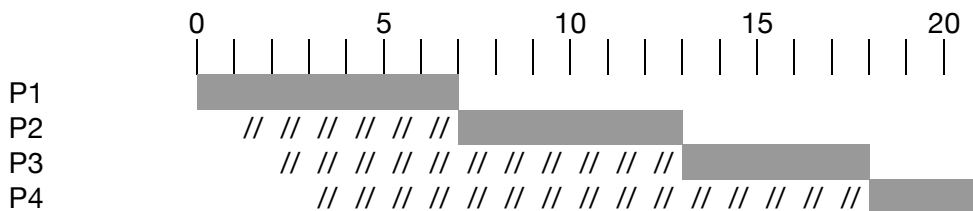
- a. P1 rimane in esecuzione, poiché gli altri due processi pronti sono meno prioritari. Lo scadere di un quanto di tempo non ha effetto in un sistema real-time a meno che non comporti la modifica delle priorità (priorità dinamiche)
- b. P1 termina e va in esecuzione il processo P2, primo della coda dei processi pronti e della stessa priorità di P3.
- c. P4 va in stato di pronto ma poiché è più prioritario di P2 e P3 va immediatamente in esecuzione, mentre il processo P2 va in stato di pronto in coda a P3.
- d. P4 va in stato di attesa e l'esecuzione passa al processo P3
- e. Essendo P2 alla stessa priorità di P3, l'esecuzione di P3 non viene interrotta. Usualmente però i sistemi real-time introducono uno scheduling round-robin tra i processi alla stessa priorità per evitare il monopolio della CPU da parte di un solo processo a scapito degli altri della stessa priorità. Andrebbe quindi in esecuzione il processo P2 mentre P3 verrebbe portato nella coda dei processi pronti, al livello di priorità che gli compete (in questo caso è l'unico processo pronto).

5) Quattro processi (da P1 a P4) entrano nello stato di pronto a distanza di 100 mS l'uno dall'altro e nell'ordine da P1 a P4. Si supponga che i processi siano CPU-bound, e che il tempo stimato del prossimo CPU-burst sia rispettivamente di 700, 600, 500 e 300 mS. I processi sono alla stessa priorità e dopo questo CPU burst inizia un'operazione di I/O molto lunga. Per le politiche di scheduling elencate di seguito, determinare il tempo di attesa per ciascun processo (considerando solo il CPU burst) e il tempo di attesa medio dei quattro processi:

- First Come First Served
- Shortest Process Next
- Round-robin con un quanto di tempo di 300 millisecondi

Soluzione

La politica di scheduling *First Come First Served* prevede che i processi vengano serviti come descritto nello schema seguente. Gli intervalli di tempo tratteggiati rappresentano i processi nello stato di pronto, quelli in colore scuro rappresentano i periodi di esecuzione.

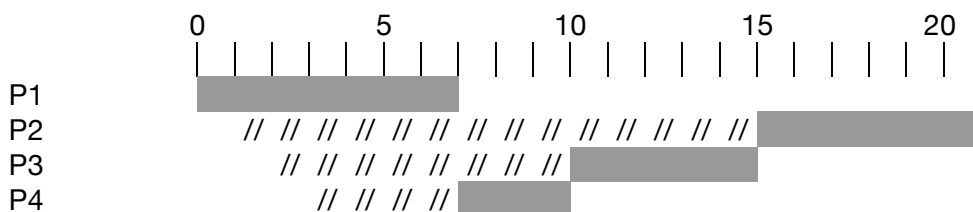


Da questo schema derivano i seguenti tempi di attesa nello stato di pronto:

$P1 = 0$
 $P2 = 600 \text{ mS}$
 $P3 = 1100 \text{ mS}$
 $P4 = 1500 \text{ mS}$

tempo di attesa medio = $(0+600+1100+1500)\text{mS}/4 = 3200\text{mS}/4 = 800 \text{ mS}$

