SISTEMI OPERATIVI, CORSI A e B - SESTO APPELLO - 14/9/2006

COGNOME ..... NOME ..... MATRICOLA ..... CORSO |A| |B|

ESERCIZIO A-1 (4 punti)

Un sistema con processi A, B, C, D,E e risorse dei tipi R1, R2, R3, R4, rispettivamente di molteplicità [3, 6, 5, 8], ha raggiunto lo stato mostrato nelle tabelle seguenti che è SICURO.

MOLTEPLICITA' →

R1	R2	R3	R4
3	6	5	8

ASSEGNAZIONE ATTUALE →

	R1	R2	R3	R4
A			1	1
B	1			2
C		1	1	1
D		2	2	1
E	2		1	1

	R1	R2	R3	R4
A	1		2	
B			1	2
C	1	2		1
D		2		1
E	1	2	1	

ESIGENZA ATTUALE

DISPONIBILTA' ATTUALE

R1	R2	R3	R4
0	3	0	2

Si chiede:

1) Se il processo B richiede in questo stato due risorse di tipo R4 il sistema resta in uno stato sicuro?
2) Se invece il processo C richiede una risprsa di tipo R4 il sistema resta in uno stato sicuro?

SOLUZIONE

1) Sicuro [SI/NO] ? NO

Stato raggiunto dopo aver assegnato due risorse di tipo R4 al processo B:

MOLTEPLICITA' →

R1	R2	R3	R4
3	6	5	8

ASSEGNAZIONE ATTUALE →

	R1	R2	R3	R4
A			1	1
B	1			4
C		1	1	1
D		2	2	1
E	2		1	1

	R1	R2	R3	R4
A	1		2	
B			1	
C	1	2		1
D		2		1
E	1	2	1	

ESIGENZA ATTUALE

DISPONIBILTA' ATTUALE

R1	R2	R3	R4
0	3	0	0

Nessun processo può terminare in questo stato, quindi lo stato è insicuro.

2) Sicuro [SI/NO] ? SI

Stato raggiunto dopo aver assegnato una risorsa di tipo R4 al processo C:

MOLTEPLICITA' →

R1	R2	R3	R4
3	6	5	8

ASSEGNAZIONE ATTUALE →

	R1	R2	R3	R4
A			1	1
B	1			2
C		1	1	2
D		2	2	1
E	2		1	1

	R1	R2	R3	R4
A	1		2	
B			1	2
C	1	2		
D		2		1
E	1	2	1	

ESIGENZA ATTUALE

DISPONIBILTA' ATTUALE

R1	R2	R3	R4
0	3	0	1

Il processo ...D …può terminare, il vettore delle risorse disponibili diventa:

0	5	2	2

Il processo ...B …può terminare, il vettore delle risorse disponibili diventa:

SISTEMI OPERATIVI, CORSI A e B - SESTO APPELLO - 14/9/2006

COGNOME ..... NOME ..... MATRICOLA ..... CORSO |A| |B|

1	5	2	4

Il processo ...A …può terminare, il vettore delle risorse disponibili diventa:

1	5	3	5

Il processo ...C …può terminare, il vettore delle risorse disponibili diventa:

1	6	4	7

Il processo ...E …può terminare, il vettore delle risorse disponibili diventa:

3	6	5	8

ESERCIZIO A-2 (4 punti)

Un sistema che gestisce il processore combinando le politiche a priorità e RoundRobin con la tecnica delle code multiple (una coda FIFO per ogni valore di priorità; i processi pronti di uguale priorità sono inseriti in una stessa coda; il processore viene assegnato al processo che occupa la prima posizione nella coda non vuota di massima priorità; ai processi pronti di uguale priorità si applica la politica Round Robin).
Il quanto di tempo è di *5 msec*.
La politica prevede il prerilascio, che avviene immediatamente dopo l'evento che lo provoca, senza attendere l'esaurimento del quanto di tempo corrente. Quando un processo va in esecuzione gli viene assegnato un intero quanto di tempo, indipendentemente dal tempo consumato nel precedente turno di esecuzione.
Al tempo T, nel sistema sono presenti i seguenti processi:

- Processo A, con priorità 1, che al tempo t è pronto;
- Processo B, con priorità 3, che al tempo t è in stato di attesa;
- Processo C, con priorità 2, che al tempo t è pronto;
- Processo D, con priorità 2, che al tempo t è in stato di attesa.
- Processo E, con priorità 2, che al tempo t è pronto;

