

**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** Descrivere i principali vantaggi e svantaggi della memoria virtuale.

**Esercizio 2** [] Considerate i 2 processi sotto

```
var x = 0
```

proc. P1	proc. P2
print(x)	x := x+1
print(x)	print(x)
x:= x-1	

Agite, se necessario, sul codice, inserendo opportune operazioni su semaforo, in modo che vengano stampati i valori 0,1,0 (nell'ordine) Indicare bene i semafori utilizzati e il loro valore di inizializzazione.

**Risposta(Sketch)**

```
var x = 0
```

proc. P1	proc. P2
	P(S)
print(x)	x := x+1
V(S);P(T)	V(T); P(U)
print(x)	print(x)
x:= x-1	
V(U)	

con inizialmente tutti i semafori a 0

**Esercizio 3** Come funziona l'algoritmo di LRU?

**Esercizio 4** Indicate almeno 4 motivi (se sono piu' di 4 ancora meglio) per cui un processo che sta eseguendo sul processore, quindi che si trova in stato running, potrebbe passare in uno stato diverso. Distinguate i casi in cui il cambio di stato avviene per cause che dipendono dal processo stesso (quindi un cambio volontario) e cause esterne (cambio involontario)

**Risposta(Sketch)**

volontariamente	involontariamente
termina	quanto di tempo scaduto
operazione di I/O	page fault
lancio di una sottoattivit�	
con maggiore priorit�	
operazione P	
con semaforo < 1	

**Esercizio 5** 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 6** 1. Come funziona l'algoritmo della seconda chance?

2. Perche' l'algoritmo della seconda chance viene considerato una approssimazione di quello LRU?

**Risposta**(Sketch) vedi appunti di corso.

Sull'ultima: Perche' usando il reference bit riesce a discriminare tra pagine riferite di recente e non riferite di recente

**Esercizio 7** Si consideri il problema dei lettori e scrittori visto a lezione, dove i codici del generico scrittore e del generico lettore sono riportati qui di seguito.

1. Inserite le operazioni di decremento/incremento (dette anche wait e signal, oppure P e V) su semaforo mancanti necessarie per il corretto funzionamento del sistema, indicando anche il semaforo mancante. Ricordarsi dei valori di inizializzazione. Come per tutte le risposte, motivate brevemente anche le aggiunte fatte.
2. Si consideri l'algoritmo del punto precedente. Supponiamo che, in un dato periodo, un processo scrittore stia aggiornando il file, e piu' processi lettori tentano di accedere in lettura al file. Dove si addormentano i processi lettori (in quale punto del codice)?

semafori e variabili condivise necessarie

```
semaphore write = 1;
int numlettori = 0;
```

```
scrittore {
```

```
...Esegui la scrittura del file ...
```

```
}
```

```
lettore {
```

```
numlettori++;
if numlettori == 1 ;
... leggi il file ...
```

```
numlettori--;  
if numlettori == 0 ;
```

### Risposta(Sketch)

semafori e variabili condivise necessarie

```
semaphore mutex = 1;  
semaphore write = 1;  
int numlettori = 0;  
  
scrittore {  
wait(scrivi);  
...Esegui la scrittura del file ...  
signal(scrivi) }  
  
lettore {  
wait(mutex);  
numlettori++;  
if numlettori == 1 wait(scrivi);  
signal(mutex);  
... leggi il file ...  
wait(mutex);  
numlettori--;  
if numlettori == 0 signal(scrivi);  
signal(mutex);
```

Il primo lettore si addormenta sul semaforo scrivi, gli altri si addormentano sul semaforo mutex

**Esercizio 8** Tre processi P1 P2 e P3 arrivano uno dopo l'altro nell'ordine indicato ma in tempi diversi in coda di ready, ed hanno anche diversi periodi di CPU burst (cioe' di uso della CPU). Indicare una condizione che garantisca che il turnaround medio del sistema relativo ai tre processi risulta lo stesso sia adottando un algoritmo di scheduling FCFS che adottando un algoritmo di scheduling SJF preemptive.

**Risposta(Sketch)** E' sufficiente che:  $Burst\ P1 < Burst\ P2 < Burst\ P3$

**Esercizio 9** In un sistema che adotta la memoria paginata le pagine hanno una dimensione tale da produrre una frammentazione interna media per processo di 1 Kbyte (cioe'  $2^{10}$  byte). La dimensione della tabella delle pagine piu' grossa del sistema è di X byte, e si sa che se tale dimensione fosse anche solo X+1 byte, il sistema dovrebbe adottare una paginazione a piu' livelli. Ogni entry di una tabella delle pagine occupa 2 byte, e tutti i bit vengono usati per scrivere il numero di un frame.

1. Quanto e' grande una pagina?
2. Quante entry ci sono al piu' nella tabella di pagine?
3. Quanto sono grossi lo spazio di indirizzamento logico e fisico del sistema?

**Risposta(Sketch)**

1. Il sistema adotta pagine da 2 Kbyte, ossia  $2^{11}$  byte
2. La tabella delle pagine piu' grossa del sistema occupa un intero frame, e quindi ha in tutto  $2^{11}/2 = 2^{10}$  entry
3. Lo spazio logico e' quindi grande  $2^{10} * 2^{11} = 2^{21}$  byte (2 megabyte) e quello fisico e' grande  $2^{16} * 2^{11} = 2^{27}$  byte (128 megabyte)