Per la politica *Shortest Process Next* si osserva che il primo processo a partire è sempre il processo P1, anche se non è il più breve, poiché al tempo 0 gli altri processi non sono ancora entrati nello stato di pronto. Si ha quindi il seguente schema:



da cui derivano i seguenti tempi di attesa nello stato di pronto:

$P1 = 0$
 $P2 = 1400 \text{ mS}$
 $P3 = 800 \text{ mS}$
 $P4 = 400 \text{ mS}$

tempo di attesa medio = $(0+1400+800+400)\text{mS}/4 = 2600\text{mS}/4 = 650 \text{ mS}$

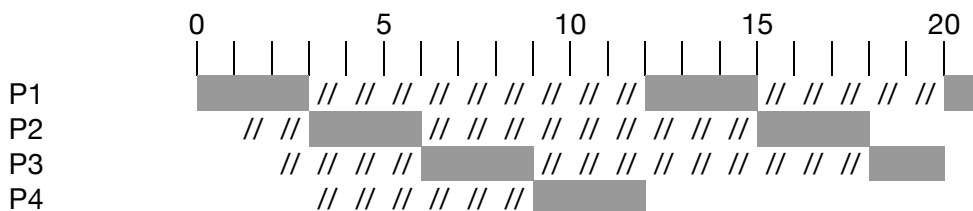
Per la politica *Round Robin* è necessario considerare tutti i periodi trascorsi nello stato di pronto, quindi anche quelli che intercorrono tra un quanto di tempo in cui il processo è in esecuzione e il successivo, fino a terminazione del burst di CPU.

Bisogna anche considerare che cosa succede allo scadere del primo turno di RR, dopo i primi 300 mS: dal momento che tutti gli eventi, anche se contemporanei da un punto di vista teorico, sono in realtà elaborati in modo sequenziale, allo scadere dei primi 300 mS il processo P4 potrebbe essere inserito nella coda dei processi pronti *prima* o *dopo* l'interruzione di P1 da parte dello scheduler. Ricordiamo che un processo viene posto in stato di pronto solo a seguito di uno dei seguenti eventi: (1) viene creato, (2) termina un'attesa su un evento esterno, (3) viene interrotto dallo scheduler mentre è in esecuzione. Il terzo caso non è coerente con i dati del problema; negli altri due casi l'evento viene servito da una system call (nel caso della creazione) o da una risposta a interrupt (nel caso della fine dell'attesa), eseguite in modo sequenziale rispetto all'esecuzione dello scheduler.

Se il processo P4 entra nello stato di pronto prima che lo scheduler interrompa l'esecuzione del processo P1, dopo l'interruzione di P1 la situazione della coda dei processi pronti è la seguente:

--> [P1 | P4 | P3 | P2] -->

Si ha quindi il seguente comportamento:



da cui derivano i seguenti tempi di attesa nello stato di pronto:

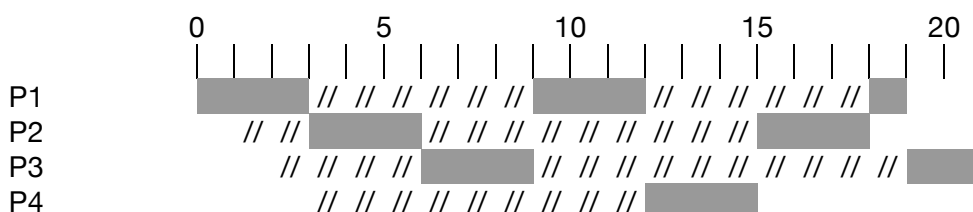
$P1 = 900 \text{ mS} + 500 \text{ mS} = 1400 \text{ mS}$
 $P2 = 200 \text{ mS} + 900 \text{ mS} = 1100 \text{ mS}$
 $P3 = 400 \text{ mS} + 900 \text{ mS} = 1300 \text{ mS}$
 $P4 = 600 \text{ mS}$

tempo di attesa medio = $(1400 + 1100 + 1300 + 600) \text{ mS} / 4 = 4400 \text{ mS} / 4 = 1100 \text{ mS}$.

