

Cognome: _____ ; Nome: _____ ; matricola: _____ ; Ing. _____

Quesiti ed Esercizi**CONSEGNARE SOLO QUESTO FOGLIO**

A&T -> solo per Automazione e Telecomunicazioni

I -> solo per Informatica

Tempo a disposizione: 30 minuti.**Max 18 punti**Dovunque appaiano, utilizzare i seguenti valori delle variabili indicate negli esercizi.

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

X = (max 9);

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

Y = (max 9);

Z = 1 se X è pari; Z = 0 se X è dispari ;

Z = ;

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

W = ;

- 1) Si supponga che la cwd sia /home/utente1. Descrivere l'**albero delle directory** risultante dopo che sono stati lanciati i seguenti comandi:

```
touch file1 file2
mkdir mydir
mkdir mydir/mydir
cd mydir/
cp ../f* mydir/
mv mydir/f?
```

- 2) Indicare il **contenuto del file** output.txt, dopo che sono stati lanciati i seguenti comandi

```
echo "hello"> output.txt
```

```
echo "world"> output.txt
```

```
echo "world">> output.txt
```

- 3) L'utente user1 lanciando il comando jobs ha ottenuto il seguente output:

```
[1]+  Stopped   sleep 100
[2]-  Running   sleep 200 &
```

Indicare il **comando** la cui esecuzione sta continuando in background.

Dire altresì come è possibile **terminare il processo** relativo al comando sleep 100.

- 4) Dato il seguente output:

```
mydir/
dir*
myfile
project/
```

indicare qual è il **comando** che lo ha generato.

Indicare se ci sono **file eseguibili**.

- 5) Con riferimento all'esercizio precedente, indicare se il comando

```
mkdir dir
```

produce un **errore** o meno. Spiegare il motivo.

- 6) La variabile d'ambiente OLDPWD contiene la path della cwd precedente a quella attuale. Per maggiore chiarezza si guardino i comandi sotto-elencati:

```
$>pwd
/home/user1
$>cd mydir
$>pwd
/home/user1/mydir
$>echo $OLDPWD
/home/user1
```

Indicare come creare un nuovo comando old che permetta di **ritornare alla cwd precedente**.

- 7) Indicare il **comando per listare** tutti i file contenuti nella directory /usr/bin che hanno i primi due caratteri maiuscoli e terminano con un numero.

- 8) Spiegare la **differenza** tra i comandi

```
kill -15 1234
kill -9 1234
```

- 9) Dato il file pippo con i seguenti **permessi di accesso**:

```
-rwxr-xrwx 2 user1 groupA
```

Indicare come cambiano tali permessi quando vengono lanciati i seguenti comandi

```
chmod 755 pippo
chmod a-x+w pippo
chmod o-x pippo
```

