Sistemi Operativi – I Prova in itinere

2 Maggio 2017

Esercizio 1 (8 punti)

Si consideri il seguente codice concorrente:

```
#include <unistd.h>
#include <stdio.h>
main()
{
    int i;
    for(i=0;i<3;i++)
        if(i%2==0)
            fork();
        else{
            fork();
            fork();
        }
}
```

Si disegni il relativo albero dei processi motivandone la costruzione.

Esercizio 2 (12 punti)

Si considerino i tre processi concorrenti:

```
semaphore S1 = 3;
semaphore S2 = 0;
```

```
[Processo 1]          [Processo 2]          [Processo 3]
L1:wait(S1)           L2:wait(S2)           L3:wait(S2)
   type("C")             type("A")             type("D")
   signal(S2)            type("B")             goto L3
   goto L1               signal(S2)
                         goto L2
```

che condividono i semafori **S1** e **S2**. L'esecuzione si considera terminata quando i tre processi sono tutti bloccati su una **wait()**.

1) Al termine dell'esecuzione dei tre processi, quante **C** risulteranno essere state stampate?
2) Al termine dell'esecuzione dei tre processi, quante **D** risulteranno essere state stampate?
3) Qual è il numero minimo di **A** ottenuto dall'esecuzione concorrente dei tre processi?
4) **CABABDDCABCABD** costituisce un possibile output frutto dell'esecuzione concorrente dei tre processi?
5) **CABACDBCABDD** costituisce un possibile output frutto dell'esecuzione concorrente dei tre processi?
6) È possibile che l'esecuzione concorrente dei tre processi termini con **S1** e/o **S2** diversi da 0?

Esercizio 3 (8 punti)

Un insieme di task indipendenti P_i, con i=1,…,7, deve essere eseguito su di un unico processore, gestito mediante un algoritmo a code multiple con priorità statiche. I tempi di arrivo, il tipo di processo e l'algoritmo di scheduling per ciascuna coda sono schematizzati nella seguente tabella:

	Processo	CPU–burst	Arrivo	Algoritmo
1° Foreground	P1	50	0.0	RR con quanto=15
	P2	15	30.0	
	P3	45	30.0	
2° Foreground	P4	40	0.0	SJF Preemptive
	P5	10	120.0	
Background	P6	30	60.0	FCFS
	P7	20	130.0	

Si tracci il diagramma di Gantt e si calcoli il tempo medio di attesa e il tempo medio di turnaround.

Esercizio 4 (7 punti)

Si supponga che in un sistema siano presenti i processi P_0, P_1, P_2, P_3, P_4, e un insieme di risorse di quattro tipi diversi, A, B, C, D, e di trovarsi nella seguente configurazione (Allocation, Max, Available):

	A	B	C	D		A	B	C	D		A	B	C	D
P_0	2+X	1	4	0		4	3	8	2		2	1	2	1
P_1	0	6	2+Y	4		4	8	6	6					
P_2	4	6	4	2+Z		5	7	8	4					
P_3	X	3	3	3		2	5	6	5					
P_4	1	2	1	2		3	3	2	3					

Determinare se esistono valori di X, Y e Z tali che:

a) il sistema sia in uno stato sicuro;
b) l'ulteriore richiesta di P_1 descritta da (1,2,2,0) possa essere soddisfatta.

In entrambi i casi, se possibile, indicare le schedulazioni dei cinque processi che garantiscono l'assenza di stallo. In caso contrario, mostrare quali risorse e processi potrebbero essere coinvolti nello stallo.