



**Tempo totale a disposizione: 60 minuti.**

## QUESITI & ESERCIZI (max 26 punti)

**max 2 punti per ognuno dei Q&E; max 4 punti per i Q&E num. 11, 12 e 13**

### RACCOMANDAZIONI

- ➡ curare la **comprensione** dei quesiti/esercizi e l'**appropriatezza** del **linguaggio** (🙄) e della **grafia** (😬) adoperati;
- ➡ formulare **risposte puntuali, concise e comprensibili**, evitando inutili divagazioni;
- ➡ **giustificare il perché** delle asserzioni formulate;
- ➡ non trascurare di **dare risposta ad eventuali richieste multiple** contenute nei quesiti/esercizi;

- 1) Che differenza esiste tra il comando **cat** e il comando **tail**?
- 2) Quale sarà l'effetto dell'esecuzione della seguente successione di comandi?  

```
> ln -s matilda link1
> ln -s link1 link2
> cat link2
```
- 3) Si usino almeno due modalità per assegnare contemporaneamente il permesso di scrittura ed esecuzione sul file **text.txt** a qualsiasi utente.
- 4) Creare un alias per il seguente comando:  

```
ps wu -U userA
```

 Si scriva quindi un comando che consenta di verificare la corretta creazione dell'alias.
- 5) Si supponga di essere nella directory di lavoro corrente, l'output del comando **ls -li** è il seguente:
 

```
418384 -rw-rw-rw- 3 user1 groupA 322
giu 11 11:16 paperino.txt
413568 lrwxrwxrwx Y user1 groupA 547
lug 15 13:20 pluto.txt → pippo.txt
302457 -rw-rw-rw- X user1 groupA 321
giu 10 12:15 pippo.txt
```

 Spiegare se e come è possibile stabilire quanti sono i link **fisici** e i link **simbolici** al file **pippo.txt**? Se sì, dire quanti sono.
- 6) Si scriva il comando per trovare i file – presenti nella directory corrente dell'utente – il cui nome inizia per vocale e termina con una cifra.
- 7) In un sistema a memoria virtuale, la relazione tra spazio logico e spazio fisico di indirizzamento è tale che:
  - (a) lo spazio fisico è sempre maggiore dello spazio logico
  - (b) lo spazio fisico è sempre minore dello spazio logico
  - (c) lo spazio fisico di indirizzamento può essere indifferentemente maggiore o minore dello spazio logico
- 8) Qual è il significato di predicibilità di un sistema in tempo reale? E quale la differenza fra predicibilità deterministica, probabilistica e deterministica a run-time?
- 9) Quali sono i costituenti di un *dominio di protezione* e a chi può essere associato un dominio di protezione?
- 10) Si faccia riferimento ad un file system con concatenazione logica dei blocchi. Si assuma che il disco abbia 5Y0 cilindri, X0 tracce per cilindro e 50 blocchi per traccia. A quale blocco logico corrisponderà il blocco fisico con coordinate cyl = 130, trk = 15, blk = 40?
- 11) In un sistema operativo time sharing senza priorità, un processo in esecuzione può essere interrotto dallo scheduler e portato nello stato di *ready* prima dello scadere del quanto di tempo? Specificare la(e) risposta(e) corretta(e) fra le seguenti:
  - (a) no, perché altrimenti il processo verrebbe rallentato e potrebbe essere soggetto a starvation
  - (b) no, perché lo scopo di un sistema time sharing, in assenza di priorità, è quello di eseguire tutti i processi con gli stessi diritti (fair scheduling)
  - (c) no, perché in un sistema time-sharing senza priorità lo scheduler non interrompe mai un processo in esecuzione, ma lascia che il processo si sospenda da solo
  - (d) sì, perché altrimenti gli altri programmi non potrebbero continuare la loro esecuzione
  - (e) sì, se il processo esegue sempre le stesse istruzioni, perché potrebbe essere in un loop infinito

- 12) Si faccia riferimento ad un file system indicizzato con indici a due livelli, dimensione del blocco logico pari a 512 byte e indirizzi di 4 byte. Si supponga inoltre di trascurare la presenza degli indirizzi diretti presenti nell'*index block* dell'*i-node*.

