

**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.**

**Esercizio 1** Considerate i 3 processi sotto

P1	P2	P3
print(A)	print(B)	print(C)
print(D)	print(E)	

Agite, se necessario, sul codice, inserendo opportune operazioni su semaforo, in modo che le sole stringhe stampabili siano ABDEC oppure ABEDC.

**Risposta(Sketch)**

P1	P2	P3
print(A)	P(S) print(B)	P(U) P(U) print(C)
V(S);P(T)	V(T)	
print(D)	print(E)	
V(U)	V(U)	

**Esercizio 2** 1. Sia dato un sistema di memoria con indirizzi virtuali suddivisi in 2 campi,  $a$  e  $b$ , il primo utilizzato per indirizzare la tabella delle pagine e il secondo che rappresenta l'offset entro la pagina selezionata. Dall'ampiezza di quale o quali campi dipende il numero di pagine indirizzate nel sistema?

2. Stessa domanda di sopra, nel caso in cui gli indirizzi virtuali sono suddivisi in 4 campi,  $a$ ,  $b$ ,  $c$ ,  $d$ , i primi 3 dei quali sono utilizzati per indirizzare tre livelli gerarchici di tabelle delle pagine mentre il quarto campo rappresenta l'offset.

**Risposta(Sketch)** Nel primo caso,  $a$ , nel secondo caso  $a$ ,  $b$ ,  $c$ .

**Esercizio 3** 1. Cos'e' un semaforo binario?

2. quale di queste affermazioni e' vera per un semaforo binario:

- (a) gestisce solo l'accesso a due risorse condivise
- (b) gestisce solo le richieste di accesso provenienti da due processi

Motivare la risposta

**Risposta(Sketch)** vedi appunti lezione. Domanda 2: nessuna delle 2 (in generale gestisce una risorsa condivisa, con mutual exclusion, e i processi possono essere tanti, qualsiasi numero)

**Esercizio 4** Cinque processi, identificati dalle lettere A, B, C, D, E rispettivamente, arrivano agli istanti 0, 1, 2, 6, 7 rispettivamente. Tali processi hanno un tempo di esecuzione di 3, 7, 2, 3, 1 unità di tempo rispettivamente e priorità 3, 5, 2, 4, 1 rispettivamente (dove 5 è la massima priorità e 0 la minima). Per ognuna delle seguenti politiche di ordinamento:

1. Round Robin con quanto di tempo di ampiezza 2
2. Priority Scheduling (preemptive)

Illustrarne la esecuzione con un diagramma Gantt. Determinare, trascurando i ritardi dovuti allo scambio di contesto: il tempo medio di attesa. Nel caso di arrivi simultanei di processi allo stato di pronto, fatta salva (eventualmente) il rispettivo valore di priorità, si dia la precedenza ai processi usciti dallo stato di esecuzione rispetto a quelli appena arrivati. Nel caso di due processi aventi la stessa priorità, di cui uno in esecuzione, si dia la precedenza a quello in esecuzione.

**Risposta**(Sketch)

1. RR:

[A] 2 [B] 4 [A] 5 [C] 7 [B] 9 [D] 11 [E] 12 [B] 14 [D] 15 [B] 16

W.T: A=2, B=8, C=3, D=6, E=4, quindi WT medio = 23/5

2. Priority scheduling

[A] 1 [B] 8 [D] 11 [A] 13 [C] 15 [E] 16

W.T: A=10, B=0, C=11, D=2, E=8, quindi WT medio = 31/5

**Esercizio 5** [] Esistono i semafori in Java? Perché?

**Risposta**(Sketch) No, non esistono. Java adotta dei meccanismi di sincronizzazione più ad alto livello (synchronised methods) che facilitano la programmazione, per esempio limitano la possibilità di avere dei deadlock

**Esercizio 6** [] Considera i 3 processi sotto

<pre>P1 = repeat   print(A);   SC.V;   SA.P forever</pre>	<pre>P2 = repeat   print(B);   SC.V;   SB.P forever</pre>	<pre>P3 = repeat   SC.P   SC.P   print(C);   SA.V;   SB.V; forever</pre>
---	---	--

dove SA, SB, SC sono semafori. Supponiamo che i 3 processi sono fatti partire in parallelo, contemporaneamente, e con i 3 tre semafori inizializzati a 0. Quali stringhe possono essere stampate?

**Risposta**(Sketch) una stringa infinita, in cui le lettere nelle posizioni  $n*3+1$ ,  $n*3+2$ , e  $n*3+3$ , per  $n \geq 1$  possono essere ABC oppure BAC

**Esercizio 7** [] Si consideri un file system che permette aliasing (cioe' links). Supponiamo che abbiamo i file:

```
/home/john/file.txt  
/user/local/temp.txt
```

dove "temp.txt" e "file.txt" fanno riferimento agli stessi insiemi di blocchi di dati su disco (cioe': lo stesso file). Se "temp.txt" era un "hard-link", cosa succederebbe se il file "file.txt" fosse cancellato? E se "temp.txt" era un soft-link ?