Se invece il processo P4 entra nello stato di pronto dopo che lo scheduler ha interrotto l'esecuzione del processo P1, dopo l'interruzione di P1 la situazione della coda dei processi pronti è la seguente:

--> [P4 | P1 | P3 | P2] -->

Si ha quindi il seguente comportamento:



da cui derivano i seguenti tempi di attesa nello stato di pronto:

$P1 = 600 \text{ mS} + 600 \text{ mS} = 1200 \text{ mS}$
 $P2 = 200 \text{ mS} + 900 \text{ mS} = 1100 \text{ mS}$
 $P3 = 400 \text{ mS} + 1000 \text{ mS} = 1400 \text{ mS}$
 $P4 = 900 \text{ mS}$

tempo di attesa medio = $(1200 + 1100 + 1400 + 900) \text{ mS} / 4 = 4600 \text{ mS} / 4 = 1150 \text{ mS}$.

6) Quattro processi (da P1 a P4) entrano nello stato di pronto a distanza di 100 mS l'uno dall'altro e nell'ordine da P1 a P4. I processi hanno stessa priorità. Si supponga che il tempo stimato del prossimo CPU-burst di ogni processo sia rispettivamente di 600, 300, 500 e 200 mS. Dopo questo CPU burst inizia per ogni processo un'operazione di I/O molto lunga. Per le politiche di scheduling elencate di seguito, determinare il tempo di attesa per ciascun processo nello stato di pronto, (considerando solo il CPU burst) e il tempo di attesa medio dei quattro processi:

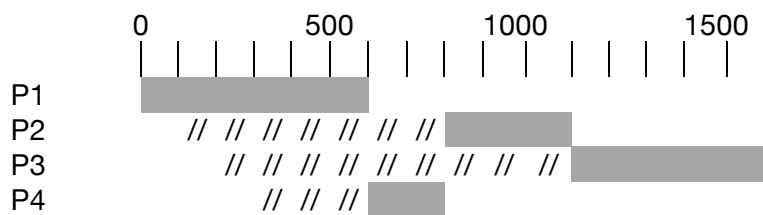
- Shortest Process Next
- First Come First Served
- Round-robin con un quanto di tempo di 400 mS
- Round-robin con un quanto di tempo di 200 mS

Soluzione

Negli schemi seguenti le parti in grigio rappresentano i processi in esecuzione, le parti tratteggiate rappresentano i processi nello stato di pronto.

a. *Shortest Process Next*: manda in esecuzione i processi dando precedenza a quelli con il CPU burst minore.

Il primo processo a partire è sempre il processo P1, anche se non è il più breve, poiché al tempo 0 gli altri processi non sono ancora entrati nello stato di pronto. Al tempo 600mS, quando P1 termina il suo CPU burst, gli altri processi sono tutti nello stato di pronto per cui verranno eseguiti nell'ordine dal più breve al più lungo:

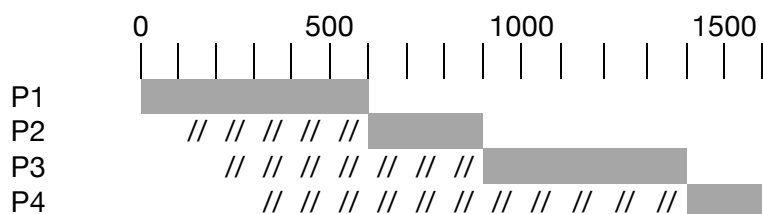


da cui i seguenti tempi di attesa nello stato di pronto:

P1 = 0
 P2 = 700 mS
 P3 = 900 mS
 P4 = 300 mS

tempo di attesa medio = $(0+700+900+300)\text{mS}/4 = 1900\text{mS}/4 = 475$ mS

b. *First Come First Served*: manda in esecuzione i processi nell'ordine in cui arrivano, per l'intera durata del loro CPU burst.



da cui i seguenti tempi di attesa nello stato di pronto:

P1 = 0

P2 = 500 mS

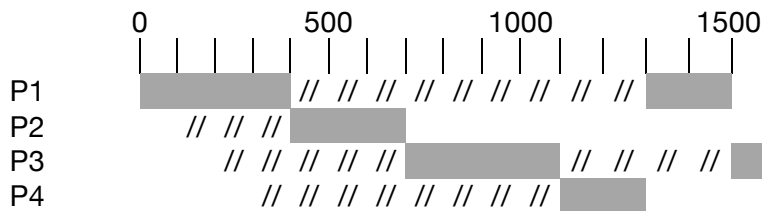
P3 = 700 mS

P4 = 1100 mS

tempo di attesa medio = $(0+500+700+1100)\text{mS}/4 = 2300\text{mS}/4 = 575$
mS

c. *Round-robin* con un quanto di tempo di 400 mS: alterna i processi in una coda circolare allo scadere del quanto di tempo.

Per il calcolo del tempo di attesa è necessario considerare tutti i periodi trascorsi nello stato di pronto, quindi anche quelli che intercorrono tra un quanto di tempo e l'altro. L'evoluzione del sistema è la seguente:



da cui i seguenti tempi di attesa nello stato di pronto:

P1 = 900 mS

P2 = 300 mS

P3 = 900 mS

P4 = 800 mS

tempo di attesa medio = $(900+300+900+800)\text{mS}/4 = 2900\text{mS}/4 = 725$
mS

d. *Round-robin* con un quanto di tempo di 200 mS

In questo caso si deve osservare che il tempo di ingresso nella coda di pronto del processo P3 coincide con lo scadere del primo intervallo di tempo di 200 mS su cui si basa il meccanismo di round-robin. Nella realtà, i due eventi non saranno contemporanei, e il verificarsi dell'uno prima dell'altro modifica la situazione della coda dei processi pronti, e quindi l'evoluzione del sistema.

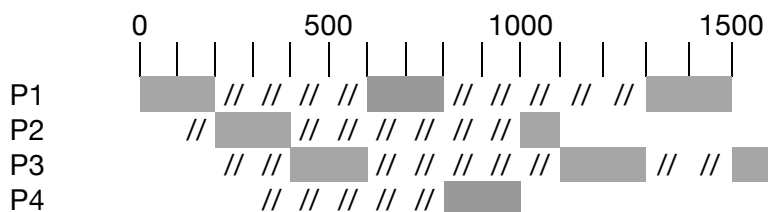
Se il processo P3 entra nello stato di pronto prima che lo scheduler interrompa l'esecuzione del processo P1, portandolo nello stato di pronto, la situazione della coda dei processi pronti è:

--> [P1 | P3 | P2] -->

Quindi andrà in esecuzione il processo P2, P3 diventerà il primo processo nella coda e sarà eseguito al terzo turno di round-robin. Successivamente, allo scadere del secondo quanto di tempo sarà già presente anche il processo P4 e la situazione della coda dei processi pronti sarà:

--> [P2 | P4 | P1 | P3] -->

L'esecuzione complessiva è quindi la seguente:



da cui i seguenti tempi di attesa nello stato di pronto:

P1 = 900 mS

P2 = 700 mS

P3 = 900 mS

P4 = 500 mS

tempo di attesa medio = $(900+700+900+500)\text{mS}/4 = 3000\text{mS}/4 = 750 \text{ mS}$