La composizione delle tre code al tempo T è la seguente:

coda 1 (priorità massima=3): - coda 2: C->E coda 3 (priorità minima=1): A

Si chiede quale è il processo in esecuzione e la composizione delle 3 code al tempo T+14
se si verificano le seguenti sequenze di eventi (**da considerare in alternativa**):

1. Al tempo T+3 si sospende il processo in esecuzione, al tempo T+9 si sospende il processo in esecuzione, al tempo T+12 viene riattivato il processo B;
2. Al tempo T+7 viene riattivato il processo B, al tempo T+11 si sospende il processo in esecuzione, al tempo T+12 viene riattivato il processo D;

SOLUZIONE

1. Al tempo T:	in esecuzione C	Coda 1: -	Coda 2: E	Coda 3: A
Al tempo T+ 3:	in esecuzione E	Coda 1: -	Coda 2: -	Coda 3: A
Al tempo T+ 9 :	in esecuzione A	Coda 1: -	Coda 2: -	Coda 3: -
Al tempo T+ 12:	in esecuzione B	Coda 1: -	Coda 2: -	Coda 3: A
Al tempo T+ 14:	in esecuzione B	Coda 1: -	Coda 2: -	Coda 3: A

2. Al tempo T:	in esecuzione C	Coda 1: -	Coda 2: E	Coda 3: A
Al tempo T+ 5:	in esecuzione E	Coda 1: -	Coda 2: C	Coda 3: A
Al tempo T+ 7:	in esecuzione B	Coda 1: -	Coda 2: C->E	Coda 3: A
Al tempo T+ 11:	in esecuzione C	Coda 1: -	Coda 2: E	Coda 3: A
Al tempo T+ 12:	in esecuzione C	Coda 1: -	Coda 2: E->D	Coda 3: A
Al tempo T+ 14:	in esecuzione C	Coda 1: -	Coda 2: E->D	Coda 3: A

ESERCIZIO A-3 (3 punti)

SISTEMI OPERATIVI, CORSI A e B - SESTO APPELLO - 14/9/2006

COGNOME ..... NOME ..... MATRICOLA ..... CORSO |A| |B|

In un sistema vengono generati 6 processi (A,B,C,D,E), con i tempi di arrivo e le durate (in millisecondi) sotto specificate:

Processo	Durata	Tempo di arrivo
A	12	0
B	24	7
C	6	9
D	10	13
E	11	27
F	17	30

Tutti i processi avanzano senza mai sospendersi.
Calcolare il tempo di completamento di ogni processo (definito come differenza tra il tempo di completamento dell'esecuzione e il tempo di arrivo nel sistema) con la politica di scheduling SJF. Si ignori il tempo di commutazione di contesto.

Soluzione

PROCESSO	INIZIA ESECUZIONE	TERMINA ESECUZIONE	TEMPO DI COMPLETAMENTO	CODA DI PROCESSI PRONTI AL TERMINE DEL PROCESSO
A	0	12	12	C,B
C	12	18	18	D,B
D	18	28	28	E,B
E	28	39	39	F,B
F	39	56	56	B
B	56	80	80	-

ESERCIZIO A-4 (2 punti)

Nel sistema UNIX, il processo P genera un figlio eseguendo il seguente frammento di codice:

```
…
printf("compito ");
a = fork();
if (a>0)printf("di ");
else
   if (a==0) {
      execl("/pippo",NULL);
      printf("operativi ");
      }
printf("sistemi ");
```

Che cosa stampano per effetto di questo codice il processo padre e il processo figlio se il file `/pippo` non è un file eseguibile e la *fork* è eseguita con successo?

SOLUZIONE

Il processo P stampa: "compito", "di", "sistemi";
Il processo figlio stampa: "operativi", "sistemi".

ESERCIZIO A-5 (2 punti)

Si considerino due processi Unix A e B che comunicano tramite una pipe con un buffer di dimensione 3000 byte, e supponiamo che A abbia accesso alla pipe in scrittura mentre B abbia accesso in lettura.
Dire se il processo B è bloccato al termine delle seguenti sequenze di operazioni sulla pipe (da considerare in ALTERNATIVA):

1. Il buffer della pipe è vuoto e il processo B effettua una lettura dalla pipe.
2. Il buffer della pipe è vuoto; il processo B effettua una lettura della pipe; il processo A scrive un messaggio di dimensione 100 byte.

3. Il buffer della pipe è vuoto; il processo A scrive un messaggio di dimensione 1000 byte; il processo B legge dalla pipe.
4. Il buffer della pipe è pieno e il processo B effettua una lettura dalla pipe.

