

Esercizi

1. Scrivere lo pseudocodice di una soluzione basata su semafori che coordini la seguente situazione:
 Un processo *server* S detiene **due istanze** di una risorsa W che è necessaria a due processi *client* C_1 e C_2 per effettuare la loro elaborazione. In particolare, C_1 necessita di entrambe le istanze di W , mentre C_2 necessita di una sola istanza. Le istanze di W sono condivise, e dopo essere state usate una volta (da C_1 o C_2) devono essere rimesse a disposizione del server, che deve avere il controllo dell'assegnazione della risorsa.
 Si supponga che C_1 , C_2 e S operino secondo il classico schema dell'elaborazione "infinita" (cioè `while(1) { ... }`).
[6 punti]

2. Si consideri il seguente insieme di processi:

<i>Processo</i>	<i>Burst</i>	<i>Tempo di Arrivo</i>
1	3	0
2	5	0.5
3	1.5	1
4	2	1.5

Si mostri l'esecuzione dei processi usando gli algoritmi di scheduling FCFS, SJF, HRRN, Round-Robin (quanto=1). Si calcoli il tempo di risposta e di turnaround per ogni processo, e i rispettivi tempi medi (su tutti i processi).
[4 punti]

3. Si descriva la struttura e il principio di funzionamento della tabella delle pagine invertita (*inverted page-table*), indicando quando il suo uso si rende necessario.
[4 punti]
4. Si fornisca un esempio di situazione (sequenza di accessi a disco) in cui gli algoritmi SCAN e C-SCAN fanno attraversare alla testina lo stesso numero di cilindri.
[5 punti]
5. Date 5 partizioni di memoria di dimensioni 100K, 500K, 200K, 300K, and 600K (nell'ordine), come vengono allocati processi di dimensioni 212K, 417K, 112K e 426K (nell'ordine) applicando gli algoritmi First-fit, Best-fit e Worst-fit? Quale algoritmo fa un uso più efficiente della memoria?
[5 punti]
6. Si consideri il seguente programma C:

```
#define N 300
int main() {
    int i;
    for (i = 0; i < N/2; i++)
        scambia(A[i], A[N-i-1]);
}
```

Il programma è *interamente* memorizzato negli indirizzi 0-99 di una memoria con 32 bit per parola. Il vettore `int A[N]`; è memorizzato a partire dalla locazione 100. Si assuma che il tipo `int` occupi una parola di memoria. Tale programma viene eseguito in un sistema con gestione della memoria con paginazione pura, avente dimensione di pagina pari a 400 byte (100 parole di memoria).

Sapendo che al processo siano allocati 2 frame, e che il primo frame contiene inizialmente il programma (e l'altro è vuoto), determinare:

- La reference string corrispondente all'esecuzione del programma;
- Il numero di page fault che si verificano durante l'esecuzione del programma usando l'algoritmo FIFO, LRU e quello ideale.

NOTA: l'operazione `scambia(A[i], A[i-1])` viene realizzata come:

- (i) `A[N-i-1] → registro1;`
- (i) `A[i] → registro2;`
- (iii) `registro1 → A[N-i-1]`
- (iv) `registro2 → A[i]`

[7 punti]