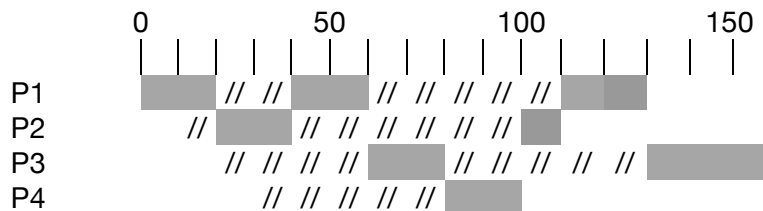
Se invece il processo P3 entra nella coda dei processi pronti dopo che lo scheduler ha interrotto il processo P1, la situazione della coda sarà la seguente:

--> [P3 | P1 | P2] -->

Quindi andrà in esecuzione il processo P2, ma diventerà primo processo nella coda dei processi pronti il processo P1, interrotto e riportato nella coda un istante prima dell'ingresso in coda del processo P3. Successivamente, allo scadere del secondo quanto di tempo, entrato anche P4 nella coda dei processi pronti la situazione sarà la seguente:

--> [P2 | P4 | P3 | P1] -->

L'esecuzione complessiva è quindi la seguente:



da cui i seguenti tempi di attesa nello stato di pronto:

P1 = 700 mS

P2 = 700 mS

P3 = 900 mS

P4 = 500 mS

tempo di attesa medio = $(700+700+900+500)\text{mS}/4 = 2800\text{mS}/4 = 700 \text{ mS}$

7) Quattro processi P1-P4, che hanno la stessa priorità, si trovano contemporaneamente all'istante t_0 nello stato di pronto. I tempi stimati dei loro prossimi CPU-burst sono rispettivamente di 70, 40, 30 e 60 mS, cui segue, per ogni processo, un'attesa su evento esterno molto lungo. Allo stesso istante t_0 il processo in esecuzione esegue una system call ed entra nello stato di attesa per un periodo molto lungo.

- a. Illustrare come procede l'esecuzione per le politiche di scheduling elencate di seguito e calcolare per ciascun processo il tempo di permanenza nello stato di pronto a partire dall'istante t_0 , (considerando solo il tempo necessario a completare il CPU burst):
 - Shortest Process Next
 - First Come First Served
 - Round-robin con un quanto di tempo di 40 mS
- b. Discutere brevemente come cambia l'esecuzione (tralasciando il calcolo dei tempi di permanenza nello stato di pronto) se i processi hanno priorità differenti secondo queste relazioni:

$$P1 = P4 < P3 < P2$$

Soluzione

a - SPN

Ordine di esecuzione: P3 - P2 - P4 - P1

Tempi di attesa: P1 = 130mS, P2 = 30mS, P3 = 0, P4 = 70mS

a - FCFS

Ordine di esecuzione : P1 - P2 - P3 - P4

Tempi di attesa: P1 = 0, P2 = 70mS, P3 = 110 mS, P4 = 140 mS

a - RR

Ordine di esecuzione: P1 (40), P2 (40), P3 (30), P4 (40), P1 (30), P4 (20)

Tempi di attesa: P1 = 0 + 150 mS, P2=40 mS, P3=80+40=120 mS, P4=110+30=140 mS

b - SPN

L'ordine di esecuzione segue la priorità poi, a parità di priorità, la durata. L'ordine di esecuzione quindi diventa:

P2 - P3 - P4 - P1

b - FCFS

L'ordine di esecuzione segue la priorità, a parità di priorità l'ordine di arrivo. L'ordine di esecuzione quindi diventa:

P2 - P3 - P1 - P4

b - RR

L'ordine di esecuzione segue la priorità L'ordine di esecuzione quindi diventa (tra parentesi il quanto di tempo utilizzato):

P2(40mS), P3(30mS), P1(40mS), P4 (40mS), P1(30mS), P4(20mS)

8) La politica di scheduling Shortest Process Next riporta il tempo di attesa minore. E' sempre vero?