Soluzione

1. ……SI……
2. ……NO……
3. ……NO……
4. ……NO……

ESERCIZIO B-1 (4 PUNTI)

Un disco con 2 facce, 200 settori per traccia e 50 cilindri ha un tempo di seek (proporzionale al numero di cilindri attraversati) pari a 1 ms per ogni cilindro. Il periodo di rotazione è di 5 msec: di conseguenza il tempo impiegato per percorrere un settore è 0,025 msec.

Inoltre si fanno le seguenti ipotesi :

- il tempo necessario per raggiungere il settore desiderato dopo un'operazione di seek è sempre quello massimo, pari a 5 msec.
- il disco dispone di un buffer della capacità di un settore.

Al tempo 0 la testina è posizionata sul cilindro 44, settore 180 e sono pendenti le seguenti richieste di lettura o scrittura:

- cilindro 30 : settore 180 della faccia 0
- cilindro 20 : settore 130 della faccia 1
- cilindro 1 : settore 66 della faccia 0

Successivamente arrivano i seguenti comandi:

- al tempo 30: cilindro 1 settore 66 della faccia 0.
- al tempo 35: cilindro 41 settore 9 della faccia 0.
- al tempo 40: cilindro 12 settore 10 della faccia 1.

Lo scheduling è effettuato con politica SSF. Calcolare il tempo necessario per eseguire tutte le operazioni. Il tempo di esecuzione di ogni operazione è uguale alla somma dell'eventuale tempo di *seek,* del ritardo rotazionale (tempo necessario per raggiungere il settore indirizzato) e del tempo di percorrenza del settore indirizzato. Quando si raggiunge un cilindro, i comandi pendenti devono essere eseguiti nell'ordine in cui sono elencati.

Per il ritardo rotazionale dopo un'operazione di *seek* si assume sempre il valore di caso peggiore, pari a un periodo di rotazione.

Soluzione

op. su cilindro:	30	settore:	180						
inizio:	0	seek:	14	rotazione:	5	percorrenza:	0,025	fine:	19,025
op. su cilindro:	20	settore:	130						
inizio:	19,025	seek:	10	rotazione:	5	percorrenza:	0,025	fine:	34,05
op. su cilindro:	1	settore:	5						
inizio:	34,05	seek:	19	rotazione:	5	percorrenza:	0,025	fine:	58,075
op. su cilindro:	1	settore:	66						
inizio:	58,075	seek:	0	rotazione:	1,5	percorrenza:	0,025	fine:	59,6
op. su cilindro:	12	settore:	10						
inizio:	59,6	seek:	11	rotazione:	5	percorrenza:	0,025	fine:	75,625
op. su cilindro:	41	settore:	9						
inizio:	75,625	seek:	29	rotazione:	5	percorrenza:	0,025	fine:	109,65

SISTEMI OPERATIVI, CORSI A e B - SESTO APPELLO - 14/9/2006

COGNOME ..... NOME ..... MATRICOLA ..... CORSO |A| |B|

ESERCIZIO B-2 (4 punti)
In un sistema operativo che utilizza la rilocazione statica e gestisce la memoria con partizioni fisse, la memoria fisica ha un'ampiezza di 8 Mbyte ed è configurata con le seguenti partizioni:

- Partizione 1, ampiezza 1 Mbyte, riservata al Sistema Operativo;
- Partizione 2, ampiezza 2 Mbyte, disponibile per processi di ampiezza a con a≤2 Mbyte;
- Partizione 3, ampiezza 5 Mbyte, disponibile per processi di ampiezza a con 2<a≤5 Mbyte;

A partire dal tempo t=0 vengono generati nell'ordine i seguenti processi:

- t= 0: Processo P1, che occupa 1,4 Mbyte con tempo di completamento di 4 sec;
- t= 1: Processo P2, che occupa 0,6 Mbyte con tempo di completamento di 5 sec;
- t= 2: Processo P3, che occupa 4,7 Mbyte con tempo di completamento di 4 sec;
- t= 3: Processo P4, che occupa 3,8 Mbyte con tempo di completamento di 2 sec;
- t= 4: Processo P5, che occupa 0,5 Mbyte con tempo di completamento di 2 sec;
- t= 5: Processo P6, che occupa 2,2 Mbyte con tempo di completamento di 3 sec;

Ogni processo viene caricato nella minima partizione capace di contenerlo. Il caricamento dei processi nelle rispettive partizioni avviene con politica FIFO, e l'istante di caricamento coincide con la terminazione del processo in esecuzione. Una volta caricato in memoria, ogni processo mantiene l'assegnazione fino alla sua terminazione. Si trascura il tempo necessario per il caricamento in memoria dei processi.
I processi caricati in memoria (incluso l'eventuale processo appena caricato) vengono schedulati con politica Shortest Job First (SJF) e si suppone che i processi, una volta in esecuzione, mantengano lo stato di esecuzione fino alla terminazione, senza mai sospendersi.
Si chiede a quale tempo avvengono il caricamento in memoria e l'uscita dal sistema di ogni processo.

