

**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 ognuno dei 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** Che cos'e' la tabella delle pagine di un processo?

**Risposta**(Sketch)

vedi appunti corso

**Esercizio 2** 1. Cosa e' un algoritmo di page replacement?

2. Che cos'e' il fenomeno del trashing?

3. Come si puo' pensare che il SO eviti che si raggiunga una situazione di trashing?

**Risposta**(Sketch)

1. vedi appunti corso

2. Nei sistemi che implementano la memoria virtuale, e' un fenomeno per il quale i processi passano la maggior parte del loro tempo generando page fault e aspettando che la pagina mancante sia stata caricata in memoria primaria per poter ripartire. La CPU e' quindi scarsamente utilizzata.

3. Monitorando il numero di page fault per processo e quindi concedendo dei frame ad un processo o togliendoglieli quando tale numero supera o e' inferiore a certe soglie predeterminate.

In caso comunque si dovesse raggiungere una situazione problematica: occorre diminuire il livello di programmazione

**Esercizio 3** 1. Riportare il diagramma di stato della vita di un processo.

2. E' vero o falso che un processo puo' uscire volontariamente dallo stato di Ready ?

**Risposta**(Sketch)

1. Si veda il testo o i lucidi del corso

2. Falso. Un processo in stato di Ready non e' in esecuzione, e quindi puo' cambiare stato (passando allo stato Running) solo se selezionato dallo scheduler.

**Esercizio 4** Considerare questi 3 processi, lanciati in parallelo:

P1	P2	P3
<A>	<B>	<C>
<D>		

Volgiamo che: B sia eseguita dopo A e C; e che D sia eseguita per ultima. Non ci sono altri vincoli, in particolare nessun vincolo sull'ordine di esecuzione di A rispetto a C. Proponete una soluzione usando semafori. Indicare i valori di inizializzazione.

**Risposta(Sketch)**

P1	P2	P3
	S.P	
	T.P	
<A>	<B>	<C>
T.V	Z.V	S.V
Z.P		
<D>		

con tutti semafori a 0

**Esercizio 5** Si consideri un sistema in cui la tabella delle pagine di un processo puo' avere al massimo 256 entry. Un indirizzo fisico generato dal sistema e' scritto su 14 bit, e la RAM e' suddivisa in 32 frame.

1. Quanto e' grande lo spazio di indirizzamento logico del sistema?
2. Quale/quali dei seguenti numeri binari NON e' certamente un indirizzo logico corretto per il sistema sopra descritto?
  - (a) 1011
  - (b) 1100 0110 1011 0001 1000
  - (c) 1000 0000 0000 0000 0000
  - (d) 100 1011 1010 1011
3. E' vero che la tabella delle pagine di un processo di questo sistema potrebbe dover essere a sua volta paginata? Perche'?

**Risposta(Sketch)**

1. Indirizzi fisici su 14 bit, di cui 5 bit per numero frame ( $2^5 = 32$ ); ne consegue che offset prende 9 bit; quindi anche la parte offset ind. logico sara' di 9 bit.  
Allora spazio logico sara':  $2^8 \times 2^9 = 2^{17}$  B dove 8 ricavato da  $2^8 = 256$
2. (b) e (c), poiche' numeri che necessitano piu' di 18 bit.
3. No. Infatti, la tabella delle pagine di un processo ha al massimo 256 entry. Ogni entry deve contenere il numero di un frame, che e' scritto su 5 bit. Se anche si usa un byte per ogni entry, in tutto la tabella occupa 256 byte. Poiche' un frame e' grande 512 byte, la tabella in questione puo' essere contenuta in un unico frame, e non deve essere paginata a sua volta

**Esercizio 6** Un processo produce la seguente stringa di riferimenti:

1234125123

1. Riportate la sequenza di pagine memorizzate nei frame ad ogni riferimento, e il numero di page fault generati, nel caso di algoritmo LRU con 4 frame disponibili.

**Risposta(Sketch)**

1. (a) 4 frame con LRU: 6 page fault

```

1   2   3   4       1   2
1  12  123 1234   1234 1234

5   1   2   3
1254 1254 1254 1253

```

**Esercizio 7** Quattro processi arrivano al tempo indicato e consumano la quantità di CPU indicata nella tabella sottostante

Processo	T. di arrivo	Burst
P1	0	7
P2	2	6
P3	2	2
P4	7	1

1. Calcolare il waiting time medio per i processi nel caso dell'algoritmo di scheduling SJF preemptive, riportando anche il diagramma di Gantt corrispondente
2. L'algoritmo SJF preemptive garantisce che un processo che arriva in coda di ready riuscirà prima o poi ad usare la CPU?
3. Cosa è il turnaround di un processo?

**Risposta(Sketch)**

1. SJF preemptive:

P1..(2)..P3..(4)..P1..(7)..P4..(8)..P1..(10)..P2..(16)

waiting time medio: P1=3; P2=8; P3=0; P4=0. waiting time medio = 11/4

2. No, poiché potrebbe sempre arrivare in coda di Ready in processo con un burst time più piccolo di un processo già in coda che attende di essere selezionato dallo scheduler
3. turnaround: vedi appunti (tempo che intercorre tra arrivo processo nell'insieme processi ready e il completamento della sua esecuzione)

**Esercizio 8** Considerare questi 3 processi, lanciati in parallelo:

P1	P2	P3
loop forever	loop forever	loop forever
<A>	<B>	<C>

Volgiamo che la sequenza di operazioni sia ABBC ABBC .... . Proponete una soluzione usando semafori. Indicare i valori di inizializzazione.

**Risposta**(Sketch)

P1	P2	P3
loop forever	loop forever	loop forever
S.P	T.P	Z.P; Z.P
<A>	<B>	<C>
T.V	Z.V	S.V
T.V		

con solo  $S=1$ , gli altri a 0

**Esercizio 9** 1. Quale tra crittografia a chiave simmetrica e a chiave pubblica e' piu' sicura? Perche'?

2. Qual'e' piu' efficiente? Perche'?

3.

4. Usare le proprieta' dell'aritmetica modulo per calcolare in modo semplice  $16^8 \bmod 13$

**Risposta**(Sketch)

1. in entrambi i casi la sicurezza dipende dalla lunghezza delle chiavi

2. la simmetrica; quella pubblica prevede operazioni complicate di calcolo (esempio in RSA operazioni su numeri molto alti), mentre la simmetrica puo' prevedere diversi passi di esecuzione ma ognuno di essi e' molto veloce (es: permutazioni, xor)

3. Siccome  $16 \bmod 13 = 3$ , possiamo calcolare  $3^8 \bmod 13$

Ora:  $3^2 \bmod 13 = 9$ , inoltre  $9^2 \bmod 13 = 3$  (cioe'  $3^4 \bmod 13 = 3$ ).

Ne consegue che  $3^8 \bmod 13 = 9$ .