Come si alloca un file di 1 Mb? Quanti blocchi di dati servono? Come si accede al suo 400° blocco? Con questa indicizzazione, qual è il numero massimo di blocchi di dati di un file?

- 13) In un sistema time-sharing con politica di scheduling *round-robin* sono presenti quattro processi nei seguenti stati: P1 in esecuzione, P2 e P3 *ready* (P2 in testa alla coda davanti a P3), P4 in *wait* per una operazione di I/O.

Descrivere come cambia lo stato del sistema (cioè come cambiano di stato i processi) se a partire dalla situazione data si verificano nell'ordine tutti e soli i seguenti eventi:

- a) trascorre un *time slice*

P1->                  P2->                  P3->                  P4->

- b) termina l'operazione di I/O del processo in attesa

P1->                  P2->                  P3->                  P4->

- c) il processo in esecuzione chiede una operazione di I/O

P1->                  P2->                  P3->                  P4->

- d) trascorre un *time slice*

P1->                  P2->                  P3->                  P4->

- e) termina l'operazione di I/O per il processo in attesa

P1->                  P2->                  P3->                  P4->

- f) il processo in esecuzione termina

P1->                  P2->                  P3->                  P4->

- g) trascorre un *time slice*

P1->                  P2->                  P3->                  P4->

- 14) Nei sistemi operativi Unix-like, la chiamata di sistema `fork()` ha il(i) seguente(i) effetto(i):

(a) sostituisce al processo esistente un nuovo processo, identico in tutto e per tutto al processo chiamante, tranne che nel Process Control Block (PCB);

(b) sostituisce al processo esistente un nuovo processo, costruito sulla base del file eseguibile passato come argomento alla chiamata, mantenendo il PCB del processo originale;

(c) divide il processo esistente in due processi che condividono il PCB: il primo identico al processo originale, il secondo costruito a partire dall'eseguibile passato come argomento alla chiamata;

(d) genera un nuovo processo, identico in tutto e per tutto al processo chiamante, tranne che nel Process Control Block (PCB);

(e) genera un nuovo processo, lanciando una nuova istanza del processo chiamante a partire dal file eseguibile corrispondente;

(f) è l'unico meccanismo disponibile con cui un processo pu generare un altro processo;

(g) è il meccanismo più frequentemente usato dai processi per generare nuovi processi.

- 15) Si consideri un sistema costituito da quattro risorse dello stesso tipo condivise da tre processi, ciascuno dei quali ha bisogno al più di due risorse. Dimostrare che il sistema è *deadlock free*.

- 16) Dopo l'esecuzione del seguente frammento di programma eseguito dal processo P:

```
i = fork();
```

```
j = fork();
```

quanti sono i processi in esecuzione?

### AFFERMAZIONI (max 4 punti)

Si considerino le seguenti affermazioni.

Si barri la casella "Sicuramente Vera" (SV), se si è sicuri che l'affermazione è vera.

Si barri, invece, la casella "Sicuramente Falsa" (SF), se si è sicuri che l'affermazione è falsa.

**Per ogni risposta corretta 1 punto. Per ogni risposta errata -1 punto. Le affermazioni senza risposta comportano 0 punti.**

	Affermazione
1.	Un <i>interrupt vector</i> è un puntatore all'informazione necessaria per gestire un interrupt.
2.	L'interfaccia SCSI è un esempio di controllore di dispositivi.
3.	Il DMA non può essere usato con un device driver.
4.	Una <i>cache</i> è un tipo di memoria non volatile.
5.	Il <i>mode bit</i> per gestire la <i>dual-mode operation</i> è stato sostituito dal <i>mode byte</i> in molti OS come Windows XP.
6.	Il <i>message passing model</i> è un protocollo di rete ma non si applica ai sistemi operativi.
7.	Per garantire la non volatilità, molti dispositivi a dischi usano memorie flash.



