

Ricordarsi di mettere il proprio nome, cognome, e numero di matricola in tutti i fogli. Motivare sempre le risposte date. Non e' necessario dare risposte molto lunghe, ma e' importante rispondere in modo motivato ed esauriente alle domande poste (in altre parole, molto meglio una frase in piu' che una in meno).

Per avere la sufficienza, e' **necessario**** svolgere tutti i primi 4 esercizi.**

Non sono ammesse macchinette calcolatrici o altre macchine elettroniche; non e' consentito uso di appunti o libri.

Malacopia: consegnare, se necessario **solo** gli esercizi che devono essere corretti (non riportati in bella copia); barrare quindi gli altri

Esercizio 1 1. Cosa e' il context switch? Che operazioni fa il sistema operativo durante il context switch?

2. Perche' e' importante che un sistema operativo minimizzi il tempo di context switch?

Risposta(Sketch) per (2): Il cambio del processo running e' tempo di overhead
Appunti corso per i dettagli

Esercizio 2 1. Cosa e' il principio di localita' dei programmi?

2. Indicate almeno un esempio di politica del sistema operativo che sfrutta questo principio, e un esempio di di parte hardware in un calcolatore che e' stato introdotto per sfruttare questo principio.

Risposta(Sketch) Vedi appunti corso. Per ultima domanda: memorie cache, TLB, alg. LRU.

Esercizio 3 Quali sono i vantaggi della paginazione, rispetto ad una allocazione della memoria centrale di tipo contiguo? Ci sono anche degli svantaggi?

Risposta(Sketch) Vedi appunti di corso

Esercizio 4 Considerate i 2 seguenti processi

P	Q
	print E
print B	print F
print C	print G

Questi 2 processi vengono lanciati in parallelo. Agite, se necessario, sul codice in modo che tutti e soli gli output di questa esecuzione concorrente siano le sequenze EBFCG e EFBCG

Risposta(Sketch)

```

P                               Q

                                print E
P(T)                            V(T)
print B                          print F
P(R)                            V(R); P(U)
print C                          print G
V(U)

```

con tutti i semafori inizializzati a 0.

Esercizio 5 Quattro processi arrivano al tempo indicato e consumano la quantità di CPU indicata nella seguente tabella:

Processo	T. di arrivo	Burst
P1	0	10
P2	1	8
P3	1	6
P4	11	3

1. Disegnare un diagramma di Gant che illustri la loro esecuzione nel caso dell'algoritmo di scheduling SJF preemptive.
2. Calcolare il turnaround medio e il waiting time medio per i processi, assumendo di essere nel caso del punto sopra.
3. Qual'è il più grave problema che può avere un algoritmo di scheduling a priorità, e come si risolve?
4. Per un sistema time sharing, meglio usare un algoritmo di scheduling preemptive o uno non preemptive? Quale algoritmo suggerireste come migliore?

Risposta(Sketch)

1. (0)P1 (1) P3 (7)P2 (11)P4(14)P2(18)P1(27)
2. Turnaround medio: $P1 = 27; P2 = 17; P3 = 6; P4 = 3$. Quindi $53/4 = 13,25$
Waiting time medio: $P1 = 17; P2 = 9; P3 = 0; P4 = 0$. Quindi $26/4 = 6,5$
3. La possibilità di starvation. Meccanismo di aging.
4. Per un sistema time sharing, è necessario usare un algoritmo preemptive (ma non SJF, ad esempio, RR, che non produce starvation) in modo da garantire che qualsiasi processo di qualsiasi utente possa prima o poi usare la CPU

Esercizio 6 Considerate i 2 seguenti threads

```
T_1 = while true do print A
```

```
T_2 = while true do print B
```

Questi 2 thread vengono lanciati in parallelo. Agite, se necessario, sul codice in modo che l'output di questa esecuzione concorrente sia sempre la sequenza

AAABAAABAAAB . . .