Risposta

Sì, ma solo se i processi sono tutti contemporaneamente nello stato di pronto

Supponiamo di avere n processi, che richiedono rispettivamente un tempo di esecuzione del prossimo CPU burst $T_1 < T_2 < \dots < T_n$. Il tempo di attesa del processo T_i è 0 per $i=1$, e la somma dei tempi di esecuzione dei processi $1 \dots i-1$ per $i > 1$.

Il tempo medio di attesa è quindi:

$$(0 + (T_1) + (T_1+T_2) + \dots + (T_1+T_2 + \dots + T_{n-1})) / n = ((n-1)T_1 + (n-2)T_2 + \dots T_{n-1}) / n$$

Nella sommatoria, il tempo che compare col fattore più alto è T_1 , che è il tempo minore. Il tempo più elevato contribuisce alla somma col fattore più piccolo.

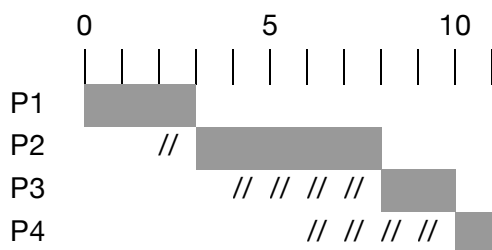
Se i processi non sono tutti nello stato di pronto, quanto detto sopra non è più vero: l'esecuzione di un processo che ha un CPU burst breve potrebbe essere rimandato di molto se al momento in cui entra nello stato di pronto è in esecuzione un processo con un CPU burst molto lungo. In questo caso una politica Round Robin potrebbe dare risultati migliori, perché può interrompere il processo in esecuzione consentendo anche a quello con CPU burst breve di progredire (e di terminare, se è più breve del quanto di tempo di Round Robin).

9) Quattro processi (da P1 a P4) chiedono l'esecuzione ed entrano nello stato di pronto a partire da un istante t_0 a distanza di 200 mS l'uno dall'altro nell'ordine da P1 a P4. Si supponga che il tempo stimato del loro primo CPU-burst sia rispettivamente di 300, 500, 200 e 100 mS. I processi sono alla stessa priorità e dopo questo CPU burst i processi entrano in uno stato di attesa molto lungo rispetto ai tempi sopra indicati. Per le politiche di scheduling elencate di seguito, determinare il tempo totale di permanenza di ogni processo nello stato di pronto prima della terminazione del CPU burst, e indicare qual è la politica di scheduling che determina il minor tempo medio per l'insieme dei quattro processi:

- First Come First Served
- Shortest Process Next
- Round-robin con un quanto di tempo di 200 millisecondi

Soluzione

La politica di scheduling First Come First Served prevede che i processi vengano serviti come descritto nella schema seguente. Le parti tratteggiate rappresentano i processi nello stato di pronto.

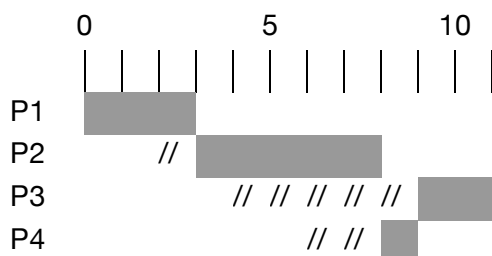


da cui i seguenti tempi di attesa nello stato di pronto:

$P1 = 0$
 $P2 = 100 \text{ mS}$
 $P3 = 400 \text{ mS}$
 $P4 = 400 \text{ mS}$

tempo di attesa medio = $(0+100+400+400)\text{mS}/4 = 900\text{mS}/4 = 225 \text{ mS}$

Per la politica Shortest Process Next si osserva che il primo processo a partire è sempre il processo P1, anche se non è il più breve, poiché al tempo 0 gli altri processi non sono ancora entrati nello stato di pronto. Al tempo 300mS, quando P1 termina il suo CPU burst, solo P2 è in stato di pronto, quindi verrà eseguito. Al tempo 800 mS, terminato il CPU burst di P2, sono pronti sia P3 sia p4, e viene seguito prima P4 perché il suo CPU burst stimato è minore:



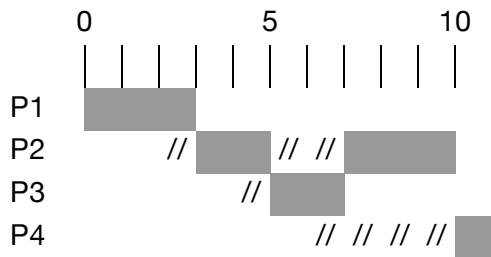
da cui i seguenti tempi di attesa nello stato di pronto:

$P1 = 0$
 $P2 = 100 \text{ mS}$
 $P3 = 500 \text{ mS}$
 $P4 = 200 \text{ mS}$

tempo di attesa medio = $(0+100+500+200)\text{mS}/4 = 800\text{mS}/4 = 200 \text{ mS}$

Per la politica Round Robin è necessario considerare tutti i periodi trascorsi nello stato di pronto, quindi anche quelli che intercorrono tra un quanto di tempo e l'altro.

Al termine del primo quanto di tempo, quando appare anche il processo P2, si hanno due possibilità, dovute alla serializzazione della gestione dei diversi eventi anche se questi sono teoricamente contemporanei: se il processo P2 non è ancora entrato nello stato di pronto, il processo P1 torna in esecuzione poiché è l'unico processo pronto e completa così il suo CPU burst, mentre P2, terminato il suo CPU burst, tornerà in coda dietro a P3 prima che si presenti il processo P4:



da cui i seguenti tempi di attesa nello stato di pronto:

$$P1 = 0$$

$$P2 = 300 \text{ mS}$$

$$P3 = 100 \text{ mS}$$

$$P4 = 400 \text{ mS}$$

$$\text{tempo di attesa medio} = (0+300+100+400)\text{mS}/4 = 800 \text{ mS}/4 = 200\text{mS}$$

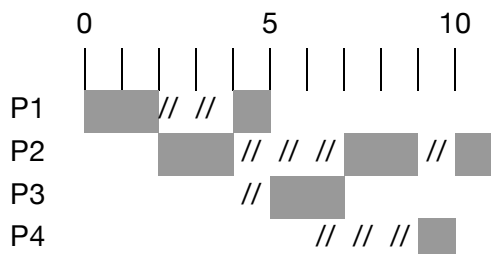
Se invece il processo P2 viene inserito nella coda dei processi pronti prima di interrompere il processo P1, ne prenderà il posto fino al tempo $t_0 + 400 \text{ mS}$. A questo punto possono verificarsi due situazioni: il processo P3 entra in stato di pronto prima che il processo P2 venga interrotto (caso a), oppure dopo (caso b), portando la coda dei processi pronti negli stati seguenti:

caso a) $\rightarrow [P2 \mid P3 \mid P1] \rightarrow$

caso b) $\rightarrow [P3 \mid P2 \mid P1] \rightarrow$

In entrambi i casi il primo processo pronto è P1, che quindi tornerà in esecuzione completando il suo CPU burst al tempo $t_0 + 500 \text{ mS}$. L'esecuzione procede nei due casi come segue:

Caso a)



da cui i seguenti tempi di attesa nello stato di pronto:

$$P1 = 200$$

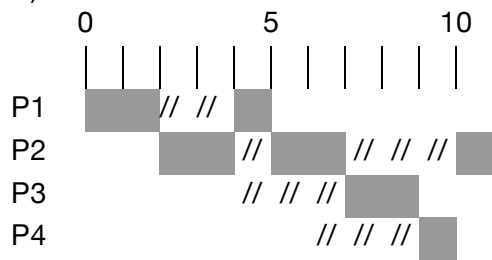
$$P2 = 400 \text{ mS}$$

$$P3 = 100 \text{ mS}$$

$$P4 = 300 \text{ mS}$$

$$\text{tempo di attesa medio} = (200+400+100+300)\text{mS}/4 = 1000 \text{ mS}/4 = 250\text{mS}$$

