

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

**ESERCIZI (Max 22 punti)****Tempo a disposizione: 45 minuti****CONSEGNARE SOLO QUESTO FOGLIO**Dovunque appaiano, utilizzare i seguenti valori delle variabili indicate negli esercizi.

X = (numero di lettere che compongono il Cognome) - 2. (max 9)

Y = (numero di lettere che compongono il 1° Nome) - 2. (max 9)

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

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

S = (penultima cifra del numero di Matricola).

T = (ultima cifra del numero di Matricola).

X = ..... ;

Y = ..... ;

W = ..... ;

Z = ..... ;

S = ..... ;

T = ..... ;

1. Dati due file, uno di nome `fileA` costituito da X linee di testo e uno di nome `fileB` costituito da Y linee di testo, indicare l'output del comando

`(cat fileA fileB) | wc -l .`

2. Scrivere una pipe di comandi che consenta di **estrarre da un file di testo** di nome `fileA` le prime X linee e di ordinarle in ordine alfabetico decrescente.

3. Supponendo di aver effettuato il login con USERID: `morelinux`, indicare come, utilizzando un **filtro**, visualizzare le directory contenute all'interno della propria home directory.

4. Si supponga che esista l'utente `user24` con password `SSDDfr` e che l'utente `user23` abbia eseguito il login. Indicare come l'utente `user23` possa **avviare una sessione di lavoro per l'utente user24** senza chiudere la corrente sessione di lavoro. Indicare altresì come l'utente `user24` possa **chiudere la propria sessione di lavoro**.

5. Scrivere il comando per creare un alias di nome `ultime` che estrae le ultime linee di un file a partire dalla X-esima linea.

6. Quanto varrà il **tempo massimo** (espresso in msec) di *roll-out* di un programma di 350 Kb, se il disco interessato ha un *transfer rate* di X Mb/sec, una velocità di Y000 giri/s e si assume che il tempo di posizionamento (*seek*) sia nullo?

7. Un computer prevede indirizzi virtuali di 48 bit e indirizzi fisici di 32 bit. Ogni pagina ha una dimensione di  $4 \cdot (1+S)$  Kbyte. **Quante** potranno essere **in totale le pagine** dei processi in esecuzione?

8. Le seguenti matrici descrivono lo stato corrente di un sistema. Si attualizzi con i propri valori di W e Z la matrice *Allocation* e si determinino gli elementi della matrice *Need*.

	<u>Allocation</u>	<u>Max</u>	<u>Need</u>	<u>Available</u>
	A B C	A B C	A B C	A B C
$P_0$	0 1 W	2 4 3	— — —	2 2 1
$P_1$	2 0 0	9 2 2	— — —	
$P_2$	1 0 W	3 0 2	— — —	
$P_3$	2 0 0	5 1 1	— — —	
$P_4$	2 2 2	2 3 4	— — —	

Specificare se lo stato è **sicuro** o no. **Perché?**

9. Quali sono e cosa asseriscono i principi alla base della politica di **virtualizzazione della memoria**?

10. Si consideri l'algoritmo di attribuzione ai processi di **priorità basate sul merito**. Se la mediana dei meriti attesa è di 0.3 mentre quella determinata al termine di un intervallo di osservazione è di 0.S, come varierà il time slice nel successivo intervallo di osservazione?
11. Si consideri un file system UNIX-like. Si supponga che esso allochi 16 cluster per volta. Da **quanti cluster di indirezione** sarà composto il file dopo aver effettuato XS00 operazioni di scrittura?
12. Si supponga che un programma sia costituito dai seguenti 4 moduli, di cui solo il 1° ed il 3° e poi, separatamente, il 2° ed il 4° debbano essere caricati in memoria contemporaneamente. Quanta sarà la memoria da allocare se il programma viene eseguito adottando la tecnica dell'**overlay** dei moduli?
- Modulo 1: 2S Kb,      Modulo 2: Y0 Kb,  
Modulo 3: X0 Kb,      Modulo 4: 6T Kb
13. Si consideri un **semaforo contatore** che sincronizza l'accesso ad una risorsa costituita da (S+Y+2) esemplari. Se la variabile semaforica ha valore S, quanti esemplari della risorsa sono ancora liberi?
14. Si consideri un File System con allocazione concatenata dei file. Quanti saranno gli accessi a disco richiesti dalla struttura di concatenazione che fa uso della **linked-list** e qual è il suo "inconveniente"?
15. Si consideri il **demand-paging** come tecnica di gestione della memoria. Sapendo che la probabilità di un **page-fault** è del X0%, che il tempo di un **page-in** è Y0 msec e che il tempo di accesso diretto alla memoria è X00 nsec, determinare il tempo medio di accesso ad una pagina di memoria.

Nel seguito vengono riportate affermazioni vere e affermazioni false:

- barra la casella "Sicuramente Vera" (SV), se sei sicuro che l'affermazione è vera;
- barra la casella "Sicuramente Falsa" (SF), se sei sicuro che l'affermazione è falsa;

Per ogni corretta risposta ottieni 1 punto. Per ogni erronea risposta ottieni -1 punto. Le affermazioni senza risposta comportano 0 punti.

Affermazione	SV	SF
Un <b>interrupt</b> è una interruzione interna a carattere asincrono segnalata alla CPU.		
La funzione del <b>bootstrap</b> non è quella di caricare in memoria e avviare il nucleo del SO.		
Lo <b>scheduler della CPU</b> è invocato con frequenza dell'ordine di grandezza del secondo.		
Non è opportuno eseguire un processo il cui <b>address space logico</b> sia più grande dell'address space fisico disponibile.		
Un solo processo non può trovarsi in uno stato di <b>deadlock</b> .		
<b>Deadlock prevention</b> è la stessa cosa che deadlock avoidance.		
Ogni <b>thread</b> non ha un proprio stack.		

Cognome: \_\_\_\_\_ ; Nome: \_\_\_\_\_ ; matricola: \_\_\_\_\_ ; Ing. \_\_\_\_\_

**Problema*****Tempo a disposizione: 60 minuti******Max 8 punti*****CONSEGNARE SOLO QUESTO FOGLIO**

Si progetti, mediante flow-chart o linguaggio strutturato, una procedura che realizzi, per un CPU Scheduler a priorità statica, l'**algoritmo di aggiornamento della coda dei processi che si trovano nello stato Ready**, allorché un processo deve essere eliminato o aggiunto alla coda.

In particolare si assuma che alla procedura siano disponibili:

- il numero **N** dei processi presenti nella coda;
- i due vettori, già ordinati in termini di priorità decrescenti, **PROC\_ID** dei codici di identificazione dei processi e **PRTY** delle rispettive priorità degli **N** processi;
- il tipo di aggiornamento **agg** della coda: **0** per cancellazione o **1** per aggiunta;
- il codice di identificazione **UPD\_PROC\_ID** e la priorità **UPD\_PRTY** del processo interessato.

Al termine della procedura devono essere aggiornati i valori di **N** e dei vettori **PROC\_ID** e **PRTY**.

Si raccomanda, al fine di semplificare la leggibilità dell'algoritmo, di utilizzare rigorosamente i nomi indicati delle variabili e di descrivere l'algoritmo con un flow-chart (o pseudocodice) rigorosamente strutturato.

**Avvertenze**

**I risultati della prova saranno pubblicati sul sito.**

**La data, l'ora e l'aula della prova orale saranno rese note in calce ai risultati della prova scritta.**