

Prendete il vostro numero di matricola. Eliminate tutti gli “0”, e chiamate  $N$  il numero risultante. Esempio se il numero di matricola e’ 0000800470, allora  $N = 847$

Nel seguito  $N_i$  e’ la cifra numero  $i$  di  $N$ , contando a partire da 0, modulo la lunghezza di  $N$ . Nell’esempio sopra:

- $N_0 = 8$
- $N_1 = 4$
- $N_2 = 7$
- $N_3 = 8$
- $N_4 = 4$

e cosi’ di seguito. Quindi per esempio  $N_2N_3$  e’ il numero 78 (e NON la moltiplicazione  $7 \times 8$ ).

Negli esercizi sotto: non eseguite calcoli o ragionamenti astratti in termini di questi  $N_i$ , ma sostituite subito ai vari  $N_i$  il valore concreto dato dal vostro numero matricola.

---

**Ad inizio foglio indicate chiaramente il vostro numeri di matricola e quasi sono nel vostro caso i numeri  $N_0, N_1, N_2, N_3$**  Motivare 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). Ricordare che lo scan finale deve contenere il libretto universitario di riconoscimento

---

## **E’ necessario svolgere gli esercizi 1 e 2**

---

**Esercizio 1** Come in esercizi a lezione, date 2 lettere H,K, usiamo la notazione  $\{HK\}$  per indicare le due sequenze possibili HK e KH. Quindi per esempio “stampare  $L\{HK\}M$ ” significa essere in grado di stampare le 2 stringhe LHKM e LKHM (e nessun altra). Considerate i seguenti processi:

P1

```
print(A)
print(B)
print(C)
```

P2

```
print(D)
print(E)
print(F)
```

Se  $N_2$  e  $N_3$  sono entrambe pari, seguite la opzione (a) sotto; se sono entrambe dispari, seguite la opzione (b) sotto; se  $N_2$  e’ pari e  $N_3$  e’ dispari, seguite la opzione (c) sotto; se  $N_2$  e’ dispari e  $N_3$  e’ pari, seguite la opzione (d) sotto.

Agite, se necessario, sul codice, inserendo opportune operazioni su semaforo, in modo che l’output di questa esecuzione concorrente siano esattamente le stringhe

- (a)  $\{AD\}BECF$
- (b)  $A\{BD\}ECF$
- (c)  $AD\{BE\}CF$
- (d)  $ADE\{BF\}C$

**Risposta**(Sketch) Nelle soluzione sotto, tutti i semafori inizializzati a 0

(a) {AD}BECF

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

(b) A{BD}ECF

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

(c) AD{BE}CF

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

(d) ADE{BF}C

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

**Esercizio 2** In un sistema operativo che adotta uno scheduling di tipo Round Robin con quanto di tempo 3, quattro processi arrivano al tempo indicato e consumano la quantita' di CPU indicata nella tabella sottostante:

Processo	T. di arrivo	Burst time
A	0	$N_2 + 5$
B	4	$N_3 + 3$
C	7	$N_1$
D	10	$N_2$

1. Riportate il diagramma di Gantt che mostra cio' che succedera' in esecuzione.
2. Qual e' il waiting time di B?

**Risposta**(Sketch) Svolgimento, assumendo  $N_2 = 7, N_3 = 6, N_1 = 8$  Quindi abbiamo

Processo	T. di arrivo	Burst time
A	0	12
B	4	9
C	7	8
D	10	7

Riporto Gantt ed evoluzione pila:

[0] A [3] A [6] B [9] A [12] C [15] B [18] D [21] A [24]  
C [27] B [30] D [33] C [35] D [36]

A  
A  
BA  
ACB  
CBDA  
BDAC  
DACB  
ACBD  
CBD  
BDC  
DC  
CD  
D

waiting time B:  $30 - 4 - 9 = 19$

**Esercizio 3** Se  $N_2 \leq N_3$  andate al punto (a); se  $N_2 > N_3$  andate al punto (b).

- (a) In un SO la tabella delle pagine puo' contenere al massimo  $2^{N_1 N_2}$  entry. Nell'indirizzo fisico, la parte offset e' fatta da  $N_1 N_3$  bit.
- (b) In un SO la tabella delle pagine puo' contenere al massimo  $2^{N_1 N_3}$  entry. Nell'indirizzo fisico, la parte offset e' fatta da  $N_1 N_2$  bit.

In entrambi i casi (a) e (b) le domande a cui rispondere sono le seguenti:

1. Quando e' grande un frame, in byte?
2. Quanti sono i bit dell'indirizzo logico?
3. Quanti frame deve avere questo sistema per poter dire con certezza che il sistema deve implementare la memoria virtuale?

4. Quanti byte dovrebbero contenere ciascuna entry della tabella delle pagine per dover adottare un sistema di paginazione a piu' livelli (cioe' con la tabella delle pagine paginata su piu' livelli)?

**Risposta**(Sketch) Caso (a)

1.  $2^{N_1 N_3}$
2.  $N_1 N_2 + N_1 N_3$
3. Meno di  $2^{N_1 N_2}$  frame
4. Dovrebbe valere  $2^{N_1 N_2} \times x > 2^{N_1 N_3}$ , quindi piu' di  $2^{N_1 N_3 - N_1 N_2}$

Caso (b)

1.  $2^{N_1 N_2}$
2.  $N_1 N_3 + N_1 N_2$
3. Meno di  $2^{N_1 N_3}$  frame
4. Dovrebbe valere  $2^{N_1 N_3} \times x > 2^{N_1 N_2}$ , quindi piu' di  $2^{N_1 N_2 - N_1 N_3}$

**Esercizio 4** Sia  $M$  il minimo tra i numeri 5 e  $N_2$ .

- (a) Usare le proprieta' dell'aritmetica modulo per calcolare in modo semplice

$$(N_2 \times 7 + M)^{N_3+5} \bmod 7$$

(quindi la base dell'esponenziazione e' il numero  $(N_2 \times 7 + M)$ , l'esponente e' il numero  $N_3 + 5$ ).

- (b) Riportare la distinzione tra crittografia a chiave simmetrica e a chiave pubblica. Quali sono i principali vantaggi della seconda?

**Risposta**(Sketch) Svolgimento, assumendo  $N_2 = 5, N_3 = 6$

Intanto, la domanda e' la stessa di chiedere  $N^{N_3+5} \bmod 7$ . Assumo  $M = 5$ . Quindi interessa  $5^{11} \bmod 7$ .

Abbiamo:  $5^2 \bmod 7 = 4$ . Quindi  $5^4 \bmod 7 = 4^2 \bmod 7 = 2$ .

Quindi  $5^8 \bmod 7 = 2^2 \bmod 7 = 4$ .

Siccome  $11 = 8 + 2 + 1$ , diventa  $(4 + 4 + 5) \bmod 7 = 6$ .

Per la parte (b), vedere note corso

**Esercizio 5** Riportate lo pseudocodice che descrive l'implementazione dell'operazione di P(S) su un semaforo. Che problemi di atomicita' (cioe' di interferenze possibili) presenta? (A lezione si sono viste 2 possibili implementazioni di semaforo, sto facendo riferimento alla prima, dove il semaforo e' anche chiamato "spinlock")

**Risposta**(Sketch) vedere note corso.

**Esercizio 6** 1. Qual'e' lo svantaggio principale del metodo di "File allocation" chiamato "Linked list" ?

2. Quali sono, sempre in questo metodo, i possibili pro e contro che emergono al variare delle dimensioni dei blocchi?

**Risposta**(Sketch)

1. Accesso diretto ai record logici inefficiente. Nel caso peggiore, un file system deve accedere tutti i blocchi di un file (caso di singly-linked list) per trovare i dati richiesti, o meta' (caso di doubly-linked list).
2. Un blocco grande riduce il numero di operazioni IO richieste per ritrovare un certo record, ma al costo di maggiore frammentazione interna; riducendo la dimensione e' il contrario

**Esercizio 7** 1. In una politica di CPU scheduling a priorit , qual'  il problema principale?

2. Perch  nei sistemi operativi moderni la politica Round Robin   quella pi  diffusa per gestire processi utente?

**Risposta**(Sketch) vedere note corso.

**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** 1. **Riportate il diagramma di stato della vita di un processo.**

2. **In quali transizioni da uno stato all'altro deve intervenire il sistema operativo?**

**Risposta(Sketch)** In tutte, in quanto e' il SO che sposta i processi (i loro PCB) da uno stato all'altro.

**Esercizio 2** **Quali sono i principali vantaggi e svantaggi della paginazione della memoria?**

**Esercizio 3** **In un file system, come funziona l'allocazione di file di tipo "concatenato"? Quali sono gli inconvenienti principali?**

**Esiste un problema di frammentazione interna o esterna per questo tipo di allocazione?**

**Esercizio 4 []**

1. Considerate i 2 processi sotto

P1 =	P2 =
print(A);	print(B)
print(D)	print(C)
	print(E)

Agite, se necessario, sul codice, inserendo opportune operazioni su semaforo, in modo che nell'output di questa esecuzione concorrente sia la stringa ABCDE.

2. Assumete ora che le stampe sopra siano include dentro un loop:

P1 = loop( print(A); print(D) )	P2 =loop( print(B) print(C) print(E) )
---------------------------------------	---

Occorre apportare dei cambiamenti alla soluzione sopra in modo da avere come stringa stampata la stringa infinita ABCDEABCDEABCDE... ? Se si, indicare i cambiamenti.

**Risposta(Sketch)**

P1 =  print(A); V(S); P(T) print(D) V(R)	P2 =  P(S) print(B)  print(C) V(T); P(R) print(E)
---	--

con tutti i semafori a 0

Stringa infinita: si aggiunge un semaforo  $U$  inizializzato a 0, mettendo

P(U)

come ultima istruzione del loop sul primo processo e

V(U)

come ultima istruzione del loop sul secondo.

**Esercizio 5** Si consideri uno scheduler che riceve 4 job A,B,C,D con le seguenti caratteristiche:

Job	durata	tempo di inizio
A	8	0
B	3	2
C	1	4
D	2	7

Riportare il diagramma Gantt relativo ad una esecuzione con round robin e con quanto di tempo pari a 3. Qual'è il waiting time medio?

**Risposta**(Sketch)

0 [A] 3 [B] 6 [A] 9 [C] 10 [D] 12 [A] 14

Quindi waiting time A :  $14 - 8 = 6$

Waiting time B :  $6 - 3 - 2 = 1$

Waiting time C :  $10 - 1 - 4 = 5$

Waiting time D :  $12 - 7 - 2 = 3$

Waiting time medio:  $6 + 1 + 5 + 3 / 4 = 15/4$

**Esercizio 6** 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. Inserite le operazioni di decremento/incremento 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.

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; solo scrittore e lettore
int numlettori = 0;

```

```

scrittore {
P(scrivi);
...Esegui la scrittura del file ...
V(scrivi) }

```

**scrittore → prende un semaforo  
entra nel semaforo  
esegue scrittura dei file  
esce dal semaforo**