Le uniche modifiche possibili sono aggiunte di operazioni su semafori, che però devono essere inizializzati a 0.

E' possibile ottenere lo stesso effetto senza usare semafori ed usando soltanto una istruzione `sleep(n)` (dove `n` e' un intero e `sleep(n)` impone una attesa di `n` unita' di tempo al thread che la esegue) ?

Risposta(Sketch)

$$\begin{aligned} T_1 &= \text{while true do } T.P(); \text{ print } A; S.V() \\ T_2 &= \text{while true do } T.V(); T.V(); T.V(); S.P(); S.P(); S.P(); \text{ print } B; \end{aligned}$$

con entrambi i semafori inizializzati a 0.

Seconda domanda: no, non e' possibile poiche' la velocita' relativa e lo scheduling dei processi non sono predicibili

Esercizio 7 E' possibile che un processo P che gira su un sistema A dotato di memoria virtuale venga eseguito piu' velocemente che su un sistema B del tutto identico ad A ma non dotato di memoria virtuale?

Risposta(Sketch) Si, ad esempio se P non genera mai page fault e non e' necessario caricare in memoria tutto il codice di P per l'esecuzione.

Esercizio 8 In un SO, le pagine sono grandi 2^{17} byte. La macchina su cui viene installato il SO usa 31 bit per scrivere un indirizzo fisico, e lo spazio di indirizzamento fisico e' un quarto dello spazio di indirizzamento logico.

1. Qual'e' il numero max di frame?
2. Quanti bit sono necessari per scrivere un numero di frame?
3. Quanto puo' essere grande, al massimo, la page table di un processo? (assumete, come bit di controllo, il solo bit di validita')
4. Il sistema potrebbe dover usare una paginazione a piu' livelli?

Risposta(Sketch)

1. Lo spazio di indirizzamento logico e' $2^{31} \times 2^2 = 2^{33}$ B. Una page table puo' avere al massimo $2^{33}/2^{17} = 2^{16}$ entry, mentre il numero massimo di frame del sistema e' pari a $2^{31}/2^{17} = 2^{14}$.
2. Quindi, abbiamo bisogno di 14 bit per scrivere il numero di un frame.
3. Essendoci poi un solo bit di controllo, arriviamo a 15 bit, per i quali servono 2 B. Cosi' la dimensione massima della PT diventa $2 * 2^{16} = 2^{17}$ byte
4. Siccome 2^{17} non e' superiore alla taglia del frame, la PT puo' essere contenuta in un'unico frame. Non e' necessaria la paginazione su piu' livelli.

Esercizio 9 1. Nella cifratura a chiave simmetrica, perche' una permutazione sull'alfabeto non e' considerata una buona chiave?

2. Nella "cifatura a blocchi" perché è desiderabile che blocchi identici siano tradotti diversamente tra di loro, pur usando la stessa chiave? E come viene ottenuto questo?
3. Usare le proprietà dell'aritmetica modulare per calcolare in modo semplice $7^9 \bmod 9$
4. Quale teorema ci permette di calcolare immediatamente $7^6 \bmod 9$?

Risposta(Sketch) vedere appunti di corso

Per penultima domanda:

Sfruttiamo la proprietà che l'operazione di modulo distribuisce sulle operazioni come moltiplicazione ed esponenziazione (vedi appunti di corso). $7^8 \times 7 = (7^2)^4 \times 7$ Quindi possiamo calcolare $7^2 \bmod 9$ cioè $49 \bmod 9$, che è 4.

Ora possiamo calcolare $7^4 \bmod 9$ come $4^2 \bmod 9$, cioè 7.

e anche

$7^8 \bmod 9$ come $7^2 \bmod 9$, cioè 4.

Quindi $7^9 \bmod 9$ diventa $(4 \times 7) \bmod 9$ che da 1

Per ultima domanda: si tratta del teorema di Eulero: abbiamo $\phi(9) = 6$ e 7 è primo con 9.