**Risposta**(Sketch) Se temp.txt era un hard-link e si toglie file.txt, si decrementerebbe il reference count del file descriptor cui il file.txt fa riferimento. Si potrebbe ancora accedere a temp.txt normalmente. Se temp.txt fosse un soft-link, togliendo file.txt, si toglierebbe il descrittore e i data blocks cui fa riferimento. Non si farebbe nulla su temp.txt. Ma quando si cercasse poi di accedere a temp.txt, il sistema si lamenterebbe dando "invalid filename" o qualcosa del genere, indicando che fa riferimento ad un file che non esiste.

**Esercizio 8 (Virtual memory)** []

1. Nel caso di politica di Optimal Replacement, mostrare i contenuti dei frame della memoria dopo ogni referenza a pagina, data la reference string

1, 6, 1, 4, 1, 2, 5, 7, 3

e 4 frame a disposizione con contenuto iniziale: 4 8 3 5. Indicare chiaramente i page fault. Quanti ne occorrono?

2. In un algoritmo di "frame allocation", qual'e' la differenza tra local replacement e global replacement?

Si consideri la seguente reference string:

14, 18, 22, 23, 13, 21, 15, 17, 23, 28, 27, 21, 14, 13

dove la prima cifra in una referenza indica il numero di processo, la seconda cifra indica numero di pagina del processo.

Di nuovo, si assuma di avere 4 frame, con contenuto iniziale: 15 12 23 24.

Rispondere alla stessa domanda del punto 1 assumendo che ora pero' si usa un algoritmo di Optimal Replacement di tipo "local".

3. Per i punti 1 e 2 sopra, calcolare il tempo medio di accesso assumendo che: il tempo di swap e' 40 millisecondi; il tempo di accesso alla memoria e' 10 microsecondi; in media 100 accessi sono effettuati ad una pagina una volta che questa e' chiamata (cioe' ogni volta che compare nella reference string).

**Risposta**(Sketch) 1:

4 4 - 2 - -  
8 1 - - 7 -  
3 3 6 - - -  
5 5 - - - 3

(altre esecuzioni possibili)

2: local: sostituisco una pagina del processo in esecuzione global: sostituisco una pagina di un processo qualunque

15  
12  
23  
24

situazione finale (unica possibile):

14  
13  
27  
21

3: per 1:

$$( (5 \cdot 40 \cdot 1000) + 9 \cdot 10 \cdot 100 ) / 9 \cdot 100$$

(dove 5 e' dato dal numero di page fault, e 9 sono le pagine utilizzate). Simile il calcolo al punto 2

**Esercizio 9** Quelli sotto sono alcuni tra gli algoritmi per page-replacement visti a lezione. Indicare per almeno 2 di essi il problema principale che comporta.

- First-In First-Out
- Optimal Algorithm
- LRU (Least Recently Used)

**Risposta**(Sketch)

1. Si puo' togliere una pagina ancora fortemente usata
2. assolutamente troppo costoso da implementare
3. necessaria assistenza hardware; resta comunque costoso (c'e' qualcosa da fare per ogni accesso alla memoria); in genere si implementano delle approssimazioni dell'algoritmo

**Esercizio 10** 1. Riportare la distinzione tra crittografia a chiave simmetrica e a chiave pubblica. Quali sono i principali vantaggi della seconda?

2. Descrivere i passi che portano alla determinazione della chiave pubblica e privata nell'algoritmo di RSA.

**Risposta**(Sketch)

1. Vedi appunti. Vantaggi: potere gestire le firme digitali, e casi simili. Piu' facile distribuire chiavi

2. vedi appunti

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

**Esercizio 11** Considerate il protocollo di mutual exclusion per 2 processi sotto:

```
// Process 1                                // Process 2
<init1>;                                    <init2>;
while(true) {                                while(true) {
    turn = 2;                                  turn = 1;
    c1 = true;                                  c2 = true;
    while (c2 && turn == 2) { };                while (c1 && turn == 1) { };
    <crit1>;                                    <crit2>;
    c1 = false;                                  c2 = false;
    <rem1>;                                    <rem2>;
}                                                }

// Shared variables
boolean c1 = false, c2 = false;
integer turn = 1;
```

Questo protocollo soddisfa le proprietà di mutual exclusion e assenza di deadlock? Spiegate intuitivamente perché.

**Risposta**(Sketch) Sì, vedi appunti lezione.