```

lettore {
P(mutex);
numlettori++;
if numlettori == 1 P(scrivi);
V(mutex);
... leggi il file ...
P(mutex);
numlettori--;
if numlettori == 0 V(scrivi);
V(mutex);

```

**lettore → prende il semaforo di mutex  
mutex → incrementa un qualcosa e se lui e' il primo lettore  
allora blocca lo scrittore, P e V di mutex sono importanti  
perchè altrimenti i lettori partirebbero simultaneamente  
eaccadrebbe:  
1. aumento dei lettori a 2 (aumento lettori)**

**Esercizio 7** Cosa e' la starvation? La soluzione del problema dei lettori e scrittori sopra garantisce l'assenza di starvation?

**Risposta(Sketch)** Starvation: vedi note corso.

No, infatti un qualsiasi processo scrittore potrebbe dover attendere all'infinito senza riuscire a ad entrare in sezione critica. Al contrario i processi lettori sono liberi da starvation.

**Esercizio 8** In un sistema con paginazione della memoria centrale, un indirizzo fisico e' scritto su 37 bit, l'offset piu' grande all'interno di una pagina e' pari a 11 1111 1111, e la tabella delle pagine piu' grande del sistema occupa 32 megabyte.

1. Quanti frame ci sono nel sistema?
2. Quante entry ci sono nella tabella di pagine? (potete assumere che ci sia solo il bit di validita' e nessun altro bit di controllo)



3. Quanto e' grande lo spazio di indirizzamento logico del sistema?
4. E' necessaria la paginazione a piu' livelli della tabella di pagine? Se si, su quanti livelli?
5. E' necessaria la memoria virtuale?

**Risposta(Sketch)**

1. Nell'indirizzo fisico,  $37-10=27$  bit sono usati per numero di frame. Quindi ci sono  $2^{27}$  frame
2. Una entry prende 4 B (che servono per i 27 bit del num di frame e poi il resto dei bit e' inutilizzato o usato per bit validita') Quindi ci sono  $2^{25}/2^2 = 2^{23}$  entry
3. Spazio di indirizzamento logico:  $2^{23} \times 2^{10} = 2^{33}$
4. si (tab pagine piu' grande del singolo frame) Siccome  $2^{25}/2^{10} = 2^{15}$ , la TP di secondo livello avra'  $2^{15}$  entry, e siccome una entry prende 4B, abbiamo una taglia di  $2^{17}$  B per tale TP di secondo livello, che a sua volta non e' quindi contenibile in un frame. Avremo quindi bisogno di un terzo di livello di TP (al terzo livello servono  $2^7$  entry, quindi  $2^9$  B e questi stanno dentro un singolo frame)
5. si: Lo spazio logico non e' piu' grande dello spazio fisico ( $2^{33}$  contro  $2^{37}$ ); quindi strettamente parlando non e' necessaria la memoria virtuale. Ma comunque le grandezze non sono molto diverse (tra l'altro c'e' anche da tenere presente che parte dello spazio fisico dovra' essere dedicato al SO), e quindi una memoria virtuale e' altamente consigliabile per il discorso multiprogrammazione.

**Esercizio 9**

1. Nella cifratura a chiave simmetrica, perche' una permutazione sull'alfabeto non e' considerata una buona chiave?
2. Usare le proprieta' dell'aritmetica modulo per calcolare in modo semplice  $(5^2 \times 8^3) \bmod 7$ , e anche  $11^6 \bmod 7$

**Risposta(Sketch)**

1. metodi statistici consentono di risalire alla chiave facilmente, per messaggi abbastanza lunghi.
- 2.

$$\begin{aligned}
 & (5^2 \times 8^3) \bmod 7 \\
 = & ((5^2) \bmod 7 \times (8^3) \bmod 7) \bmod 7 \\
 = & ((5^2) \bmod 7 \times (8^3) \bmod 7) \bmod 7 \\
 = & (25 \bmod 7 \times ((8 \bmod 7)^3) \bmod 7) \bmod 7 \\
 = & (4 \times 1) \bmod 7 = 4
 \end{aligned}$$

Per  $11^6 \bmod 7$ : Con simili proprieta' ci si riconduce a calcolare  $4^6 \bmod 7$ .

$$4^6 \bmod 7 = (4^2 \bmod 7 \times 4^4 \bmod 7) \bmod 7$$

Similmente, siccome  $4^2 \bmod 7 = 2$ ,  $4^4 \bmod 7 = 4$

Quindi abbiamo  $2 \times 4 \bmod 7 = 1$ .

Notare che  $11^6 \bmod 7 = 1$  si poteva anche dedurre direttamente dal teorema di Fermat (secondo il quale, per  $p$  primo e  $n$  non multiplo di  $p$  vale  $n^{p-1} \bmod p = 1$ ).

**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  $3^8 \bmod 13 = 9$ .

**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 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 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 possibilità di starvation. Con un 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** Supponete che un processo esegua una operazione di decremento (detta anche wait) su un semaforo. Può modificarsi lo stato del processo a seguito di questa operazione? Come?

**Risposta**(Sketch) Se la risorsa gestita dal semaforo è “occupata”, ci sarà una transizione in stato “waiting”. Altrimenti il processo, o resta running, oppure può 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'ä 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

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

**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 + almeno uno degli esercizi sulla concorrenza.**

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

**Esercizio 1** [] Nei casi sotto, indicate se l'esecuzione concorrente delle 2 istruzioni puo' portare delle interferenze e perche' (si supponga  $x$  sia stata inizializzata correttamente, e che  $y, z$  siano altre variabili, diverse tra di loro)

1.  $x := x+1$              $x := x+1$

2.  $y := x$              $z := x$

**Risposta**(Sketch) solo nel primo caso, a causa della traduzione in linguaggio macchina necessaria prima della esecuzione (vedere appunti corso)

**Esercizio 2** []

Cosa e' il principio di localita' dei programmi? Menzionare un algoritmo visto a lezione per il quale questo principio e' importante, e spiegate brevemente perche'.

**Risposta**(Sketch) Vedi appunti di corso

**Esercizio 3** [] In un sistema a paginazione, spiegare cosa e' la tabella delle pagine e che informazioni contiene.

**Risposta**(Sketch) Vedi appunti di corso

**Esercizio 4** Disegnare il grafo che rappresenta gli stati possibili di un processo, e le transizioni tra stati. Per ogni transizione, indicare almeno una ragione che puo' causare quella transizione.

**Risposta**(Sketch) Vedi appunti di corso

**Esercizio 5** []

1. Considerate i 2 processi

<code>repeat</code>	<code>repeat</code>
<code>  print(A);</code>	<code>  print(B);</code>
<code>forever</code>	<code>forever</code>

Agite, se necessario, sul codice, inserendo opportune operazioni su semaforo, in modo che nell'output di questa esecuzione concorrente valga la seguente proprieta'. Ad ogni istante e' vero che:

$$-3 \leq \text{numero A stampate} - \text{numero di B stampate} \leq 3$$

Indicare bene i semafori utilizzati e il loro valore di inizializzazione. Motivare brevemente le risposte fornite.

Una soluzione sara' tanto migliore quanto piu' alto e' il numero di stringhe che possono essere stampate (in altre parole, quanto piu' alto e' il non-determinismo).

2. Se la vostra soluzione al punto precedente usa dei semafori con una inizializzazione diversa da 0, allora indicate le modifiche da fare per avere tutti i semafori inizializzati a 0.

**Risposta(Sketch)**

<code>repeat</code>	<code>repeat</code>
<code>  S.P()</code>	<code>  T.P()</code>
<code>  print(A);</code>	<code>  print(B);</code>
<code>  T.V()</code>	<code>  S.V()</code>
<code>forever</code>	<code>forever</code>

In fase di inizializzazione mettere  $S=3$ ,  $T=3$

per avere tutti i semafori inizializzati a 0, sufficiente mettere, prima dei loop 3 S.V e 3 T.V (posso essere messi in una posizione qualunque)

**Esercizio 6** In un sistema la memoria fisica e' divisa in  $2^{22}$  frame, un indirizzo logico e' scritto su 35 bit, e all'interno di una pagina, l'offset massimo e' 111111111 (sono 10 cifre). In questo esercizio, ignoriamo la presenza del validity bit, del dirty bit e altri bit di controllo nella tabella di pagine.

1. Quante entry ha la tabella di pagine piu' grande del sistema?
2. Quanti frame occupa la tabella di pagine piu' grande del sistema?
3. Il sistema deve adottare una paginazione a piu' livelli?

**Risposta(Sketch)**

1. Un frame/pagina e' grande  $2^{10}$  byte, e quindi la page table piu' grande puo' avere  $2^{(35-10)} = 2^{25}$  entry.
2. Nel sistema vi sono  $2^{22}$  frame, per cui sono necessari tre byte per scrivere il numero di un frame, e quindi la page table piu' grande occupa  $(2^{25} \times 3)/2^{10}$  frame, cioe'  $2^{15} \times 3$  frame.
3. Si perche' la page table piu' grande non puo' essere memorizzata in un unico frame.

**Esercizio 7** Si consideri uno scheduler che riceve 4 job A,B,C,D con le seguenti caratteristiche:

Job	durata	tempo di inizio
A	6	0
B	5	2
C	2	3
D	8	9

Calcolare il turnaround medio se lo scheduling impiegato e' SJF preemptive. Riportare il diagramma Gantt relativo.

**Risposta**(Sketch) SJF:

[A]3[C]5[A]8[B]13[D]21

Turnaround A=8, B=11, C=2, D=13. Medio :  $34/4$

### Esercizio 8 (Process scheduling) [3]

1. Cosa indica la "starvation" di un processo?
2. Puo' esserci starvation con la strategia RR?
3. E con la SJF?
4. E con una strategia a priorita'? Se si, come si potrebbe evitare il problema?

**Risposta**(Sketch) 1. un processo non ha mai la possibilita' di girare. 2. No. 3. Si. 4. Si; tecnica "aging".

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

2. Usare le proprieta' dell'aritmetica modulo per calcolare in modo semplice:

- $(5^2 \times 8^2) \bmod 9$  ;
- $11^{10} \bmod 9$

**Risposta**(Sketch)

1. metodi statistici consentono di risalire alla chiave facilmente, per messaggi abbastanza lunghi.
- 2.

$$\begin{aligned}
 & (5^2 \times 8^2) \bmod 9 \\
 = & ((5^2) \bmod 9 \times (8^2) \bmod 9) \bmod 9 \\
 = & (7 \times 1) \bmod 9 \\
 = & 7 \bmod 9
 \end{aligned}$$

Ora  $11^{10} \bmod 9$ . Intanto sfruttando le proprieta' di distributivita' di del modulo, questo e' la stessa cosa di  $2^{10} \bmod 9$

$$\begin{aligned}
 & 2^{10} \bmod 9 \\
 = & (2^2 \bmod 9 \times 2^8 \bmod 9) \bmod 9
 \end{aligned}$$

Calcolo  $2^8 \bmod 9$ :

Abbiamo  $4 = 2^2 \bmod 9$ , quindi  $2^4 \bmod 9 = (2^2 \bmod 9 \times 2^2 \bmod 9) \bmod 9 = 16 \bmod 9 = 7$ .

Similmente,  $2^8 \bmod 9 = 4$

Ne concludiamo che  $2^{10} \bmod 9 = 16 \bmod 9 = 7$ .

**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. Perché i moderni sistemi operativi usano una gestione paginata della memoria primaria?

2. Perché implementano normalmente anche la memoria virtuale?

**Esercizio 2** Per una periferica di I/O, cosa e' il controller? E il device driver? Fanno entrambi parte del sistema operativo?

**Esercizio 3** In un sistema con memoria virtuale le pagine sono formate da  $2^{10}$  byte, la RAM e' fatta di  $2^8$  frame, e lo spazio di indirizzamento logico e' di  $2^4$  pagine.

1. Qual e la lunghezza in bit di un indirizzo logico?

2. Qual e la lunghezza in bit di un indirizzo fisico?

**Risposta(Sketch)**

1.  $10 + 4$

2.  $10 + 8$

**Esercizio 4** Siano dati tre programmi concorrenti A, B e C, al cui interno vengono eseguite, rispettivamente, le procedure Pa, Pb e Pc. Si vuole essere certi che Pa verra' eseguita prima di Pb e Pc, mentre Pb e Pc devono poter essere eseguiti in qualsiasi ordine (quindi puo' succedere che venga eseguita prima Pb e poi Pc, o viceversa). Fornite una possibile soluzione al problema, usando semafori; usate il minor numero di semafori possibile.

**Risposta(Sketch)**

Semaphore sync = 0;

A	B	C
	P(sync);	P(sync);
Pa;	Pb;	Pc
V(sync);		
V(sync);		

**Esercizio 5** In un sistema operativo che adotta uno scheduling preemptive, quattro processi arrivano al tempo indicato e consumano la quantità di CPU indicata nella tabella sottostante)

Processo	T. di arrivo	Burst
P1	0	15
P2	3	8
P3	6	4
P4	11	3

Qual e' il waiting time medio migliore (ossia ottimale) che potrebbe essere ottenuto per lo scheduling dei quattro processi della tabella? Disegnare un diagramma di Gant che illustri la loro esecuzione

#### Risposta(Sketch)

Diagramma di GANT, assumendo come algoritmo di scheduling SJF preemptive:

(0)\_ P1\_ (3)\_ P2\_ (6)\_ P3\_(10)\_ P2\_ (11)\_ P4\_ (14)\_ P2\_ (18)\_ P1\_ (30)

Waiting time medio:

$$P1 = (30 - 0) - 15 = 15;$$

$$P2 = (18 - 3) - 8 = 7;$$

$$P3 = (10 - 6) - 4 = 0;$$

$$P4 = (14 - 11) - 3 = 0;$$

$$\text{waiting time medio} = 22/4$$

**Esercizio 6** In un sistema che implementa la paginazione della memoria ma non la memoria virtuale, un indirizzo fisico e' scritto su 32 bit, l'offset piu' grande all'interno di una pagina e' 1111 1111 (in binario), e la tabella delle pagine piu' grande del sistema occupa 24 megabyte (assumete che la tabella di pagine non contenga bit di controllo, sicurezza, dirty bit, etc).

1. Quanto e' grande lo spazio di indirizzamento logico del sistema?
2. Se il sistema ha un tempo di accesso in RAM di 100 ns (nanosecondi) e adotta un TLB con un tempo di accesso di 10 ns, quale hit rate minimo deve garantire il TLB perche' il sistema abbia un medium access time (mat) di 130 ns?

#### Risposta(Sketch)

1. Lo spazio di indirizzamento fisico del sistema e' diviso in  $2^{32}/2^8 = 2^{24}$  byte, dunque ci vogliono 24 bit, ossia 3 byte, per scrivere il numero di un frame, e questa e' la dimensione di una entry della tabella delle pagine del sistema. La tabella delle pagine piu' grande del sistema e' grande 24 megabyte, ossia  $3 * 2^{23}$  byte e quindi ha  $2^{23}$  entry. Lo spazio di indirizzamento logico quindi e' grande  $2^{23} * 2^8 = 2^{31}$  byte.

2. a partire dalla formula:

$$\text{mat} = \text{hit-rate} (110\text{ns}) + (1 - \text{hit-rate}) * (210\text{ns}) < 130\text{ns} \text{ ricaviamo:}$$

$$110\text{hit-rate} + 210 - 210\text{hit-rate} < 130;$$

quindi

$$80 < 100 \text{ hit-rate}$$

Quindi una hit ratio almeno dell'80 per cento.

**Esercizio 7** Consideriamo i seguenti task

P1	P2	Master
loop	loop	loop
<A1>	<A2>	<M>

1. Vogliamo che immediatamente dopo ogni operazione <Ai> (per i=1,2) sia eseguita l'operazione <M>. Non ci sono vincoli sull'ordine per le operazioni <Ai>. Proponete una soluzione usando semafori. Una soluzione sara' tanto migliore quanto piu' non-determinismo permette.
2. Aggiungete ora il vincolo che le operazioni <Ai> siano eseguite nell'ordine dato dal loro indice, ciclicamente. Proponete una soluzione usando semafori. (Una soluzione a questa domanda sarebbe anche una soluzione al punto precedente; si richiede tuttavia di fornire 2 soluzioni distinte, poiche' la soluzione al primo punto e' piu' semplice e puo' richiedere, per esempio, meno semafori.)

#### Risposta(Sketch)

1.	P1	P2	Master
	loop	loop	loop
	S.P;	S.P;	T.P;
	<A1>	<A2>	<M>
	T.V;	T.V;	S.V;

con inizialmente S=1 e T = 0.

2.	P1	P2	Master
	loop	loop	loop
	U1.P	U2.P	
	S.P;	S.P;	T.P;
	<A1>	<A2>	<M>
	T.V;	T.V;	S.V;
	U2.V;	U1.V	

con inizialmente U1=S=1 e U2=T = 0.

**Esercizio 8** Cos'e' il problema della frame allocation? Descrivere brevemente 2 politiche per questa situazione.

**Esercizio 9** Descrivere una soluzione al problema delle firme digitali (inviare messaggi firmati, in modo da essere certi sul sender del messaggio) usando la crittografia.

**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** — 1. A cosa serve il Process Control Block?

2. Quali informazioni contiene (indicare almeno 3 dei campi)?

3. Quanti Process Control Block esistono in un momento qualsiasi di attivita' di una macchina?

**Risposta(Sketch)** PCB rappresentano processi. Contiene informazioni importanti come process ID, process state, CPU state (program counter, register) per riprendere l'attivita' di un processo, informazioni per la gestione della memoria, etc. Il loro numero e' = numero dei processi (in generale c'e' una tabella per i PCB, quindi c'e' un upper bound)

**Esercizio 2** Pensando all'architettura di un processore, che cosa indicano i termini pipelining, superscalarita' e multicore?

**Esercizio 3** — 1. Qual'e' lo svantaggi principale del metodo di "file allocation" chiamato "Linked list" ?

2. Quali sono, sempre in questo metodo, i possibili pro e contro che emergono al variare delle dimensioni dei blocchi?

**Risposta(Sketch)** 1: Nel caso peggiore, un file system deve accedere tutti i blocchi di un file (caso di singly-linked list) per trovare i dati richiesti, o meta' (caso di doubly-linked list). 2: Un blocco grande riduce il numero di operazioni IO richieste per ritrovare un certo record, ma al costo di maggiore frammentazione interna; riducendo la dimensione e' il contrario

**Esercizio 4** Riportare il codice di 2 thread che, quando lanciati in parallelo, possano stampare, come tutti e soli output possibili, le sequenze **CAPRI** e **CARPI**.

I thread possono usare operazioni di semaforo e operazioni di stampa di singole lettere (esempio `print R`).

**Risposta(Sketch)** Varie soluzioni possibili. Esempio

$$T_1 = S.P(); printR; T.V();;$$
$$T_2 = printC; printA; S.V(); printP; T.P(); print I$$

con inizialmente tutti i semafori a 0.



**Esercizio 5** Si consideri il protocollo di mutual exclusion per i 2 processi mostrati sotto:

```

// Shared variables
boolean c1 = false, c2 = false;

// Process 1
<init1>;
while(true) {
    c1 = true;
    while (c2) { };
    <crit1>;
    c1 = false;
}

// Process 2
<init2>;
while(true) {
    c2 = true;
    while (c1) { };
    <crit2>;
    c2 = false;
}
```

dove `crit1` e `crit2` sono sezioni critiche. Il protocollo fornisce una soluzione al problema di Sezione Critica? (Come al solito, motivare la risposta)

**Risposta**(Sketch) No, puo' portare a deadlock (e' l'algoritmo 2 visto a lezione)

**Esercizio 6** Un hard disk ha la capienza di  $2^{26}$  byte, ed e' formattato in blocchi da 1024 byte. Viene adottata una allocazione indicizzata dello spazio su disco.

1. Quanti bit occorrono per memorizzare il numero di un blocco?
2. Si consideri un file A della dimensione di 800 K byte. Un solo blocco indice e' sufficiente a indirizzare tutti i blocchi di A? Quanti accessi al disco sono necessari per leggere l'ultimo blocco di A?

**Risposta**(Sketch)

1. Ci vogliono 16 bit (2 byte) per memorizzare il numero di un blocco, ottenuti con  $2^{26}/2^{10} = 2^{16}$
2. Un blocco indice puo' contenere  $1024/2 = 512$  puntatori a blocco.

Un solo blocco indice non e' sufficiente a indirizzare tutti i blocchi di A. Se si assume una allocazione indicizzata a schema concatenato, con un secondo blocco indice possiamo indirizzare tutti i blocchi di file. Poiche' e' gia' in RAM il numero del primo blocco indice, e' necessario leggere il primo blocco indice per recuperare il valore del secondo blocco indice, leggere il secondo blocco indice per sapere il numero dell'ultimo blocco del file, leggere questo blocco, per un totale di 3 letture in RAM (stesso risultato si ottiene assumendo una allocazione indicizzata a due livelli).

- Esercizio 7**
1. In un sistema in cui la memoria e' gestita con la paginazione, supponiamo che gli indirizzi logici siano di 31 bit, e che nella tabella di pagine si usino 8 byte per scrivere il numero di un frame. Quale dimensione minima dovrebbero avere le pagine di questo sistema per essere certi di non dover ricorrere ad una paginazione a piu' livelli?
  2. Se in un sistema non si vuole usare una paginazione a piu' livelli, quale soluzione alternativa si puo' adottare?

**Risposta**(Sketch)

1. Poniamo  $31 = m + n$  ( $m$  = bit usati per scrivere un numero di pagina,  $n$  = bit usati per scrivere l'offset).

Allora il numero di entry della PT piu' grande, moltiplicato per la dimensione di una entry deve poter essere contenuto in una pagina /frame, ossia:  $2^m \times 2^3 \leq 2^n$ .

Da cui:  $m + 3 \leq n$ . Poiche'  $m = 31 - n$ , risolvendo il semplice sistema si ha  $n = 17$ , ossia le pagine devono almeno essere grandi  $2^{17} = 128$  Kbyte.

2. Una inverted page table

**Esercizio 8** In un sistema con memoria virtuale, supponiamo che il grado di utilizzazione del processore (cpu) e del disco per la paginazione siano i seguenti:

1. cpu 15%, disco 95%
2. cpu 95%, disco 10%
3. cpu 20%, disco 5%

Ognuno di questi 3 casi: rappresenta una situazione ottimale o no? Perche'? Se la situazione non e' ottimale, cosa e' ragionevole faccia il sistema operativo?

**Risposta**(Sketch) 1. Thrashing: ridurre il livello di multiprogrammazione 2 okay 3. si puo' aumentare la multiprogrammazione

**Esercizio 9** Per i processi indicati nella tabella sotto, disegnare un diagramma di Gant che illustri la loro esecuzione usando:

- First-Come First Served
- Shortest Job First preemptive
- Round Robin (quantum = 2)

Indicare anche, nel sistema con RR, il waiting time medio.

Processi	Tempo di arrivo	Tempo di esecuzione
<i>A</i>	0.000	4
<i>B</i>	2.001	7
<i>C</i>	3.001	2
<i>D</i>	3.002	2

**Risposta**(Sketch)

1. [A] 4 [B] 11 [C] 13 [D]15 2. [A]4 [C]6 [D]8 [B]15 3. [A]2[A]4[B]6[C]8[D]10[B]12[B]14[B]15  
Waiting time :  $0+6+3+5=14/4=3.50$

**Esercizio 10** 1. In protocolli di sicurezza, descrivere il significato dei termini: confidentiality; message integrity; end-point authentication.

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

**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** Cos'e' il process control block? A cosa serve? Che informazioni contiene (indicate almeno 2 campi)?

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

**Esercizio 2** Considerate i seguenti processi:

P	Q
$x := x+1$	$x := x-1$
$x := x*2$	

Essi vengono lanciati in parallelo con valore di  $x$  iniziale 0. Alla fine della loro esecuzione,  $x$  puo' avere i valori 0, 1. Puo' avere altri valori?

**Risposta**(Sketch) Si, ci possono essere altri valori perche' c'e' una variabile  $x$  condivisa e le operazioni di incremento o moltiplicazione indicate non sono atomiche (le operazioni effettivamente eseguite sono quelle del linguaggio assembler). Per esempio  $x$  puo' assumere il valore  $-1$ . Si veda su questo le note corso su critical region.

**Esercizio 3** 1. Un file occupa, sull'hard disk, 3 blocchi. Descrivete come questo file viene memorizzato sull'hard disk secondo le tre tecniche fondamentali di allocazione dello spazio su disco. Si consiglia vivamente di accompagnare la descrizione ad opportuni disegni esemplificativi. Ad esempio, assumete che l'ipotetico hard disk su cui e' memorizzato il file sia formato da 16 blocchi in tutto, e che non contenga altri file. Scegliete anche dei valori adeguati – ovviamente compresi tra 0 e 15 – per i 3 blocchi in cui e' memorizzato il file.

2. Quale delle tre tecniche descritte e' meno adatta ad un accesso diretto ai dati del file, e perche'?

**Risposta**(Sketch)

1. vedi appunti di corso
2. L'allocazione concatenata, perche' sono necessari  $n$  accessi al disco per leggere i dati del file contenuti nell' $n$ -esimo blocco del file

**Esercizio 4** 1. Quali sono i vantaggi della paginazione della memoria?

2. E quali gli svantaggi?

**Risposta**(Sketch) Vantaggi: no frammentazione esterna; non necessario compattare; migliore gestione dimensioni variabili di processi

Svantaggi: Maggior lavoro per SO (gestione dei frame liberi, aumento del tempo di context switch), memoria presa da tabelle delle pagine, maggior tempo di traduzione degli indirizzi da logici a fisici, supporto hardware per tale traduzione (memoria associativa TLB)

**Esercizio 5** Un sistema ha una memoria RAM formata da 1024 frames di 512 byte ciascuno. Lo spazio di indirizzamento logico e' grande il quadruplo di quello fisico.

1. Qual e' l'indirizzo fisico piu' grande ammesso dal sistema?
2. Qual e' l'indirizzo logico piu' grande ammesso dal sistema?
3. In quante pagine e' suddiviso lo spazio di indirizzamento logico del sistema?
4. Se la tabella delle pagine e' mantenuta esclusivamente in RAM, di quanto si degradano approssimativamente le prestazioni del sistema rispetto ad un sistema equivalente ma senza paginazione della memoria?
5. In pratica, la tabella delle pagine, in quale posto (o quali posti) e' mantenuta? Perche' questo limita l'eccessiva perdita di prestazioni del sistema?

**Risposta**(Sketch)

1. La RAM e' formata da  $2^{10} * 2^9 = 2^{19}$  celle, il cui indirizzo va da 0 a  $2^{19} - 1$
2. Lo spazio logico e' pari a  $4 * 2^{19} = 2^{21}$  per cui indirizzo piu' grande e'  $2^{21} - 1$
3. Lo spazio di indirizzamento logico e' suddiviso in  $2^{21}/2^9 = 2^{12}$  pagine
4. Il tempo medio di esecuzione dei programmi raddoppia, poiche' il tempo di accesso ad ogni cella di memoria (dovendo passare attraverso la tabella delle pagine) raddoppia
5. Si usa una memoria associativa di supporto (TLB). Vedere appunti di corso.

**Esercizio 6** Sotto, e' riportato il programma java che implementa il "bounded buffer" usando semafori, come visto a lezione. I produttori chiamano il metodo **enter** ed i consumatori il metodo **remove**.

```
public class BoundedBuffer
{
    public BoundedBuffer()
    {
        in = 0;
        out = 0;

        buffer = new Object[BUFFER_SIZE];
        mutex = new Semaphore(1);
        empty = new Semaphore(BUFFER_SIZE);
        full = new Semaphore(0);
    }
    public void enter(Object item) {
        empty.P();
```

```

        mutex.P();

        buffer[in] = item;
        in = (in + 1) % BUFFER_SIZE;

        mutex.V();
        full.V();
    }
    public Object remove() {
        full.P();
        mutex.P();

        Object item = buffer[out];
        out = (out + 1) % BUFFER_SIZE;

        mutex.V();
        empty.V();
        return item;
    }
    private static final int    BUFFER_SIZE = 2;
    private Semaphore mutex;
    private Semaphore empty;
    private Semaphore full;
    private int in, out;
    private Object[] buffer;
}

```

Rispondere alle domande seguenti:

1. Che problemi ci sarebbero se il semaforo **mutex** fosse tolto?
2. Che problemi ci sarebbero se i semafori **empty**, **full** fossero tolti?
3. Nel codice per il metodo **remove**, il comando **full.P()** precede **mutex.P()**. E' possibile modificare l'ordine di questi 2 comandi?
4. Nel codice per il metodo **remove**, il comando **mutex.V()** precede il comando **empty.V()**. E' possibile modificare l'ordine di questi 2 comandi?

**Risposta**(Sketch) Alla 1a domanda: mutex implementa la mutua exclusion su accesso buffer. Senza questo semaforo quindi piu' processi potrebbero lavorare in contemporanea sul buffer (esempio piu' produttori e allora ci potrebbe essere conflitto su operazioni su indice **in**)

Alla 2a domanda: full e empty servono per verificare le condizioni di accesso buffer; senza di essi un produttore potrebbe inserire nel buffer anche se questo e' pieno, e similmente per un consumatore a buffer vuoto.

Alla 3a domanda: L'ordine delle 2 operazioni non puo' essere invertito, poiche' si potrebbe avere un deadlock (bloccaggio) da parte di produttori e consumatori.

alla 4a: L'ordine delle 2 operazioni potrebbe essere invertito, e la soluzione sarebbe ancora corretta.

**Esercizio 7** 1. Supponiamo che un sistema prende 1 unita' di tempo per fare un a context switch. Abbiamo un sistema con 3 task; il primo richiede 70 unita' di tempo, il secondo 25 unita' e il terzo 45. Dato uno scheduler di tipo round robin, con quanto pari a 20 unita' di tempo, mostrate con un diagramma di Gantt quale task gira e quando. Calcolate poi il waiting time di ogni task, e impostate la formula per il waiting time medio.

2. Quale o quali sono a vostro avviso i vantaggi principali della politica di round robin?

**Risposta**(Sketch)

[A 20]21[B 41]42[C 62]63[A 83]84[B 89]90[C 110] 111 [A 131] 132 [C 137]138[A 148]

Vantaggio RR: responsiveness

**Esercizio 8 (CPU scheduling)** 1. In pratica, nel CPU scheduling dei sistemi operativi moderni si usano 'multilevel queue' (code multilivelli). Cosa sono e perche' si usano?

2. Cosa e' la process affinity di un processo rispetto ad certo processore?

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

**Esercizio 9** 1. Cifratura a chiave simmetrica; perche' una permutazione sull'alfabeto non e' considerata una buona chiave?

2. Nella "cifratura a blocchi", con blocchi di 64 bit, e' fattibile usare come chiave una qualunque permutazione di questi bit, rappresentata da una tabella che specifica come viene tradotta ogni sequenza di bit?

**Risposta**(Sketch) 1a domanda: attacchi di tipo statistico

2a domanda: con 64 bit una tabella avrebbe  $2^{64}$  entry, troppo grande per essere manipolata (esempio: comunicata o aggiornata)

**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** Cosa e' la tabella delle pagine? Cosa contiene, e come si usa?

**Esercizio 2** Cosa e' il 'context switch'? (Cosa succede quando avviene, e perche' avviene)

**Esercizio 3** Con riferimento allo scheduling del processore, qual'e' la differenza tra un algoritmo di tipo preemptive e uno non-preemptive (cioe' con o senza prelazione)?

**Esercizio 4** || Considerate i 2 processi sotto