- CdL Ing. El. & Tlc (DM 270) - Fondamenti dei Sistemi Operativi (6 CFU)* ☐  
*CdL Magistrale Ing. Tlc (DM 270) - Fondamenti dei Sistemi Operativi (6 CFU)* ☐  
*CdL Ing. Tlc (DM 509) - Sistemi Operativi (6 CFU)* ☐  
*CdL Ing. El. & Tlc (DM 270) - Sistemi Operativi (9 oppure 12 CFU)* ☐

Cognome: \_\_\_\_\_ ; Nome: \_\_\_\_\_ ; matricola: \_\_\_\_\_

### Quesiti ed Esercizi

max 2 punti per ognuno dei Q&E; max 4 punti per i Q&E num. 11, 12 e 13

*Dovunque appaiano, utilizzare i seguenti valori delle variabili indicate negli esercizi.*

X = (numero di lettere che compongono il Cognome) - 2.

Y = (numero di lettere che compongono il 1° Nome) - 2.

Z = 1 se X è pari; 0 se X è dispari ;

W = 1 se Y è pari ; 0 se Y è dispari ;

S = (penultima cifra del numero di Matricola).

T = (ultima cifra del numero di Matricola).

X = ..... (max 9);

Y = ..... (max 9);

Z = ..... ;

W = ..... ;

S = ..... ;

T = ..... ;

1) Che differenza esiste tra il comando .....

7) In un sistema a memoria virtuale, la .....

(a)

(b)

(c)

2) Quale sarà l'effetto dell'esecuzione .....

8) Qual è il significato di .....

3) Si usino almeno due modalità ....

9) Quali sono i costituenti di un .....

4) Creare un alias per il seguente ....

5) Si supponga di essere nella directory .....

10) Si faccia riferimento ad un file system .....

6) In riferimento ad una struttura .....

11) In un sistema operativo time sharing .....

(a)

(b)

(c)

(d)

(e)

12) Si faccia riferimento ad un file system . . . . .

14) Nei sistemi operativi Unix-like, . . . . .

- (a)
- (b)
- (c)
- (d)
- (e)
- (f)
- (g)

13) In un sistema time-sharing con . . . . .

15) Si consideri un sistema costituito . . . . .

a) trascorre un *time slice*

P1->                  P2->                  P3->                  P4->

b) termina l'operazione di I/O del processo in attesa

P1->                  P2->                  P3->                  P4->

c) il processo in esecuzione chiede una operazione di I/O

P1->                  P2->                  P3->                  P4->

d) trascorre un *time slice*

P1->                  P2->                  P3->                  P4->

e) termina l'operazione di I/O per il processo in attesa

P1->                  P2->                  P3->                  P4->

f) il processo in esecuzione termina

P1->                  P2->                  P3->                  P4->

g) trascorre un *time slice*

P1->                  P2->                  P3->                  P4->

16) Dopo l'esecuzione del seguente frammento . . . . .

## AFFERMAZIONI

Si considerino le seguenti affermazioni.

Si barri la casella "Sicuramente Vera" (SV), se si è sicuri che l'affermazione è vera.

Si barri, invece, la casella "Sicuramente Falsa" (SF), se si è sicuri che l'affermazione è falsa.

**Per ogni risposta corretta 1 punto. Per ogni risposta errata -1 punto. Le affermazioni senza risposta comportano 0 punti.**

	Affermazione	SV	SF
1.	Un <i>interrupt vector</i> è un puntatore all'informazione necessaria per gestire un interrupt.		
2.	L'interfaccia SCSI è un esempio di controllore di dispositivi.		
3.	Il DMA non può essere usato con un device driver.		
4.	Una <i>cache</i> è un tipo di memoria non volatile.		
5.	Il <i>mode bit</i> per gestire la <i>dual-mode operation</i> è stato sostituito dal <i>mode byte</i> in molti OS come Windows XP.		
6.	Il <i>message passing model</i> è un protocollo di rete ma non si applica ai sistemi operativi.		
7.	Per garantire la non volatilità, molti dispositivi a dischi usano memorie flash.		