

Ricordarsi di mettere il proprio nome, cognome, e numero di matricola in tutti i fogli.

Motivare, brevemente ma chiaramente, tutte le risposte

Per avere la sufficienza, e' **necessario**** svolgere ognuno dei primi 3 esercizi.**

Esercizio 1 Considerate i 3 processi seguenti:

P1	P2	P3
print(A)	print(B)	print(C)
print (OK)	print (OK)	print (OK)

Si vuole aggiungere operazioni di semaforo in modo che la stringa finale stampata sia C B A OK OK OK, e che il valore di eventuali semafori alla fine della esecuzione del sistema siano gli stessi che i semafori avevano al momento della inizializzazione. Ricordarsi che, come sempre, e' preferibile massimizzare la concorrenza nel sistema.

Risposta(Sketch)

P1	P2	P3
T.P	S.P	
print(A)	print(B)	print(C)
U.V U.V	T.V U.P	S.V U.P
print (OK)	print (OK)	print (OK)

Tutti semafori a 0

Esercizio 2 Disegnare il grafo che rappresenta gli stati possibili di un processo, e le transizioni tra stati. Per ogni transizione, indicare almeno una ragione che puo' causare quella transizione.

Risposta(Sketch) Vedi appunti di corso

Esercizio 3 1. In un sistema con una una tabella di pagine (che non e' "paginata" su piu' livelli) come e' fatto un indirizzo logico e cosa indicano le sue componenti? E un indirizzo fisico?

Esercizio 4 Abbiamo 3 processi P1, P2,P3 che eseguono concorrentemente. Ogni Pi ripetutamente esegue una operazione op-i. Vogliamo che in un qualunque momento una qualunque di queste operazioni op-i sia stata eseguita al massimo una volta in piu' di una qualunque altra operazione op-j.

1. Proporre una soluzione al problema usando semafori, e spiegare perche' funziona. Si tratta di aggiungere operazioni su semafori all'interno del codice dei Pi, e non si possono aggiungere altri processi al sistema.
2. La stessa domanda di sopra, ma in questo caso c'e' il vincolo che all'interno di ogni Pi posso aggiungere al piu' 2 operazioni su semaforo. Posso pero' aggiungere altre componenti al sistema. (Qualora la soluzione al punto precedente sia quella proposta anche per questo punto: indicare la cosa chiaramente)

Come sempre, sono preferibili soluzioni che massimizzano la concorrenza.

Risposta(Sketch)

1. Uso 6 semafori binari, inizializzati ad 1 nello schema seguente:

P1

Loop

P(S12)

P(S13)

op1

V(S21)

V(S31)

P2

Loop

P(S21)

P(S23)

op2

V(S12)

V(S32)

P3

Loop

P(S31)

P(S32)

op3

V(S13)

V(S23)

2. Si aggiunge una 4a componente che effettua la sincronizzazione [Idea: uso un coordinator che aspetta un segnale da tutti i processi, e poi invia un segnale a tutti i processi quando possono ripartire] Tutti i semafori sotto sono inizializzati a 0

P1

Loop

P(S1)

op1

V(T1)

P2

Loop

P(S2)

op2

V(T2)

P3

Loop

P(S3)

op3

V(T3)

Coordinator

Loop

V(S1)

V(S2)

V(S3)

P(T1)

P(T2)

P(T3)

Esercizio 5 Un hard disk ha una dimensione di 2^{28} byte, suddivisi in blocchi da 2Kbyte. Il sistema operativo che usa l'hard disk adotta una allocazione indicizzata dello spazio su disco, e usa 4 byte per scrivere il numero di un blocco. Sul disco e' memorizzato un file A grande 350 Kbyte. Tutti gli attributi del file A sono gia' in RAM. (Tutte le risposte sotto date vanno adeguatamente motivate)

1. Quante operazioni di I/O su disco sono necessarie per leggere il byte numero 190.000 del file?
2. E' noto che, nel file system memorizzato sull'hard disk, la quasi totalita' dei file occupa un solo blocco di dati, e i file restanti occupano due blocchi di dati. In confronto alle altre tecniche di allocazione dello spazio su disco (contigua e concatenata), possiamo dire che viene sprecato tanto o poco spazio sull'hard disk, a causa dell'uso dell'allocazione indicizzata?
3. Sempre nelle ipotesi del punto (2), in confronto alle altre tecniche di allocazione dello spazio su disco (contigua e concatenata), possiamo dire che l'accesso ai dati dei file e' piu' lento o piu' veloce, a causa dell'uso dell'allocazione indicizzata?

Risposta(Sketch)

1. In un blocco indice possono essere contenuti $2048 / 4 = 512$ puntatori a blocco. L'uso di un solo blocco indice e' quindi sufficiente a memorizzare i numeri di tutti i blocchi in cui e' memorizzato A, per cui sono necessarie 2 operazioni di I/O: lettura del blocco indice, lettura del centesimo blocco del file (in cui e' contenuto il byte 204.800).
2. Viene sprecato tanto spazio, in quanto per ogni file viene usato anche un blocco indice, che e' quasi completamente inutilizzato.
3. Piu' lento. Infatti, per leggere i dati di un file occorre sempre prima leggere il suo blocco indice, e quindi sono comunque necessarie due letture su disco. Per le altre tecniche di allocazione invece, nella quasi totalita' dei casi basterebbe una lettura su disco, e negli altri casi ne sarebbero sufficienti due.

Esercizio 6 Dato un sistema con una RAM di 4 frame, la tabella sotto indica il numero di pagina memorizzata nel frame, il tempo di caricamento, il tempo dell'ultimo accesso alla pagina, e il reference bit.

Page	Load_Time	Last_Reference	Reference_Bit
0	22	32	0
1	34	36	1
2	10	33	1
3	23	38	1

1. Quale pagina sara' sostituita con l'algoritmo di page replacement FIFO? Spiegare brevemente anche il perche'.
2. Stessa domanda per la LRU.
3. Stessa domanda per l'algoritmo della "second chance" (assumendo che l'ordine con cui le pagine vengono esaminate e' dalla piu' vecchia, cioe' presente da piu' tempo, a quella piu' giovane).

Risposta(Sketch)

1. 2
2. 0
3. 0

Esercizio 7 Cosa e' il principio di localita' dei programmi? Date 2 esempi in cui durante il corso questo principio e' stato evocato per giustificare certe scelte in ambito di Sistemi Operativi o di hardware sottostante.

Esercizio 8 Cos'e' l'algoritmo della "Second Chance"? Brevemente, come funziona? E' un raffinamento, o approssimazione, di quale o quali algoritmi?

Risposta(Sketch) Vedi note corso.

Esercizio 9 Che cosa si intende per 'crittografia a chiave pubblica' e 'crittografia a chiave simmetrica' ?

In uno scambio di comunicazioni che richiede firma digitale, quale dei due schemi e' preferibile e perche'?

Esercizio sotto: Come alternativa all'esercizio sopra (con meno punti in palio)

Esercizio 10 1. Come funziona l'algoritmo di Shortest Job First?

2. Che svantaggi ha?
3. Come lo si puo' approssimare?

Risposta(Sketch) Vedi appunti di corso