```
P1 = loop forever      P2 = loop forever
    print(A)            print(B)
```

Agite, se necessario, sul codice, inserendo opportune operazioni su semaforo, in modo che nell'output di ogni esecuzione, ad ogni istante il numero di A sia maggiore o uguale al numero di B.

**Risposta(Sketch)**

```
P1 = loop forever      P2 = loop forever
    print(A)            P(S); print(B)
    V(S)
```

con  $S = 0$

**Esercizio 5** Si ha un hard disk di capienza di  $2^{41}$  byte, e formattato in blocchi di taglia 4096 byte.

1. Che taglia ha un puntatore a blocco?
2. Abbiamo un file della dimensione di 16384 byte. Quanti accessi al disco saranno necessari per leggere l'ultimo blocco del file se viene adottata una allocazione concatenata dello spazio su disco? Si assuma, come al solito, che sia gia' in RAM il numero del primo blocco del file stesso.

**Risposta(Sketch)** I blocchi sono  $2^{41}/2^{12} = 2^{29}$ . Pertanto un puntatore a blocco prende 4 B (necessary per coprire 29 bit).

Ogni blocco memorizza 4092 byte di dati piu' 4 byte di puntatore al blocco successivo. Il file in B e' grande  $16384 = 2^{14}$ ; senza i puntatore sarebbe  $2^{14}/2^{12} = 4$  blocchi, con i puntatori diventano 5. Per cui sono necessari 5 accessi.

**Esercizio 6** Supponiamo che le pagine in uno spazio di indirizzi virtuali siano usate secondo l'ordine seguente (quella sotto e' cioe' la "reference stream")

1 2 1 3 2 1 4 3 1 1 2 4 1 5 6 2 1

Supponiamo di avere 3 frame disponibili, inizialmente liberi. Assumiamo anche che le decisioni sulla paginazione siano fatte su domanda, cioe' quando un page fault ha luogo. Mostrare i contenuti dei frame dopo ogni accesso a memoria, secondo la politica LRU. Quanti page fault occorrono?

**Risposta**

Frame	Reference																	
	1	2	1	3	2	1	4	3	1	1	2	4	1	5	6	2	1	
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	
1		2	2	2	2	2	2	3	3	3	3	4	4	4	6	6	6	
2				3	3	3	4	4	4	4	2	2	2	5	5	5	1	
Fault?	Y	Y	N	Y	N	N	Y	Y	N	N	Y	Y	N	Y	Y	Y	Y	

Numero totale faults = 11.

**Esercizio 7** Supponiamo di avere un algoritmo di CPU scheduling non-preemptive, e 4 processi con tempi di esecuzione 5, 18, 9, 12. In quale ordine dovrebbero essere eseguiti per minimizzare il "waiting time" medio ?

**Risposta**(Sketch) seguendo shortest job first: 5,9,12,18

**Esercizio 8** [] Considerate i 2 processi sotto

```
P1 = loop forever      P2 = loop forever
        print(A)                print(B)
```

Agite, se necessario, sul codice, inserendo opportune operazioni su semaforo, in modo che l'output di questa esecuzione concorrente sia la stringa ABBABBABB...

**Risposta**(Sketch)