- 10) In un memory manager a paginazione virtuale, la dimensione della memoria virtuale è di Y Gbyte e il numero di pagina utilizza 1Y bit. Qual è la **dimensione della pagina**?
- 11) **Quanti blocchi di dati** saranno stati allocati per un file da un SO UNIX-like che occupi in totale 2Y00 blocchi?
- 12) **Quale delle proprietà acide** è garantita dalla **serializzabilità** di due o più transazioni? E perché le transazioni devono possedere tale proprietà?
- 13) Si consideri un sistema che si trovi nel seguente stato:
- | | <u>Allocation</u> | <u>Max</u> | <u>Available</u> |
|-------|-------------------|------------|------------------|
| | A B C D | A B C D | A B C D |
| P_1 | 0 0 1 1 | 0 0 1 2 | 1 5 3 1 |
| P_2 | 1 3 3 4 | 2 3 4 6 | |
| P_3 | 1 0 0 0 | 1 X 5 0 | |
| P_4 | 0 0 1 4 | 0 6 Y 6 | |
| P_5 | 0 6 3 2 | 0 6 5 2 | |
- Si specifichi se si tratta di uno **stato sicuro** e perché.
- 14) Ipotizzando un algoritmo di disk scheduling di tipo SSTF, supponendo che le testine, posizionate sul cilindro YX, avanzino verso cilindri crescenti e che si abbia la seguente coda di richieste:
- 92, 156, 26, 102, 74, 184, 55, 37
- si determini la **successione di servizio** delle richieste e si stabilisca il **tempo di seek complessivo** sapendo che il tempo minimo di seek è di 0,1 msec.
- 15) Scrivere la formula generale per il calcolo in *giri al minuto* della **velocità rotazionale** di un disco ed esprimerne il valore nel caso di un tempo medio di latenza pari a 1X msec.
- 16) Indicare il contenuto e la funzione di una **Symbolic File Directory** del modello generale di File System.
- 17) Si consideri la seguente sequenza di richieste di page-in:
- 0 1 2 3 1 0 Z 1 Z 3 W
- Si supponga di disporre di una memoria di 3 blocchi inizialmente disponibili. Si determini il **numero totale di page-fault** ed il contenuto della memoria al termine della sequenza, se l'algoritmo di rimozione delle pagine è il FIFO.
- 18) Da cosa è costituito il **translation look-aside buffer**? E quale il suo vantaggio?
- 19) Qual è lo **svantaggio dell'uso della tecnica del workspace privato** per garantire l'atomicità di una transazione? (I)
- 20) Qual è l'obiettivo delle tecniche denominate **'locking a due fasi'** e **'timestamping'**? E quale è la caratteristica positiva del timestamping? (I)
- 21) Quali sono gli **attributi necessari**, oltre a quelli delle risorse impiegate (Memoria, file, dispositivi), di un processo in tempo reale? (A&T)
- 22) Quando un **task si dice che è periodico** in un sistema in tempo reale? (A&T)

Cognome: _____ ; Nome: _____ ; matricola: _____ ; Ing. _____

Problema

CONSEGNARE SOLO QUESTO FOGLIO

Tempo a disposizione: 60 minuti

Max Flow-chart 6 punti; Max Codice 6 punti

Si progettino, mediante **flow-chart o linguaggio strutturato**, due **procedure** che simulino lo scambio di messaggi tra due o più processi in una mailbox.

In particolare si suppone che alla prima procedura di nome *SEND* vengano forniti:

- la mailbox, dal nome *MAILBOX*, sotto forma di array di strutture di 100 elementi costituiti dai campi *MESSAGE* (stringa di 10 caratteri) contenente il messaggio da scambiare e *ID_PROC* (intero) contenente l'identificativo del destinatario. Se il campo *MESSAGE* vale empty, l'elemento di *MAILBOX* deve intendersi disponibile;
- il messaggio da scambiare *MES* (stringa di 10 caratteri);
- il numero identificativo del processo destinatario *ID_REC* (intero).

Invece alla seconda procedura, di nome *RECEIVE*, vengono forniti:

- la mailbox definita come sopra;
- il numero identificativo del processo *ID_PROC* (intero).

La prima procedura dovrà provvedere a verificare in via preliminare che la mailbox non sia già piena. In caso di esito favorevole, si incaricherà di collocare al suo interno – in corrispondenza del primo elemento disponibile – le relative informazioni da inviare, restituendo la mailbox aggiornata.

La seconda procedura dovrà provvedere ad effettuare in via preliminare che la mailbox non sia vuota. Nel caso ciò non sia vero, si incaricherà di estrarre dalla mailbox tutti i messaggi destinati al processo identificato da *ID_REC*, liberando le corrispondenti locazioni. Le informazioni estratte verranno salvate in un file di struct denominato *OUT*, avente struttura analoga a quella della mailbox.

Si chiede di:

- a) descrivere il **progetto** della procedura suddetta, utilizzando i nomi indicati delle variabili e ricorrendo al **minor numero di istruzioni**;
- b) scrivere, utilizzando il linguaggio C, il **programma rigorosamente corrispondente al flow-chart** descritto.