SOLUZIONE

- Processo P1 Caricato a t= 0 Nella partizione 2 Inizia esec. a t= 0 Esce dal sistema a t= 4
- Processo P3 Caricato a t= 2 Nella partizione 3 Inizia esec. a t= 4 Esce dal sistema a t= 8
- Processo P4 Caricato a t= 8 Nella partizione 3 Inizia esec. a t= 8 Esce dal sistema a t= 10
- Processo P6 Caricato a t= 10 Nella partizione 3 Inizia esec. a t= 10 Esce dal sistema a t= 13
- Processo P2 Caricato a t= 4 Nella partizione 2 Inizia esec. a t= 13 Esce dal sistema a t= 18
- Processo P5 Caricato a t= 18 Nella partizione 2 Inizia esec. a t= 18 Esce dal sistema a t= 20

ESERCIZIO B-3 (3 PUNTI)
Un disco ha due facce composte di 500 cilindri e 1000 settori per traccia. Tradurre nelle terne <cilindro, faccia, settore> i seguenti indirizzi di settore:

1. 800.000
2. 1.400
3. 78.912
4. 560.000
5. 909.111
6. 12.345

SOLUZIONE

Indirizzo di settore	cilindro	faccia	settore
800.000	400	0	0
1.400	0	1	400
78.912	39	0	912
560.000	280	0	0
909.111	454	1	111
12.345	6	0	345

SISTEMI OPERATIVI, CORSI A e B - SESTO APPELLO - 14/9/2006

COGNOME ..... NOME ..... MATRICOLA ..... CORSO |A| |B|

ESERCIZIO B-4 (2 punti)

Calcolare il numero di blocchi nel disco e la dimensione della FAT nei seguenti casi:

1. puntatori a 16 bit, blocchi di 16 KB, disco di 800 MB
2. puntatori di 24 bit, blocchi di 4 KB, disco di 1GB (1GB=1024 MB)
3. puntatori di 32 bit, blocchi di 16 KB, disco di 400 GB
4. puntatori di 24 bit, blocchi di 2 KB, disco di 400 MB

SOLUZIONE

1. numero di blocchi del disco: 50*2^10 — Dimensione della FAT 100 Kbyte
2. numero di blocchi del disco: 256 * 2^{10} — Dimensione della FAT 768 Kbyte
3. numero di blocchi del disco: 25 * 2^{20} — Dimensione della FAT 100 Mbyte
4. numero di blocchi del disco: 200 * 2^{10} — Dimensione della FAT 600 Kbyte

ESERCIZIO B-5 (2 punti)

Si consideri un sistema dove gli indirizzi logici hanno la lunghezza di 32 bit e le pagine logiche e fisiche hanno ampiezza di 2 kByte.Per la gestione della memoria con paginazione dinamica si utilizzano tabelle delle pagine a 2 livelli. La dimensione della tabella di primo livello è di 2^{10} elementi.
Negli elementi di ogni tabella di primo o secondo livello, che occupano 4 byte, 8 bit sono riservati agli indicatori (pagina caricata, riferita, modificata, ecc.) mentre i rimanenti rappresentano l'indice di un blocco fisico.
Si chiede:

1. la lunghezza del campo offset;
2. la lunghezza (numero di elementi) di ogni tabella di secondo livello;
3. lo spazio occupato in memoria dalla tabella di primo livello e da ogni tabella di secondo livello (numero di byte);
4. la massima dimensione della memoria fisica (numero di blocchi e di byte, espressi come potenze di 2).

SOLUZIONE

1. lunghezza del campo offset: 11 bit
2. lunghezza (numero di elementi) di ogni tabella di secondo livello: 2^{11}
3. spazio occupato in memoria dalla tabella di primo livello : $2^{10}*2^2 = 2^{12}$ (4 kByte)
4. spazio occupato in memoria da ogni tabella di secondo libello : $2^{11}*2^2 = 2^{13}$ (8 kByte)
5. massima dimensione della memoria fisica: blocchi 2^{24} ; byte. $2^{24}*2^{11} = 2^{35}$ (32 GByte)