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 1. In un sistema con paginazione e memoria virtuale, cosa e' la tabella delle pagine, e a cosa serve?

2. Cosa vuole che la tabella delle pagine e' paginata su piu' livelli?

Risposta(Sketch) vedi note corso

Esercizio 2 Cosa e' il Process Control Block, e a cosa serve?

Risposta(Sketch) vedi note corso

Esercizio 3 1. Cosa e' il principio di localita' dei programmi?

2. Indicate almeno 2 esempi di politiche del sistema operativo che sfruttano questo principio.

3. Indicate almeno 1 esempio di parte hardware in un calcolatore che esiste grazie a questo principio.

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

Esercizio 4 [5] Considerate i 2 seguenti threads

```
T_1 =   print M print R

T_2 =   print A print E
```

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

Le uniche modifiche possibili sono aggiunta di operazioni su semafori. Specificate anche il valore iniziale dei semafori.

Esercizio 5 Quattro processi arrivano al tempo indicato e consumano la quantita' 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 preemtpive.

2. Calcolare il turnaround medio e il waiting time medio per i processi, assumendo di essere nel caso del punto sopra.

3. Qual'e' il piu' grave problema che puo' avere un algoritmo di scheduling a priorita', e come si risolve?

4. Per un sistema time sharing, meglio usare un algoritmo di scheduling preemptive o uno non preemptive? Quale algoritmo suggerite 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

3. La possibilita' di starvation. Con un meccanismo di aging.

4. Per un sistema time sharing, e' 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 Supponete che un processo esegua una operazione di decremento (detta anche wait) su un semaforo. Puo' modificarsi lo stato del processo a seguito di questa operazione? Come?

Risposta(Sketch) Se la risorsa gestita dal semaforo e' "occupata", ci sara' una transizione in stato "waiting". Altrimenti il processso, o resta running, oppure puo' passare ready (esempio per termine quanto tempo)

Esercizio 7 Si consideri il problema dei lettori e scrittori, dove i codici del generico scrittore e del generico lettore sono riportati qui di seguito. Inserite le operazioni di decremento/incremento (dette anche wait e signal, P e V) su semaforo mancanti necessarie per il corretto funzionamento del sistema, indicando anche il semaforo mancante ed il suo valore di inizializzazione. Motivate brevemente anche le aggiunte fatte.

```
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);
```

Esercizio 8 1. Un sistema operativo e' in grado di decidere, scegliendo tra le tre modalita' di base di allocazione dello spazio su disco, quella piu' adeguata per memorizzare un file in base alle seguenti informazioni, note al S.O. stesso: (i) numero di blocchi occupati dal file, (ii) tipo di accesso al file (sequenziale o diretto, che viene dichiarato dall'utente al momento della creazione del file stesso).

   Per ciascuno dei file riportati qui di seguito, indicate quale modalita' di allocazione scegliera' il S.O.:

   File A: 1 blocco, sequenziale

   File B: 100 blocchi, diretto

   File C: 1 blocco, diretto

   File D: 100 blocchi, sequenziale

2. Nel sistema descritto nel punto sopra, i blocchi su disco occupano 512 byte, e un puntatore a blocco e' scritto su 4 byte. Di un file si sa che deve essere acceduto in modo diretto. Quanti accessi al disco sono necessari per leggere direttamente il contenuto del blocco numero 200 del file, assumendo che gli attributi del file in questione siano gia' in memoria primaria?

3. Nel sistema del punto (1) si viene a sapere che tutti i file devono poter essere acceduti in modo diretto, e occupano sempre piu' di un blocco. Quale modalita'ă verrebbe adottata per memorizzare il file? Quale svantaggio avrebbe comunque questa modalita', nel caso di file che occupano comunque pochi blocchi?

Risposta(Sketch)

1. **FileA:** contigua (o concatenata, l'indicizzata spreca spazio inutilmente)

FileB: indicizzata

FileC: contigua (o concatenata, l'indicizzata spreca spazio inutilmente)

FileD: 100 blocchi, sequenziale:

concatenata (e' ragionevole anche l'indicizzata, sebbene produca un maggiore spreco di spazio, perche' piu' affidabile della concatenata)

2. Il S.O. usera' l'allocazione indicizzata. In un blocco indice possiamo scrivere 128 puntatori, per cui un solo blocco indice non e' sufficiente ad indirizzare il blocco 200. Sia usando l'allocazione indicizzata gerarchica che l'allocazione indicizzata concatenata, un secondo blocco indice e' sufficiente per indirizzare il blocco 200. Sono quindi necessari due accessi al disco per leggere i due blocchi indice + un accesso per leggere il blocco 200.

3. L'allocazione indicizzata. Per file piccoli, quasi tutto il blocco indice viene sprecato.

Esercizio 9 1. Quali sono i principali vantaggi e svantaggi della crittografia a chiave simmetrica e rispetto a quella a chiave pubblica?

2. Usare le proprieta' dell'aritmetica modulo per calcolare in modo semplice $15^6 \bmod 11$

Risposta(Sketch) Per la prima domanda, vedi appunti corso

Sfruttiamo la proprieta' che l'operazione di modulo distribuisce sulle operazioni come moltiplicazione ed esponenziazione (vedi appunti di corso).

$$\begin{aligned} & 15^6 \bmod 11 \\ = \; & ((15^4 \bmod 11) \times (15^2 \bmod 11)) \bmod 11 \end{aligned}$$

Calcolo quindi $(15^2 \bmod 11)$:

$$\begin{aligned} & 15^2 \bmod 11 \\ = \; & ((15 \bmod 11) \times (15 \bmod 11)) \bmod 11 \\ = \; & (4 \times 4) \bmod 11 \\ = \; & 16 \bmod 11 \\ = \; & 5 \end{aligned}$$

Calcolo ora $(15^4 \bmod 11)$:

$$\begin{aligned} & 15^4 \bmod 11 \\ = \; & ((15^2 \bmod 11) \times (15^2 \bmod 11)) \bmod 11 \\ = \; & (5 \times 5) \bmod 11 \\ = \; & 25 \bmod 11 \\ = \; & 3 \end{aligned}$$

Posso ora chiudere il calcolo iniziale

$$\begin{aligned} & ((15^4 \bmod 11) \times (15^2 \bmod 11)) \bmod 11 \\ = \; & (3 \times 5) \bmod 11 \\ = \; & 15 \bmod 11 \\ = \; & 4 \end{aligned}$$