```
P1 = loop forever      P2 = loop forever
    P(S) P(S)          P(T)
    print(A)            print(B)
    V(T);V(T)          V(S)
```

con S = 2, T = 0

**Esercizio 9** Considerate un sistema operativo A che implementa la memoria virtuale, e un altro sistema operativo B che implementa la paginazione senza memoria virtuale. Su quale dei 2 sistemi l'effettivo tempo di esecuzione di un programma sara' minore, e perche'?



**Risposta**(Sketch) Non si sa. Potrebbe essere piu' veloce su A, perche' si potra' caricare solo una parte del programma. Ma potrebbe essere piu' veloce su B se l'esecuzione su A produce molti page fault e il caricamento dell'intero programma non e' troppo penalizzante.

**Esercizio 10** Quando succede che la tabella delle pagine debba essere paginata? In questi casi come sara' allora strutturato un indirizzo logico?

Ci potrebbero essere problemi di efficienza dovuti alla paginazione a piu' livelli? Se si, come vengono risolti?

**Risposta**(Sketch) Note corso

**Esercizio 11** Descrivere una soluzione al problema delle firme digitali (inviare messaggi firmati, in modo da essere certi sul sender del messaggio) usando la crittografia.

Prendete il vostro numero di matricola. Eliminate tutti gli “0”, e chiamate  $N$  il numero risultante, e  $m$  la sua lunghezza. Esempio se il numero di matricola e’ 0000800470, allora  $N = 847$  e  $m = 3$

Nel seguito  $N_i$  e’ la cifra di  $N$  in posizione  $i \bmod m$ . Cioe’: si contano le cifre di  $N$  da sinistra verso destra, la prima cifra e’  $N_0$ , poi  $N_1$  e cosi’ di seguito, modulo la lunghezza di  $N$ . Nell’esempio sopra:

- $N_0 = 8$
- $N_1 = 4$
- $N_2 = 7$
- $N_3 = 8$
- $N_4 = 4$

Inoltre allo stesso modo  $N_2N_3$  e’ il numero 78 (e **non la moltiplicazione**  $7 \times 8$ ).

Negli esercizi sotto: non eseguite calcoli o ragionamenti astratti in termini di questi  $N_i$ , ma sostituite subito ai vari  $N_i$  il valore concreto dato dal vostro numero matricola.

---

**Ad inizio foglio indicate chiaramente il vostro numeri di matricola e quali sono nel vostro caso i numeri  $N_0, N_1, N_2, N_3$ . Inoltre all’inizio di ogni esercizio in cui intervengono tali numeri, indicatene il valore**

Motivare le risposte date. Non e’ necessario dare risposte 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).

**Ricordare che lo scan finale deve contenere il libretto universitario di riconoscimento; e che il compito da inviato in pdf e in un unico file, come allegato**

---

**E’ necessario svolgere gli esercizi 1 e 2**

---

**Esercizio 1** Dati numeri  $x, y, z$ , sia  $\overline{[x, y, z]}$  la stringa ottenuta ordinando i 3 numeri in ordine non decrescente (esempio: se  $x = 6, y = 2, z = 4$  allora  $\overline{[x, y, z]} = 2, 4, 6$ ).

Considerate i seguenti processi:

P1	P2	P3
<pre>loop forever print(N<sub>1</sub>)</pre>	<pre>loop forever print(N<sub>2</sub>)</pre>	<pre>loop forever print(N<sub>3</sub>)</pre>

Agite, se necessario, sul codice, inserendo opportune operazioni su semaforo, in modo che l'output di questa esecuzione concorrente sia la stringa infinita

$\overline{[N_1, N_2, N_3]} \overline{[N_1, N_2, N_3]} \overline{[N_1, N_2, N_3]} \dots$

**Risposta(Sketch)** Assumo  $N_1 = 4, N_2 = 6, N_3 = 1$ . Quindi occorre stampare la stringa

146146146....

Semafori  $S, U$  inizializzati a 0, e  $T$  inizializzato a 1

P1	P2	P3
<pre>loop forever P(S) print(4) V(U)</pre>	<pre>loop forever P(U) print(6) V(T)</pre>	<pre>loop forever P(T) print(1) V(S)</pre>

Volendo, anche  $T$  si può inizializzare a 0, ma occorre spostare  $P(T)$  in fondo.

**Esercizio 2** In un sistema operativo che adotta uno scheduling di tipo Shortest Job First preemptive, quattro processi arrivano al tempo indicato e consumano la quantità di CPU indicata nella tabella sottostante:

Processo	T. di arrivo	Burst time
A	0	$N_2 + 10$
B	3	$N_3 + 6$
C	6	$N_1 + 2$
D	7	$N_2$

1. Riportate il diagramma di Gantt che mostra ciò che succederà in esecuzione.
2. Qual è il waiting time del processo C?

Nota: dire che un processo arriva a tempo  $x$  significa dire che a tempo  $x$  quel processo è effettivamente presente in coda.

Se ci sono scelte che fate (relativamente a decisioni da prendere che secondo voi non sono specificate nella politica SJF, indicatelo esplicitamente).

**Risposta(Sketch)** Svolgimento, assumendo  $N_2 = 7, N_3 = 6, N_1 = 8$  Quindi abbiamo

Processo	T. di arrivo	Burst time
A	0	17
B	3	12
C	6	10
D	7	7

Riporto Gantt ed evoluzione pila:

[0] A [3] B [7] D [14] B [22] C [32] A [46]

Waiting time per C e' :  $32 - 6 - 10 = 16$

**Esercizio 3** Sia  $M$  il massimo tra i numeri  $N_1$  e 4.

Un hard disk ha la capienza di  $2^{MN_2}$  byte, ed è formattato in blocchi da  $2^{(N_3+10)}$  byte.

1. Quanti sono i blocchi nel sistema?
2. Quando spazio prende un puntatore a blocco?
3. Quanti accessi al disco sono necessari per leggere l'ultimo blocco di un file A della dimensione di  $2^{(N_3+10)}$  byte, assumendo che sia già in RAM il numero del primo blocco del file stesso e che venga adottata una allocazione concatenata dello spazio su disco?
4. Nella allocazione concatenata, che forme di spreco di memoria sono presenti?

**Risposta(Sketch)** Assumo  $N_1 = 3, N_2 = 7, N_3 = 3$ .

1. I blocchi sono  $2^{47}/2^{13} = 2^{34}$ .
2. Un puntatore necessita di 34 bit, quindi 5 byte.
3. A è memorizzato su 2 blocchi. Quindi 2 accessi.
4. La memoria sprecata per puntatori a blocchi; e la memoria inutilizzata nell'ultimo blocco di un file.

**Esercizio 4** Considerate i seguenti processi:

P1	P2
loop forever	loop forever
print A	print B

Agite, se necessario, sul codice, inserendo opportune operazioni su semaforo, in modo che nell'output di questa esecuzione concorrente, ad ogni istante, valga, usando  $\#(A)$  per il numero di occorrenze di  $A$  e similmente per  $\#(B)$  :

- $\#(A) - \#(B) \leq N_2$ ,
- $\#(B) - \#(A) \leq N_3$

(quelle sopra sono disequazioni tra numeri interi)

**Risposta(Sketch)** Assumo  $N_2 = 5, N_3 = 3$ .

Inizializzo  $S = 5, T = 3$

P1	P2
loop forever	loop forever
P(S)	P(T)
print A	print B
V(T)	V(S)

**Esercizio 5** 1. Nella crittografia, cosa significa adottare una codifica ‘a blocchi’ ?

2. Come si evita che blocchi uguali diano codifiche identiche?

3. Supponiamo che i messaggi siano sequenze di bit, e che un blocco sia fatto da 3 bit. Seguendo il metodo (o uno dei metodi) che avete descritto al punto (2) precedente, mostrate come viene tradotto il messaggio  $AB$  dove  $A$  e’ il numero binario di 3 cifre che rappresenta il minimo tra i numeri  $N_2$  e 7, e  $B$  e’ lo stesso per il minimo tra i numeri  $N_3$  e 6.

Come tabella di mapping, prendere la seguente:

Input	Output	Input	Output
000	110	100	001
001	100	101	111
010	011	110	010
011	000	111	101

Fate vedere anche come viene fatta la decodifica.

Eventuali altri parametri che intervengono nella traduzione possono essere liberamente scelti (ovviamente indicandoli).

**Risposta(Sketch)** Per i primi 2 punti, vedere note del corso.

Svolgimento, assumendo  $N_2 = 5, N_3 = 6$

Quindi in questo caso  $AB$  sarebbe 56 in binario, cioè 101110.

Faccio la traduzione assumendo un random Initialisation Vector (IV), diciamo 110.

Ora per la traduzione faccio così. Il primo blocco di ciphertext e’ ottenuto applicando la tabella allo XOR tra  $A$  e IV. Indicando con  $\#$  lo XOR, con  $K(-)$  l’applicazione della tabella, e con  $c(1)$  il primo blocco di ciphertext, abbiamo

$$c(1) = K(A \# IV) = K(101 \# 110) = K(011) = 000$$

Per  $c(2)$ , si usa come random vector  $c(1)$ , quindi abbiamo

$$c(2) = K(B \# c(1)) = K(110 \# 000) = K(110) = 010$$

Facciamo la decodifica, indicando con  $K^{-1}$  l’inverso della tabella:

$$A = K^{-1}(c(1)) \# IV = K^{-1}(000) \# 110 = 011 \# 110 = 101$$

$$B = K^{-1}(c(2)) \# c(1) = K^{-1}(010) \# 000 = 110 \# 000 = 110$$

**Esercizio 6** Supponiamo gli indirizzi logici di un sistema usino  $N_0$  bits per il numero di pagina, e  $N_1$  bits per l’offset.

1. Quanto e’ grande al massimo lo spazio di indirizzamento logico di un processo del sistema?

2. Quante entry ci saranno al massimo in una page table?

3. Supponiamo che in un indirizzo logico il numero di pagina sia  $p$  e l’offset sia  $d$ . Indicare le azioni che devono essere svolte per costruire l’indirizzo fisico finale.

**Risposta(Sketch)**

1.  $2^{N_0+N_1}$

2.  $2^{N_0}$

3. vedere note di corso

**Esercizio 7** In cosa consiste la tecnica della memoria virtuale, e perché è importante?  
Quali sono i suoi vantaggi o inconvenienti?

**Risposta**(Sketch) Vedere note del corso

Prendete il vostro numero di matricola. Eliminate tutti gli “0”, e chiamate  $N$  il numero risultante, e  $m$  la sua lunghezza. Esempio se il numero di matricola e’ 0000800470, allora  $N = 847$  e  $m = 3$

Nel seguito  $N_i$  e’ la cifra di  $N$  in posizione  $i \bmod m$ . Cioe’: si contano le cifre di  $N$  da sinistra verso destra, la prima cifra e’  $N_0$ , poi  $N_1$  e cosi’ di seguito, modulo la lunghezza di  $N$ . Nell’esempio sopra:

- $N_0 = 8$
- $N_1 = 4$
- $N_2 = 7$
- $N_3 = 8$
- $N_4 = 4$

Inoltre allo stesso modo  $N_2N_3$  e’ il numero 78 (e **non la moltiplicazione**  $7 \times 8$ ).

Negli esercizi sotto: non eseguite calcoli o ragionamenti astratti in termini di questi  $N_i$ , ma sostituite subito ai vari  $N_i$  il valore concreto dato dal vostro numero matricola.

---

**Ad inizio foglio indicate chiaramente il vostro numeri di matricola e quali sono nel vostro caso i numeri  $N_0, N_1, N_2, N_3$ . Inoltre all’inizio di ogni esercizio in cui intervengono tali numeri, indicatene il valore**

Motivare le risposte date. Non e’ necessario dare risposte 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).

**Ricordare che lo scan finale deve contenere il libretto universitario di riconoscimento; e che il compito va inviato in pdf e in un unico file, come allegato**

---

**E’ necessario svolgere gli esercizi 1 , 2, 3**

---

**Esercizio 1** Considerate i seguenti processi:

P1	P2	P3	P4
<code>print(<math>N_1</math>)</code>	<code>print(<math>N_2</math>)</code>	<code>print(<math>N_2</math>)</code>	<code>print(<math>N_3</math>)</code>

Agite, se necessario, sul codice, inserendo opportune operazioni su semaforo, in modo che l'output di questa esecuzione concorrente sia una stringa in cui i numeri stampati appaiano in ordine *non decrescente*.

In caso di possibilita' di non-determinismo (esempio, quello prodotto dai processi P2 e P3, che stampano lo stesso numero): meglio se la vostra soluzione riflette tale non-determinismo (esempio, nel caso di P2 e P3, lascia aperta la possibilita' che esegua per primo uno qualunque dei 2 processi).

**Risposta**(Sketch) Assumo  $N_1 = 4, N_2 = 6, N_3 = 1$ . Quindi occorre stampare la stringa 1466

Semafori  $S, T$  inizializzati a 0

P1	P2	P3	P4
<code>S.P()</code>	<code>T.P()</code>	<code>T.P()</code>	
<code>print 4</code>	<code>print 6</code>	<code>print 6</code>	<code>print 1</code>
<code>T.V()</code>			<code>S.V()</code>
<code>T.V()</code>			

Esempio di altra situazione interessante:

P1	P2	P3	P4
<code>S.P()</code>			
<code>S.P()</code>	<code>T.P()</code>	<code>T.P()</code>	
<code>print 4</code>	<code>print 3</code>	<code>print 3</code>	<code>print 1</code>
	<code>S.V()</code>	<code>S.V()</code>	<code>T.V()</code>
			<code>T.V()</code>

**Esercizio 2** Sia:  $N$  il max tra  $N_2$  e 5; ed  $M$  il max tra  $N_1$  e 6.

Quattro processi arrivano al tempo indicato e consumano la quantita' di CPU indicata nella tabella sottostante:

Processo	T. di arrivo	Burst time
A	0	$N_2 + 7$
B	4	$N_3 + 4$
C	7	$M$
D	10	$N$

1. Riportate il diagramma di Gantt che mostra cio' che succedera' in esecuzione, adottando uno scheduling di tipo Round Robin con quanto di tempo 3.
2. Lo stesso, in questo caso assumendo Shortest Job First di tipo non-premptive.
3. Calcolare il waiting time di B nel 2o caso.



**Risposta(Sketch)** Assumiamo  $N_1 = 5, N_2 = 4, N_3 = 6$  quindi  $N = 5, M = 6$ .

Processo	T. di arrivo	Burst time
A	0	11
B	4	10
C	7	6
D	10	5

[0]A[3]A[6]B[9]A[12]C[15]B[18]D[21]A[23]C[26]B[29]D[31]B[32]

Evoluzione processi nella ready-queue :

0 A  
 3 A  
 6 BA  
 9 ACB  
 12 CBDA  
 15 BDAC  
 18 DACB  
 21 ACBD  
 23 CBD  
 26 BD  
 29 DB  
 31 B  
 32

SJF:

[0]A[11] D[5] C[6] B [32]

WT (B) = 18 (22-4)

**Esercizio 3** Spiegare cosa e' il Process Control Block, cosa contiene (almeno 2 campi), e a cosa serve.

**Risposta(Sketch)** Vedere note corso.

**Esercizio 4** []

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 6 7 3. Indicare chiaramente i page fault. Quanti ne occorrono?

2. Per il punto sopra, calcolare il tempo medio di accesso assumendo che: il tempo di swap e' 20 millisecondi; il tempo di accesso alla memoria e' 10 microsecondi; in media 500 accessi sono effettuati ad una pagina una volta che questa e' chiamata (cioe' ogni volta che compare nella reference string).

**Risposta(Sketch)**

1. Ci sono 4 page fault. Ci sono varie evoluzioni possibili, tra cui questa:

4	2	
6		5
7		
3	1	3

Si poteva finire anche con, esempio

2		4	5
6	oppure	6	2
7		3	3 .....
3		5	1

2.  $( (4*20*1000) + 9*10*500 ) / 9*500$

(dove 4 e' dato dal numero di page fault, e 9 sono le pagine utilizzate, ognuna 500 volte).

**Esercizio 5** Si consideri un sistema in cui:

- Un indirizzo fisico del sistema e' scritto su  $(20 + N_2)$  bit, e la RAM e' suddivisa in  $2^{N_3+10}$  frame.
- in una tabella delle pagine di un processo, l'indice piu' grande usabile nella tabella delle pagine di quel processo puo' essere

$$\underbrace{1 \dots 1}_{N_4+10} \text{ (in binario).}$$

1. Come e' fatto l'indirizzo fisico ?
2. Quanto e' grande una pagina?
3. Quanto e' grande lo spazio di indirizzamento logico del sistema (cioe' il processo piu' grande che il sistema ammette) ?
4. Il sistema deve usare una paginazione a piu' livelli? (motivate numericamente la vostra risposta).

**Risposta(Sketch)** Risposta, nel caso  $N_4 = 5, N_3 = 3, N_2 = 5$ .

1. 25 bit di IF; 13 bit di n. frame  
Quindi 12 bit di offset

$$\begin{array}{cc} \text{[N frame]} & \text{[ offset ]} \\ 13 & 12 \end{array}$$

2. un numero di frame e' scritto su 13 bit, e la dimensione di un frame, e quindi di una pagina, e' di  $2^{12}$  byte ( $25 - 13 = 12$ ).
3. Poiche' il numero piu' grande di una pagina e' 11111111111111, ci possono essere al massimo  $2^{15}$  pagine, e lo spazio di indirizzamento logico e' di  $2^{15} \times 2^{12}$  byte, cioe'  $2^{27}$  byte
4. Si. La tabella delle pagine piu' grande del sistema ha  $2^{15}$  entry, e ogni entry contiene il numero di un frame, per cui sono necessari almeno due byte per ogni entry. La dimensione di quella tabella sara' quindi di  $2^{15} \times 2 = 2^{16}$  byte ossia maggiore della dimensione di un frame.

**Esercizio 6** Sia  $N = N_1$ , se  $N_1$  diverso da 1; altrimenti  $N$  e' il max tra 3 e  $N_2$ .  
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

$$\underbrace{A \dots A}_N B \underbrace{A \dots A}_N B \dots$$

Le uniche modifiche possibili sono aggiunta di operazioni su semafori.

**Risposta(Sketch)** Supponiamo che  $N$  sia 2 (per altri numeri, la soluzione e' simile)

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

con semafori = 0

**Esercizio 7** 1. Nella cifratura a chiave simmetrica, perche' una permutazione sull'alfabeto non e' considerata una buona chiave?

2. Siano:

- $N = N_2$ , se  $N_2$  diverso da 7 e diverso da 1;  $N = 6$  altrimenti;
- $M = N_1$ , se  $N_1$  diverso da 7 e diverso da 1;  $M = 11$  altrimenti

Usare le proprieta' dell'aritmetica modulo per calcolare in modo semplice

$$(N^{11} \times M^{N_3+5}) \bmod 7$$

**Risposta(Sketch)**

1. note corso

2. soluzione per  $N_2 = 6, M = 11, N_3 = 4$ .

$$(6^{11} \times 11^9) \bmod 7$$

$$(6^{11} \times 4^9) \bmod 7$$

poiche'  $11 \bmod 7 = 4$ .

Inoltre abbiamo  $6^2 \bmod 7 = 36 \bmod 7 = 1$

Quindi anche  $6^4 \bmod 7 = 6^8 \bmod 7 = 1$ .

Siccome  $11 = 8 + 4 + 1$ ,  $6^{11} \bmod 7 = 6 \bmod 7 = 6$ .

Similmente  $4^2 \bmod 7 = 16 \bmod 7 = 2$

Quindi

$$4^4 \bmod 7 = 2^2 \bmod 7 = 4$$

$$4^8 \bmod 7 = 4^2 \bmod 7 = 2$$

Quindi  $4^9 \bmod 7 = (4 \times 2) \bmod 7 = 1$

Quindi  $(6^{11} \times 4^9) \bmod 7 = 6$

**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** Cos'e' la memoria cache? A cosa serve? Dove si trova?

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

**Esercizio 2** 1. Cosa e' un page fault?

2. Nella gestione dei page fault da parte del sistema operativo, qual'e' il passo piu' costoso (in termini di tempo)?

**Esercizio 3** Cosa significa che un processo e' in stato ready?

Con quali algoritmi e' gestito l'insieme dei processi ready?

**Esercizio 4** In una stringa, usiamo la notazione {AB} per dire che in quel punto la stringa puo' avere sia AB che BA.

Considerare questi 2 processi, lanciati in parallelo:

P1	P2
loop forever	loop forever
print<A>	print<B>

Volgiamo che le strighe stampate siano {AB}{AB}....

Proponete una soluzione usando semafori. Indicare i valori di inizializzazione. (Come al solito, una soluzione con non-determinismo, che ammette piu' stringhe stampate, e' preferibile ad una deterministica.)

Proponete anche soluzioni (se non gia' date per il punto precedente) in cui tutti i semafori siano inizializzati a 0, e tutti a 1.

**Risposta**(Sketch)

P1	P2
loop forever	loop forever
print<A>	print<B>
S.V	T.V
T.P	S.P

con S,T inizialmente = 0.

P1	P2
loop forever	loop forever
T.P	S.P
print<A>	print<B>
S.V	T.V

con S,T inizialmente = 1.

**Esercizio 5** Usiamo la stessa notazione per stringhe dell'esercizio sopra.  
Considerare questi 3 processi, lanciati in parallelo:

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

Volgiamo che le stringhe stampate siano: A{BC}A{BC}...

Proponete una soluzione usando semafori. Indicare i valori di inizializzazione.

**Risposta**(Sketch)

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

con  $S = 2$  e gli altri inizialmente a 0. Nota: semafori Z,U necessari, per evitare il rischio di potere stampare 2 B o 2 C consecutive

Un'altra soluzione:

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

con  $S = 2$  e gli altri inizialmente a 0.

**Esercizio 6** Si consideri uno scheduler che riceve 4 job A,B,C,D con le seguenti caratteristiche:

Job	durata	stimata	tempo di inizio
A	30 ms		0
B	60 ms		0
C	12 ms		0
D	36 ms		0

Calcolare il turnaround medio a seconda che lo scheduling impiegato sia

1. SJF
2. Round robin con quanto di tempo pari a 12ms

Si consideri (in entrambi i casi) un tempo di context switch pari a 1ms.

Riportare sempre il diagramma Gantt relativo.

**Risposta(Sketch)**

SJF, nelle versioni preemptive e non, dà lo stesso risultato perché i processi sono tutti presenti all'inizio

SJF

0- C-12 13 -A- 43 44 -D- 81 81 -B- 141

turnaround : A = 43, B= 141, C=12, D = 80

turnaround medio 276/6

RR

0 -A- 12 13 -B- 25 26 -C- 38 39 -D- 51 52 -A- 64 65 -B- 77 78 -D- 90 91  
-A- 97 98 -B- 110 111 -D- 123 124 -B- 136 137 -B- 149

turnaround : A = 97, B= 149, C=38, D = 123

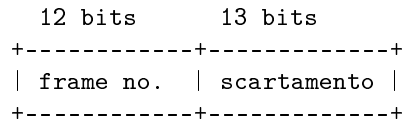
**Esercizio 7** Si consideri un sistema con indirizzi logici di 32-bit, fisici di 25-bit, e che usa la paginazione per gestire la memoria. Una pagina contiene 8192 bytes di RAM.

1. Quanti bytes di RAM ci sono nel sistema?
2. E quanti frame ?
3. Mostrare la composizione degli indirizzi fisici e logici (quanti bit hanno questi indirizzi, e come sono strutturati).
4. Quante entry al più contiene la tabella di pagine di un dato processo?
5. Può essere necessaria la paginazione della tabella di pagine di un processo?

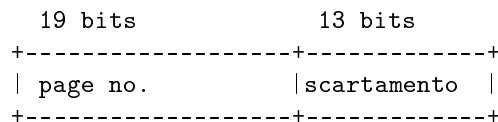
**Risposta(Sketch)**

1. 33.554.432 bytes ( $2^{25}$  bytes o 32 Mbytes).
2. 4.096 frame ( $2^{12}$ ).
3. Mostrare la composizione degli indirizzi fisici e logici (quanti bit hanno questi indirizzi, e come sono strutturati).

#### FISICI



#### LOGICI



4. 524.288 entry ( $2^{19}$  entry).
5. Sì, poiché il numero di entry è superiore alla dimensione di un frame  $2^{13}$ , quindi anche supponendo 2 byte soltanto per ogni riga della tabella, si potrebbero superare le dimensioni di un frame.

**Esercizio 8** In una tabella di pagine, cos'è il valid/invalid bit? A cosa serve?  
E il dirty bit? Cos'è, e a cosa serve?

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

**Esercizio 9** La politica LRU, per page replacement, non è utilizzabile in pratica, e deve essere approssimata. Perché? Indicare chiaramente i problemi che la disciplina presenta.

**Risposta**(Sketch) For paging, the updating is required for every memory reference and this adds considerable overhead making it impractical. Would need sophisticated HW support.

**Esercizio 10** Descrivere una soluzione al problema delle firme digitali (inviare messaggi firmati, in modo da essere certi sul sender del messaggio) usando la crittografia.



**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** Considerate i seguenti 2 processi:

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

Usando semafori, assicurarsi che le uniche stringhe possibili dai 2 processi quando lanciati in parallelo siano ACBDE e ACBED

**Risposta**(Sketch)

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

**Esercizio 2** Come funziona l'allocazione di file di tipo 'indicizzato'? Che vantaggi/svantaggi presenta?

**Esercizio 3** Cosa e' un algoritmo di CPU scheduling? Cosa deve cercare di ottimizzare? (non interessano descrizioni di algoritmi specifici)

**Esercizio 4** Cosa e' una interrupt? Come viene gestita dal Sistema Operativo?

**Esercizio 5** In un sistema operativo che adotta uno scheduling con preemption, quattro processi arrivano al tempo indicato e consumano la quantità di CPU indicata nella tabella sottostante)

Processo	T. di arrivo	Burst
P1	0	10
P2	2	7
P3	5	3
P4	9	2

1. Quale strategia permette di ottenere il waiting time medio migliore (ossia ottimale) per lo scheduling dei quattro processi della tabella? Riportate il corrispondente diagramma di Gantt.

2. All'interno di un sistema operativo, un certo processo P e' correntemente in stato di "Ready", e si sa che, una volta acquisita la CPU, non dovra' piu' rilasciarla volontariamente prima di aver terminato la propria esecuzione (in altre parole, non dovra' piu' eseguire operazioni di I/O, di sincronizzazione o di comunicazione con altri processi). Quale/quali, tra gli algoritmi di scheduling FCFS, SJF preemptive, SJF non-preemptive, round robin garantisce/garantiscono che il processo P riuscirà a portare a termine la propria computazione?

**Risposta(Sketch)**

1. Diagramma di GANT, assumendo come algoritmo di scheduling SJF preemptive:

(0) ... P1 .. (2) ... P2 ... (5) ... P3 ... (8) ... P2 ... (9) ... P4 .. (11) ... P2 .. (14) .. P1 ... (22)

Waiting time medio:

$$P1 = (22 - 0) - 10 = 12;$$

$$P2 = (14 - 2) - 7 = 5;$$

$$P3 = (8 - 5) - 3 = 0;$$

$$P4 = (11 - 9) - 2 = 0;$$

$$\text{waiting time medio} = 17/4$$

2. FCFS, e round robin. Infatti, nel caso di SJF (preemptive e non) potrebbe sempre arrivare in coda di ready un processo che deve usare la CPU per un tempo minore di quanto rimane da eseguire a P.

**Esercizio 6** Cosa e' l'algoritmo di rimpiazzamento della seconda chance? (cioe' come funziona) E' stato pensato come miglioramento di quale o quali algoritmi? Che vantaggi ha rispetto a questo/questi?

**Esercizio 7** Si consideri un sistema in cui il numero massimo per una pagina (cioe' il numero piu' alto che puo' rappresentare un numero di pagina) e' 1111 1111 1111 (in binario). Un indirizzo fisico del sistema e' scritto su 26 bit, e la RAM e' suddivisa in  $2^{15}$  frame.

1. Quanto e' grande un frame?
2. Quanto e' grande, in megabyte, lo spazio di indirizzamento logico del sistema?
3. Quali informazioni conterra' ciascuna entry di una tabella delle pagine di questo sistema, se il sistema usa l'algoritmo di rimpiazzamento della seconda chance?
4. Nel caso non si verifichino mai page fault, qual'e', in nanosecondi, il tempo medio di accesso in RAM del sistema se viene usato un TLB con un tempo di accesso di 5 nanosecondi, un hit-ratio del 95% e un tempo di accesso in RAM di 0,08 microsecondi? (e' sufficiente riportare l'espressione aritmetica che fornisce il risultato finale)

**Risposta(Sketch)**

1. un numero di frame e' scritto su 15 bit, e la dimensione di un frame (e quindi anche di una pagina), e' di  $2^{11}$  byte ( $26 - 15 = 11$ ).
2. Poiche' il numero piu' grande di una pagina e' come indicato, ci possono essere al massimo  $2^{12}$  pagine, e lo spazio di indirizzamento logico e' di  $2^{12} \times 2^{11} = 2^{23}$  byte (pari a circa 8 megabyte).
3. Il numero del frame che contiene la pagina corrispondente, il bit di validita' della pagina, il reference bit.
4.  $T_{medio} = 0,95 * (80+5) + 0,05 * (2*80 + 5)$  nanosecondi

**Esercizio 8** Nei microprocessori moderni, quali forme di parallelismo sono utilizzate per migliorare le prestazioni?

**Esercizio 9** Supponiamo di avere un buffer *Buf* di 4 semafori; quindi *Buf*[*i*] rappresenta il semaforo *i* (*i* = 0...3). Tutti i semafori sono inizializzati a 0. Supponiamo anche di avere 4 processi *P<sub>i</sub>* (*i* = 0...3) ognuno dei quali esegue questo codice