Caso b)



da cui i seguenti tempi di attesa nello stato di pronto:

P1 = 200

P2 = 400 mS

P3 = 300 mS

P4 = 300 mS

tempo di attesa medio = $(200+400+300+300)\text{mS}/4 = 1200 \text{ mS}/4$
 $= 300\text{mS}$

La politica di scheduling che minimizza il tempo medio è la politica Shortest Process Next, uguagliata dalla politica Round Robin in una particolare serializzazione di eventi.

10) Cinque processi (da P1 a P5) chiedono l'esecuzione contemporaneamente. Il loro tempo stimato di CPU burst è di 90, 50, 40, 30 e 70 millisecondi, e i processi sono alla stessa priorità. Per le politiche di scheduling elencate di seguito, determinare il tempo di attesa per ciascun processo, e dire quale è la soluzione che minimizza il tempo medio di attesa:

- First Come First Served
- Shortest Process Next
- Round Robin, con quanto di tempo di 15 mSec

Soluzione

Per le politiche FCFS e SPN il tempo di attesa di ogni processo è la somma dei tempi di esecuzione dei processi che lo precedono.

FCFS: P1 = 0

P2 = 90 mS

P3 = 90 + 50 = 140 mS

P4 = 90 + 50 + 40 = 180 mS

P5 = 90 + 50 + 40 + 30 = 210 mS

Tempo medio = $(0+90+140+180+210) / 5 = 620 / 5 = 124 \text{ mS}$

SPN: L'ordine di esecuzione è P4, P3, P2, P5 e P1. Si hanno quindi i seguenti tempi di attesa:

P1 = 30 + 40 + 50 + 70 = 190 mS

P2 = 30 + 40 = 70 mS

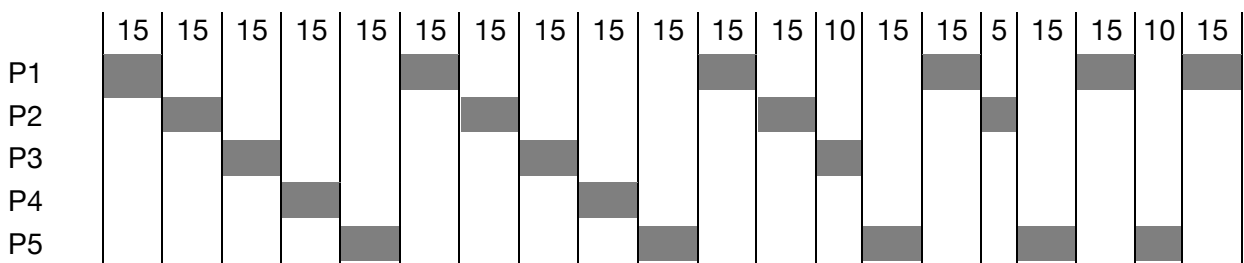
P3 = 30 mS

P4 = 0

P5 = 30 + 40 + 50 = 120 mS

Tempo medio = $(190+70+30+0+120 = 410) / 5 = 82 \text{ mS}$

RR: L'alternarsi nell'esecuzione è dato dal seguente diagramma:



dal quale risultano i seguenti tempi di attesa, contando anche quelli tra due quanti di tempi di esecuzione:

P1 = 190 mS, P2 = 175 mS, P3 = 150 mS, P4 = 105 mS, P5 = 195 mS.

Tempo medio = $(190+175+150+105+195 = 815) / 5 = 163 \text{ mS}$

Il tempo medio di attesa è minimo con la politica Shortest Process Next (come previsto, essendo i processi contemporanei).