```
P(Buf[i])
P(Buf[i+1 mod 4])
print <something>
V(Buf[i])
V(Buf[i+1 mod 4])
```

Facciamo partire tutti i processi contemporaneamente. Si puo' produrre deadlock? motivare la risposta.

**Risposta(Sketch)** Qui c'e' un deadlock immediato perche' ogni processo cerca di acquisire un semaforo inizializzato a 0. (Piu' interessante il caso in cui i semafori sono tutti inizializzati a 1: anche in questo ci puo' essere deadlock quando ogni processi *i* acquisisce il semaforo *i* (quindi esegue la prima istruzione); allora la seconda istruzione, che chiede di acquisire il semaforo *i + 1 mod 4* sara' bloccante, perche' a questo punto il valore di tutti i semafori sara' 0)

**Esercizio 10** 1. Quale tra crittografia a chiave simmetrica e a chiave pubblica e' piu' sicura? Perche'?

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

3. Usare le proprieta' dell'aritmetica modulo per calcolare in modo semplice  $15^8 \bmod 11$

**Risposta(Sketch)**

1. la sicurezza dipende dall lunghezza delle chiavi, in entrambe. quindi non si puo' dire che una sia piu' sicura dell'altra.
2. la simmetrica, decisamente (calcoli piu' semplici per una macchina)
3.  $15^8 \bmod 11$   
 $= 4^8 \bmod 11$   
 Ora abbiamo  $4^2 \bmod 11 = 5$  e quindi  
 $4^4 \bmod 11 = 5^2 \bmod 11 = 3$   
 Quindi  $4^8 \bmod 11 = 3^2 \bmod 11 = 9$

**Ricordarsi di mettere il proprio nome, cognome, e numero di matricola in tutti i fogli.**

**Motivare, brevemente ma chiaramente, tutte le risposte**

**Per avere la sufficienza, e' **\*\*necessario\*\*** svolgere ognuno dei primi 3 esercizi.**

**Esercizio 1** Considerate i 3 processi seguenti:

P1	P2	P3
print(A)	print(B)	print(C)
print (OK)	print (OK)	print (OK)

Si vuole aggiungere operazioni di semaforo in modo che la stringa finale stampata sia C B A OK OK OK, e che il valore di eventuali semafori alla fine della esecuzione del sistema siano gli stessi che i semafori avevano al momento della inizializzazione. Ricordarsi che, come sempre, e' preferibile massimizzare la concorrenza nel sistema.

**Risposta(Sketch)**

P1	P2	P3
T.P	S.P	
print(A)	print(B)	print(C)
U.V U.V	T.V U.P	S.V U.P
print (OK)	print (OK)	print (OK)

Tutti semafori a 0

**Esercizio 2** Disegnare il grafo che rappresenta gli stati possibili di un processo, e le transizioni tra stati. Per ogni transizione, indicare almeno una ragione che puo' causare quella transizione.

**Risposta(Sketch)** Vedi appunti di corso

**Esercizio 3** 1. In un sistema con una una tabella di pagine (che non e' "paginata" su piu' livelli) come e' fatto un indirizzo logico e cosa indicano le sue componenti? E un indirizzo fisico?

**Esercizio 4** Abbiamo 3 processi P1, P2,P3 che eseguono concurrently. Ogni Pi ripetutamente esegue una operazione op-i. Vogliamo che in un qualunque momento una qualunque di queste operazioni op-i sia stata eseguita al massimo una volta in piu' di una qualunque altra operazione op-j.

1. Proporre una soluzione al problema usando semafori, e spiegare perche' funziona. Si tratta di aggiungere operazioni su semafori all'interno del codice dei Pi, e non si possono aggiungere altri processi al sistema.
2. La stessa domanda di sopra, ma in questo caso c'e' il vincolo che all'interno di ogni Pi posso aggiungere al piu' 2 operazioni su semaforo. Posso pero' aggiungere altre componenti al sistema. (Qualora la soluzione al punto precedente sia quella proposta anche per questo punto: indicare la cosa chiaramente)

Come sempre, sono preferibili soluzioni che massimizzano la concorrenza.

**Risposta**(Sketch)

1. Uso 6 semafori binari, inizializzati ad 1 nello schema seguente:

P1

Loop

P(S12)

P(S13)

op1

V(S21)

V(S31)

P2

Loop

P(S21)

P(S23)

op2

V(S12)

V(S32)

P3

Loop

P(S31)

P(S32)

op3

V(S13)

V(S23)

2. Si aggiunge una 4a componente che effettua la sincronizzazione [Idea: uso un coordinator che aspetta un segnale da tutti i processi, e poi invia un segnale a tutti i processi quando possono ripartire] Tutti i semafori sotto sono inizializzati a 0

P1

Loop

P(S1)

op1

V(T1)

P2

Loop

P(S2)

op2

V(T2)

P3

Loop

P(S3)

op3

V(T3)

Coordinator

Loop

V(S1)

V(S2)

V(S3)

P(T1)

P(T2)

P(T3)

**Esercizio 5** Un hard disk ha una dimensione di  $2^{28}$  byte, suddivisi in blocchi da 2Kbyte. Il sistema operativo che usa l'hard disk adotta una allocazione indicizzata dello spazio su disco, e usa 4 byte per scrivere il numero di un blocco. Sul disco e' memorizzato un file A grande 350 Kbyte. Tutti gli attributi del file A sono gia' in RAM. (Tutte le risposte sotto date vanno adeguatamente motivate)

1. Quante operazioni di I/O su disco sono necessarie per leggere il byte numero 190.000 del file?
2. E' noto che, nel file system memorizzato sull'hard disk, la quasi totalita' dei file occupa un solo blocco di dati, e i file restanti occupano due blocchi di dati. In confronto alle altre tecniche di allocazione dello spazio su disco (contigua e concatenata), possiamo dire che viene sprecato tanto o poco spazio sull'hard disk, a causa dell'uso dell'allocazione indicizzata?
3. Sempre nelle ipotesi del punto (2), in confronto alle altre tecniche di allocazione dello spazio su disco (contigua e concatenata), possiamo dire che l'accesso ai dati dei file e' piu' lento o piu' veloce, a causa dell'uso dell'allocazione indicizzata?

**Risposta(Sketch)**

1. In un blocco indice possono essere contenuti  $2048 / 4 = 512$  puntatori a blocco. L'uso di un solo blocco indice e' quindi sufficiente a memorizzare i numeri di tutti i blocchi in cui e' memorizzato A, per cui sono necessarie 2 operazioni di I/O: lettura del blocco indice, lettura del centesimo blocco del file (in cui e' contenuto il byte 204.800).
2. Viene sprecato tanto spazio, in quanto per ogni file viene usato anche un blocco indice, che e' quasi completamente inutilizzato.
3. Piu' lento. Infatti, per leggere i dati di un file occorre sempre prima leggere il suo blocco indice, e quindi sono comunque necessarie due letture su disco. Per le altre tecniche di allocazione invece, nella quasi totalita' dei casi basterebbe una lettura su disco, e negli altri casi ne sarebbero sufficienti due.

**Esercizio 6** Dato un sistema con una RAM di 4 frame, la tabella sotto indica il numero di pagina memorizzata nel frame, il tempo di caricamento, il tempo dell'ultimo accesso alla pagina, e il reference bit.

Page	Load_Time	Last_Reference	Reference_Bit
0	22	32	0
1	34	36	1
2	10	33	1
3	23	38	1

1. Quale pagina sara' sostituita con l'algoritmo di page replacement FIFO? Spiegare brevemente anche il perche'.
2. Stessa domanda per la LRU.
3. Stessa domanda per l'algoritmo della "second chance" (assumendo che l'ordine con cui le pagine vengono esaminate e' dalla piu' vecchia, cioe' presente da piu' tempo, a quella piu' giovane).

**Risposta**(Sketch)

1. 2
2. 0
3. 0

**Esercizio 7** Cosa e' il principio di localita' dei programmi? Date 2 esempi in cui durante il corso questo principio e' stato evocato per giustificare certe scelte in ambito di Sistemi Operativi o di hardware sottostante.

**Esercizio 8** Cos'e' l'algoritmo della "Second Chance"? Brevemente, come funziona? E' un raffinamento, o approssimazione, di quale o quali algoritmi?

**Risposta**(Sketch) Vedi note corso.

**Esercizio 9** Che cosa si intende per 'crittografia a chiave pubblica' e 'crittografia a chiave simmetrica'?

In uno scambio di comunicazioni che richiede firma digitale, quale dei due schemi e' preferibile e perche'?

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

**Esercizio 10** 1. Come funziona l'algoritmo di Shortest Job First?

2. Che svantaggi ha?
3. Come lo si puo' approssimare?

**Risposta**(Sketch) Vedi appunti di corso

**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 + almeno uno degli esercizi sulla concorrenza.**

**Esercizio 1** 1. Cosa e' il context switch?

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

**Risposta**(Sketch) Cambio del processo running. e' tempo di overhead. Vedi appunti corso

**Esercizio 2** Che differenze ci sono tra un semaforo binario ed una variabile booleana?

**Risposta**(Sketch) vedi appunti corso

**Esercizio 3** Considerate i 2 processi sotto, che vengono fatti partire contemporaneamente. Che problema ci sarebbe? Mostrare come il problema e' risolto usando semafori.

P	Q
loop	loop
<non critical section>	<critical section>
<critical section>	<non critical section>

dove i loop sono intendersi dei loop infiniti.

**Risposta**(Sketch) Violazione critical section.

P	Q
loop	loop
	S.P()
<non critical section>	<critical section>
S.P()	S.V()
<critical section>	<non critical section>
S.V()	

con S inizializzato ad 1

**Esercizio 4** Quali sono gli elementi fondamentali della tecnica della memoria virtuale?

**Esercizio 5** Per il CPU scheduling, supponiamo che un certo insieme di processi debba essere trattato con la politica Shortest Job First (SJF) non preemptive. Esiste un assegnamento di priorita' ai processi in modo che la loro esecuzione con la politica a priorita' produca esattamente la stessa sequenza di esecuzione? Se si, come?

**Risposta**(Sketch) priorita' che riflettano lunghezza tempo di burst

**Esercizio 6** 1. Quando si parla di gestione della memoria centrale (RAM), cosa si intende con gestione di tipo "contiguo" (Contiguous Memory Allocation)? Che svantaggi principali presenta?



2. In un sistema del genere, supponiamo che un programma sia caricato in memoria all'indirizzo fisico 30000. Calcolare l'indirizzo fisico (in numero decimale) che corrisponde ai seguenti indirizzi logici: A. 1200; B. 2300; C. 7000

**Risposta**(Sketch) A: 31200; B 32300; C: 37000

**Esercizio 7** Si consideri la seguente tabella di processi e loro CPU burst

P1	120
P2	60
P3	180
P4	50
P5	300

Si faccia un diagramma Gantt che mostra la loro esecuzione rispetto ad una politica Round Robin con quanto di tempo pari a 60

**Risposta**(Sketch)

P1 [60] P2 [120] P3 [180] P4 [230] P5 [290] P1 [350] P3 [410]  
P5 [470] P3 [530] P5 [590] P5 [590] P5 [650] P5 [710]

**Esercizio 8** Consideriamo i seguenti task

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

dove i loop sono intendersi dei loop infiniti, e A, B, C sono da intendersi stampe di queste lettere. Vogliamo che nella stringa di stampa risultante, in ogni momento valga la seguente proprietà: il numero di A stampate è maggiore o uguale al numero di B stampate; il numero di B stampate è maggiore o uguale al numero di C stampate.

Proponete una soluzione usando semafori. Una soluzione sarà tanto migliore quanto più non-determinismo permette.

**Risposta**(Sketch)

P1	P2	P3
loop	loop	loop
	P(T)	P(S)
<A>	<B>	<C>
V(T)	V(S)	

con i semafori S e T inizializzati a 0. (Idea: ogni volta che stampo una A produco un "gettone", tramite semaforo T, che mi permetterà (non necessariamente subito) di stampare una B. Quindi il numero di B stampate sarà sempre inferiore o uguale al numero di A. Stesso ragionamento per B e C con semaforo S

**Esercizio 9** Un sistema con memoria paginata usa un TLB con un hit-ratio del 70%, e un tempo di accesso di 5 nanosecondi. Un accesso in RAM richiede invece 0,03 microsecondi. Qual è, in nanosecondi, il tempo medio di accesso in RAM?

**Risposta**(Sketch)  $T_{\text{medio}} = 0,7 * (30 + 5) + 0,3 * (2 * 30 + 5)$  nanosecondi

**Esercizio 10** 1. Come funziona l'algoritmo RSA?

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

**Risposta**(Sketch) Vedere appunti corso.

**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** (Gestione memoria) Com'e' la struttura di un indirizzo logico? E di un indirizzo fisico? Spiegare bene il significato dei vari campi, e le loro eventuali relazioni.

**Esercizio 2** Cos'e' una system call? A cosa serve?

**Esercizio 3** Cos'e' algoritmo di Round Robin e come funziona?

**Esercizio 4** Il seguente blocco di codice e' eseguito da due thread in parallelo. Inizialmente (prima della esecuzione dei thread) abbiamo  $x = 4$  e  $y = 5$ .

```
x = x + 1;  
y = y - 1;
```

Quali sono i possibili valori di  $x$  e  $y$  quando i thread terminano?

**Risposta**(Sketch) In assenza di interferenze i valori possibili sono  $x=6$  e  $y=3$ . Ma essendo  $x,y$  condivise ci possono anche essere interferenze, quindi con la possibilita' che ci siano anche i valori  $x = 5$  e  $y = 4$ .

**Esercizio 5** Considerate i seguenti processi, lanciati in parallelo:

P1	P2
print(A)	print(E)
print(B)	print(F)

Vogliamo che le stringhe possibili stampate siano tutte le stringhe della forma  $\{AE\}\{BF\}$ , dove la notazione  $\{XY\}$  indica che  $X$  e  $Y$  possono essere in un qualunque ordine. (Nota: che non ci sono loop.) Proponete una soluzione usando semafori.

**Risposta**(Sketch)

P1	P2
print(A)	print(E)
S.V; T.P;	T.V; S.P;
print(B)	print(F)

**Esercizio 6** Un hard disk ha la capacità di 64 Gigabyte, ed è formattato in blocchi da 1 Kilobyte, e usa la allocazione indicizzata per memorizzare i file su disco. Sull'hard disk è memorizzato un file F grande  $1/3$  (un terzo) di Megabyte.

Quante operazioni di I/O su disco sono necessarie per portare in RAM l'ultimo blocco del file F (come al solito, si assuma che inizialmente sia presente in RAM solo la copia del file directory che contiene F)

**Risposta**(Sketch) L'hard disk contiene  $2^{36}$  Byte, e permette di avere  $2^{36}/2^{10} = 2^{26}$  blocchi. Sono quindi necessari 4 byte per scrivere in numero di un blocco. Quindi un blocco indice può contenere al massimo  $2^{10}/2^2 = 2^8 = 256$  puntatori a blocco.

Il file F ha una grandezza compresa tra  $1/4$  e  $1/2$  MB, quindi tra  $2^{18}$  e  $2^{19}$  Byte. Siccome  $2^{19}/2^{10} = 2^9 = 512$  e  $2^{18}/2^{10} = 2^8 = 256$ , il file quindi necessita di un numero di blocchi compreso tra 256 e 512.

Sono quindi necessari 2 blocchi indice per tenere traccia di tutti i blocchi del file. Assumendo che tali blocchi indici siano concatenati, e sapendo che è già in RAM il numero del primo blocco indice, sono necessarie 3 operazioni di I/O: lettura dei due blocchi indice più lettura del blocco del file.

**Esercizio 7** 1. Descrivere come funziona l'allocazione indicizzata (ci si può aiutare anche con un disegno).

2. In quale caso l'allocazione indicizzata comporta un grande spreco di spazio su disco?

**Risposta**(Sketch) Per (2): Nel caso di file molto piccoli, perché il blocco indice rimane quasi completamente inutilizzato.

**Esercizio 8** 1. Cosa è un algoritmo di "frame allocation" ?

2. Indicare 2 possibili modi di effettuare il frame allocation.

3. Qual'è la differenza tra allocation di tipo "local" e di tipo "global" ?

**Esercizio 9** Cos'è la "Translation Lookaside Buffer" (TLB)? A cosa serve?

**Esercizio 10** Quali sono i principali vantaggi e svantaggi della crittografia a chiave simmetrica e rispetto a quella a chiave pubblica?

**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** Qual'e' l'idea della paginazione? Perche' e' preferibile una gestione della memoria con "paginazione" rispetto alla gestione "contigua"? (In altre parole, perche' e' stata introdotta la paginazione?)

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

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

**Risposta(Sketch)** Vedi note di corso

**Esercizio 3** 1. Che differenza c'e' tra 'thread' e 'processo'? E che vantaggi ha il 'thread'?

2. E' possibile nei sistemi operativi moderni, avere sia processi che thread, o sono due concetti incompatibili?

**Risposta(Sketch)** Vedere appunti

**Esercizio 4** In una stringa, usiamo la notazione {AB} per dire che in quel punto la stringa puo' avere sia AB che BA.

Considerare questi 2 processi, lanciati in parallelo:

P1	P2
loop forever	loop forever
print<A>	print<B>

Volgiamo che le stringhe stampate siano {AB}{AB}....

1. Proponete una soluzione usando semafori, con la condizione che tutti i semafori usati abbiano un valore di inizializzazione 0. Come al solito, una soluzione con non-determinismo, che ammette piu' stringhe stampate, e' preferibile ad una deterministica.
2. Stessa cosa del punto precedente, ma ora imponendo che tutti i semafori abbiano valore di inizializzazione 1

**Risposta(Sketch)**

P1	P2
loop forever	loop forever
print<A>	print<B>
S.V	T.V
T.P	S.P

con S,T inizialmente = 0.

Per avere inizializzazione a 1 sufficiente spostare le P su semafori all'inizio

**Esercizio 5** All'interno di un sistema operativo, un certo processo P e' correntemente in stato di "Running" e si sa che non dovra' piu' rilasciare la CPU volontariamente prima di aver terminato la propria esecuzione (in altre parole, non deve piu' eseguire operazioni di I/O, di sincronizzazione o di comunicazione con altri processi).

1. Quale/quali, tra gli algoritmi di scheduling FCFS, SJF preemptive, SJF non-preemptive, round robin garantisce/garantiscono che il processo P riuscirà a portare a termine la propria computazione senza mai perdere il controllo del processore (cioe' rimanendo sempre in stato running)?
2. Quale/quali, tra gli algoritmi di scheduling FCFS, SJF preemptive, SJF non-preemptive, round robin garantisce/garantiscono che il processo P riuscirà a portare a termine la propria computazione?
3. In quale caso un processo potrebbe uscire volontariamente dallo stato di Ready? (cioe': un processo che e' in stato Ready, in quale altro stato puo' andare?)

**Risposta**(Sketch)

1. FCFS, SJF non-preemptive.
2. FCFS, SJF non-preemptive e round robin. Infatti, nel caso di SJF preemptive potrebbe sempre arrivare in coda di ready un processo che deve usare la CPU per un tempo minore di quanto rimane da eseguire a P.
3. mai (cioe' un processo Running puo' andare solo in stato Running).

**Esercizio 6** Riportate un semplice esempio in pseudo-codice (simile a quello usato per la prima parte di questa domanda) di due processi concorrenti che usano uno o piu' semafori per sincronizzarsi e che, a seconda dell'ordine relativo in cui vengono eseguite le istruzioni dei due processi, puo' sia funzionare correttamente che portare in una situazione di deadlock. Indicate anche come devono essere inizializzati i semafori che usate.

**Risposta**(Sketch)

```

P1
wait(mutex1)
wait(mutex2)
sez. critica
signal(mutex2)
signal(mutex1)

```

```

P2
wait(mutex2)
wait(mutex1)
sez. critica
signal(mutex1)
signal(mutex2)

semaphore mutex1 = 1; semaphore mutex2 = 1;

```

**Esercizio 7** Supponiamo che le pagine in uno spazio di indirizzi virtuali siano usate secondo l'ordine seguente (quella sotto e' cioe' la "reference stream")

1 2 1 3 2 1 4 3 1 4

Supponiamo anche ci siano 3 frame disponibili, inizialmente liberi. Assumiamo anche che le decisioni sulla paginazione siano fatte su domanda, cioe' quando un page fault ha luogo. Mostrare i contenuti dei frame dopo ogni accesso a memoria, secondo la politica LRU. Quanti page fault occorrono?

**Risposta**(Sketch)

Frame	Reference										
	1	2	1	3	2	1	4	3	1	4	
0	1	1	1	1	1	1	1	1	1	1	
1		2	2	2	2	2	2	3	3	3	
2				3	3	3	4	4	4	4	
Fault?	Y	Y	N	Y	N	N	Y	Y	N	N	

Numero totale faults = 5.

**Esercizio 8** In un sistema la memoria fisica e' divisa in  $2^{20}$  frame, un indirizzo logico e' scritto su 31 bit, e una pagina e' grande 512 byte.

Quanto e' grande un frame? Quanti frame occupa la page table piu' grande del sistema?

**Risposta**(Sketch) Un frame e' grande come una pagina, quindi  $2^9 = 512$  byte. Quindi la page table piu' grande puo' avere  $2^{(31-9)} = 2^{22}$  entry. Nel sistema vi sono  $2^{20}$  frame, per cui sono necessari tre byte per scrivere il numero di un frame, e quindi la page table piu' grande occupa  $(2^{22} \times 3)/2^9$  frame =  $2^{13} \times 3$  frame = 24K frame

**Esercizio 9** Riportare la distinzione tra crittografia a chiave simmetrica e a chiave pubblica. Quali sono i principali vantaggi della prima? E quelli della seconda?

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

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

**Risposta**(Sketch) appunti corso

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

**Risposta**(Sketch) appunti corso

**Esercizio 3** Perche' puo' essere vantaggioso usare thread al posto di processi ?

**Risposta**(Sketch) appunti corso

**Esercizio 4** 1. Cosa e' un semaforo? Perche' e' importante nei Sistemi Operativi?

2. Riportate lo pseudocodice che descrive il funzionamento delle operazioni su semaforo. Preoccupatevi di mettere in evidenza eventuali particolarita' richieste nell'implementazione di tale pseudocodice (rispetto ad una esecuzione dello pseudocodice come un comune programma).

**Risposta**(Sketch) appunti corso

**Esercizio 5** Un hard disk ha la capienza di  $2^{38}$  byte, ed e' formattato in blocchi da 2048 byte.

Quanti accessi al disco sono necessari per leggere l'ultimo blocco di un file A della dimensione di 8192 byte, assumendo che sia gia' in RAM il numero del primo blocco del file stesso e che venga adottata una allocazione concatenata dello spazio su disco?

**Risposta**(Sketch)

Ogni blocco infatti memorizza 2044 byte di dati piu' 4 byte di puntatore al blocco successivo (infatti,  $2^{38}/2^{11} = 2^{27}$ ), per cui sono necessari 5 blocchi per memorizzare l'intero file.

**Esercizio 6** Un sistema con memoria paginata usa un TLB con un hit-ratio del 92%, e un tempo di accesso di 5 nanosecondi. Un accesso in RAM richiede invece 0,08 microsecondi. Qual e', in nanosecondi, il tempo medio di accesso in RAM (esplicitate i calcoli che fate)?

**Risposta**(Sketch)  $T_{medio} = 0,92 * (80 + 5) + 0,08 * (2 * 80 + 5)$  nanosecondi

**Esercizio 7** Si consideri un sistema in cui l'indice piu' grande usabile nella tabella delle pagine di un processo e' 111 1111 1111 1111 (scritto in binario). Un indirizzo fisico del sistema e' scritto su 25 bit, e la RAM e' suddivisa in  $2^{14}$  frame.

Quanto e' grande lo spazio di indirizzamento logico del sistema? E quanti megabyte sono circa?



**Risposta**(Sketch) Un numero di frame e' scritto su 14 bit, e la dimensione di un frame, e quindi di una pagina, e' di  $2^{11}$  byte ( $25 - 14 = 11$ ). Poiche' il numero piu' grande di una pagina e' 111 1111 1111 1111, ci possono essere al massimo  $2^{15}$  pagine, e lo spazio di indirizzamento logico e' di  $2^{15} \times 2^{11}$  byte (pari a circa 64 megabyte).

**Esercizio 8** Considerate i seguenti processi, lanciati in parallelo:

P1	P2
loop forever	loop forever
{print(A)	{print(E)
print(B)}	print(F)}

Vogliamo che le stringhe possibili stampate siano tutte le stringhe della forma

$$\{AE\}F\{BE\}F\{AE\}F\{BE\}$$

dove la notazione  $\{XY\}$  indica che  $X$  e  $Y$  possono essere in un qualunque ordine. Proponete una soluzione usando semafori.

**Risposta**(Sketch)

P1	P2
loop forever	loop forever
print(A)	print(E)
S.V; T.P;	S.P;
print(B)	print(F)
S.V; T.P;	T.V

con semafori S,T inizializzati a 0

**Esercizio 9** 1. In un file system, cos'e' un link? Qual'e' la differenza tra symbolic link e hard link?

2. Che differenza c'e' tra indirizzo assoluto e relativo? Fate un esempio.

**Esercizio 10** 1. In protocolli di sicurezza, descrivere il significato dei termini: confidentiality; message integrity; end-point authentication.

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

Prendete il vostro numero di matricola. Eliminate tutti gli “0”, e chiamate  $N$  il numero risultante, e  $m$  la sua lunghezza. Esempio se il numero di matricola e’ 0000800470, allora  $N = 847$  e  $m = 3$

Nel seguito  $N_i$  e’ la cifra di  $N$  in posizione  $i \bmod m$ . Cioe’: si contano le cifre di  $N$  da sinistra verso destra, la prima cifra e’  $N_0$ , poi  $N_1$  e cosi’ di seguito, modulo la lunghezza di  $N$ . Nell’esempio sopra:

- $N_0 = 8$
- $N_1 = 4$
- $N_2 = 7$
- $N_3 = 8$
- $N_4 = 4$

Inoltre allo stesso modo  $N_2N_3$  e’ il numero 78 (e **non la moltiplicazione**  $7 \times 8$ ).

Negli esercizi sotto: non eseguite calcoli o ragionamenti astratti in termini di questi  $N_i$ , ma sostituite subito ai vari  $N_i$  il valore concreto dato dal vostro numero matricola.

---

**Ad inizio foglio indicate chiaramente il vostro numeri di matricola e quali sono nel vostro caso i numeri  $N_0, N_1, N_2, N_3$ . Inoltre all’inizio di ogni esercizio in cui intervengono tali numeri, indicatene il valore**

Motivare le risposte date. Non e’ necessario dare risposte 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).

**Ricordare che lo scan finale deve contenere il libretto universitario di riconoscimento; e che il compito va inviato in pdf e in un unico file, come allegato**

---

**E’ necessario svolgere gli esercizi 1 , 2, 3**

---

Per coloro che svolgono la prova da casa: in considerazione del minor tempo disponibile, sono richiesti solamente esercizi 1–6 e inoltre: nell’esercizio 3 solo le domande 2 e 3; nell’esercizio 4, solo le prime 2 domande. (Ovviamente tutto il resto e’ comunque facoltativo.)

**Esercizio 1** Considerate i seguenti processi:

P1	P2
<code>print(<math>N_1</math>)</code>	<code>print(<math>N_2</math>)</code>
<code>print(<math>N_3</math>)</code>	<code>print(<math>N_4</math>)</code>

Agite, se necessario, sul codice, inserendo opportune operazioni su semaforo, in modo che l'output di questa esecuzione concorrente sia una stringa in cui vengono stampati  $N_1$  ed  $N_2$ , *in ordine decrescente*, e successivamente  $N_3$  ed  $N_4$ , sempre *in ordine decrescente*. Esempio: se  $N_1 = 5$ ,  $N_2 = 8$ ,  $N_3 = 6$ ,  $N_4 = 3$  allora la stringa da stampare e' 8 5 6 3.

**Risposta**(Sketch) Assumo  $N_1 = 5$ ,  $N_2 = 8$ ,  $N_3 = 6$ ,  $N_4 = 3$  Quindi occorre stampare la stringa 8563.

Semafori  $S, T$  inizializzati a 0

P1	P2
<code>S.P()</code>	
<code>print 5</code>	<code>print 8</code>
	<code>S.V()</code>
	<code>T.P()</code>
<code>print 6</code>	<code>print 3</code>
<code>T.V()</code>	

**Esercizio 2** Quali sono i vantaggi della paginazione, rispetto ad una allocazione della memoria centrale di tipo contiguo? Ci sono anche degli svantaggi?

**Esercizio 3** Sia:  $N$  il max tra  $N_2$  e 6; ed  $M$  il max tra  $N_1$  e 7.

Quattro processi arrivano al tempo indicato e consumano la quantita' di CPU indicata nella tabella sottostante:

Processo	T. di arrivo	Burst time
A	0	$N_2 + 12$
B	3	$N_3 + 5$
C	6	$M$
D	9	$N$

1. Riportate il diagramma di Gantt che mostra cio' che succedera' in esecuzione, adottando uno scheduling di tipo Round Robin con quanto di tempo 4.
2. Lo stesso, in questo caso assumendo Shortest Job First di tipo preemptive.
3. Calcolare il waiting time di B nel 2o caso.

**Risposta**(Sketch)

Assumiamo  $N_1 = 5$ ,  $N_2 = 4$ ,  $N_3 = 6$  quindi  $N = 6$ ,  $M = 7$ .

Processo	T. di arrivo	Burst time
A	0	20
B	3	11
C	6	7
D	9	6

[0]A[4]B[8]A[12]C[16]B[20]D[24]A[28]C[31]B[34]D[36]A[40]A[44]

Evoluzione processi nella ready-queue :

0 A  
 4 BA  
 8 ACB  
 12 CBDA  
 16 BDAC  
 20 DACB  
 24 ACBD  
 28 CBDA  
 31 BDA  
 34 DA  
 36 A  
 40 A

SJF:

[0]A[3] B [6] C[13] D[19] B [27] A[44]

WT (B) = 24 (27-3)

**Esercizio 4** Si consideri un sistema in cui in una tabella delle pagine di un processo puo' avere  $2^{N_1} \times 1024$  entry. Un indirizzo fisico del sistema e' scritto su  $N_2 + 20$  bit, e la RAM e' suddivisa in  $2^{N_2} \times 1024$  frame.

1. Quanto e' grande, in byte, lo spazio di indirizzamento logico del sistema?
2. Quanta e' grande, in byte, al massimo, una tabella di pagine? (assumete esistenza del bit di validita' e del dirty bit, e nessun altro bit aggiuntivo)
3. Supponiamo ora che sia  $N_1 = 3$ , ed  $N_2 = 4$ . In questo caso e' necessaria una paginazione a 2 livelli?
4. Sempre relativamente al caso  $N_1 = 3$ , ed  $N_2 = 4$  della domanda sopra:  
 Se la paginazione a 2 livelli e' necessaria, quanti byte ha, al piu', la tabella di pagine esterna? (va bene anche se indicate semplicemente il numero di entry massimale di questa tabella)
5. In quale caso il sistema descritto qui sopra deve implementare la memoria virtuale?

**Risposta**(Sketch) Nelle risposte sotto, prendo  $N_1 = 3, N_2 = 4$  Quindi: tabella delle pagine di un processo  $2^3 \times 1024 = 2^{13}$  entry, indirizzo fisico 24 bit, RAM e' suddivisa in  $2^4 \times 1024 = 2^{14}$  frame.

1. Un numero di frame e' scritto su 14 bit, e la dimensione di un frame, e quindi di una pagina, e' di  $2^{10}$  B ( $24 - 14 = 10$ ). Poiche' le pagine sono  $2^{13}$ , lo spazio di indirizzamento logico e' di  $2^{13} \times 2^{10}$  B (pari a circa 8 megabyte).
2. La tabella delle pagine piu' grande del sistema ha  $2^{13}$  pagine, e ogni entry contiene il numero di un frame, per cui sono necessari due byte per ogni entry. La dimensione di quella tabella sara' quindi di  $2^{13} \times 2 = 2^{14}$  byte

- dalla risposta sopra, si evince che la tabella di pagine massimale e' piu' grande della dimensione di un frame, il che implica appunto la necessita' di paginare la tabella delle pagine.
- $2^{14}$  B = 16 Kbyte, e quindi questa tabella delle pagine interna occupa 16 frame. Di conseguenza la tabella esterna sara' composta da 16 entry di due byte ciascuno, ossia 32 byte.
- Lo spazio di indirizzamento logico e' minore di quello fisico, per cui l'unico caso e' quello in cui si vogliano avere in esecuzione contemporaneamente processi che, insieme, occupano uno spazio maggiore dello spazio di indirizzamento fisico.

**Esercizio 5** Considerate i seguenti processi

P1

P2

$$x = x + N_2$$

$$x = x + N_3$$

Supponiamo che  $x$  sia inizializzata cosi' :  $x = N_1$ . Al termine della esecuzione concorrente dei 2 processi, quali sono i valori possibili per  $x$ ? Ci si aspetta normalmente un uso di semafori nel codice sopra (motivare la risposta) ?

**Esercizio 6** 1. Descrivere i passi che portano alla determinazione della chiave pubblica e privata nell'algoritmo di RSA.

2. Siano:

- $N = N_2$  se  $N_2$  diverso da 1;  $N = 6$  altrimenti;
- $M = N_1$  se  $N_1$  diverso da 1;  $M = 7$  altrimenti

Usare le proprieta' dell'aritmetica modulo per calcolare in modo semplice

$$(N^7 \times M^{N_3+3}) \bmod 11$$

**Risposta**(Sketch)

1. note corso

2. soluzione

per  $N_2 = 6, M = 7, N_3 = 3$ .

$$(6^7 \times 7^6) \bmod 11$$

Abbiamo  $6^2 \bmod 11 = 36 \bmod 11 = 3$

Quindi anche  $6^4 \bmod 11 = 3^2 \bmod 11 = 9$ .

Siccome  $7 = 4+2+1$ ,  $6^7 \bmod 11 = (3 \times 9 \times 6) \bmod 11 = (27 \bmod 11 \times 6) \bmod 11 = (5 \times 6) \bmod 11 = 30 \bmod 11 = 8$ .

**Esercizio 7** 1. Cos'e' una memoria cache e perche' si usa?

2. Date degli esempi di strutture del Sistema Operativo per gestire le quali si fa ricorso ad una qualche forma di memoria cache.

**Esercizio 8** Quali sono gli stati possibili di un processo? E quali le transizioni possibili tra questi stati? Per ogni transizione indicare una causa possibile della transizione stessa.

**Esercizio 9** Un sistema (hardware + sistema operativo) soffre spesso del problema del thrashing. Indicate due modifiche al sistema, una hardware e una software, che potrebbero migliorare la situazione.