

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
 CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2009/2010
 Parte Generale – 07 aprile 2010

Esercizio -1: essersi iscritti correttamente per svolgere questa prova.

Esercizio 0: Scrivere correttamente nome, cognome, matricola e posizione in tutti i fogli prima di svolgere ogni altro esercizio. Scrivere esclusivamente a penna senza abrasioni. E' vietato l'uso delle penne cancellabili, della matita, dei coprenti bianchi per la correzione (bianchetto) e la scrittura in colore rosso (riservato alla correzione).
 Il compito e' formato da due fogli, quattro facciate compresa questa. Le soluzioni che si vuole sottoporre per la correzione devono essere scritte negli spazi bianchi di questi fogli. Non verranno corretti altri supporti.
 E' obbligatorio consegnare il compito, e' possibile chiedere che esso non venga valutato scrivendo "NON VALUTARE" in modo ben visibile nella prima facciata.

Esercizio 1: Il seguente File System di tipo UNIX contiene molteplici errori. Si chiede al candidato di trovare gli errori e di correggerli come farebbe il programma fsck (file system check) (blocco da 1k, 16 i-node, 16 blocchi).

```
l-node[0](radice): size=60;link=6;type='d';direct[0]=4
l-node[1]: size=44;link=1;type='l';direct[0]=7
l-node[2]: size=28;link=3;type='d';direct[0]=0
l-node[3]: size=1026;link=3;type='f';direct[0]=12,direct[1]=5
l-node[6]: size=2050;link=1;type='f';direct[0]=12,direct[2]=2
l-node[8]: size=16;link=3;type='d';direct[0]=3
l-node[10]: size=44;link=4;type='d';direct[0]=14
l-node[12]: size=32;link=2;type='f';direct[0]=9
data[0]={(".",2),("..",0),("passwd",12)}
data[2]="cc"
data[3]={(".",8),("..",0)}
data[4]={(".",0),("..",0),("boot",6),("lost+found",8),("etc",2),("tmp",10)}
data[5]="dd"
data[7]="/etc/passwd"
data[9]="root:x:0:0:root:/root:/bin/bash\n"
data[12]="aaaa....." (1024 volte a)
data[14]={(".",10),("..",0),("data",1),("p2",12)}
data[15]="bbbb....." (1024 volte b)
bit-map i-node liberi: 0xeab2 (il bit meno significativo indica l'i-node 0)
bitmap blocchi liberi: 0x2d62(il bit meno significativo indica il blocco 0)
```

Esercizio 2: Mostrare una stringa di riferimenti infinita relativa a quattro pagine che faccia in modo che gli algoritmi di rimpiazzamento MIN e LRU si comportino esattamente allo stesso modo. Si chiede che nessun frame contenga definitivamente (da un certo istante in poi) la stessa pagina.

Esercizio 3: In questo esercizio viene utilizzato un meccanismo di scheduling a 5 classi di priorit  (0,1,2,3,4). In ogni istante viene scelto il primo processo della coda non vuota a priorit  maggiore e viene eseguito per un time slice di 2 ms.
 E' inoltre in funzione un meccanismo di aging. Ogni processo ha una priorit  propria. Ogni processo alla partenza o dopo che e' stato eseguito entra nella classe di priorit  corrispondente alla propria. Tutti i processi tranne quello appena eseguito che non sono nella classe 4, vengono spostati in coda alla classe di priorit  superiore ogni volta che lo scheduler viene richiamato.
 Esistono nel sistema 3 processi (A,B,C) e 5 risorse di un'unica classe gestite con il meccanismo del Banchiere monovaluta. Il processo A ha priorit  propria 4, ha 1 risorsa su un massimo di 4. Il processo B ha priorit  propria 2 e 1 risorsa (max 1) e il processo C ha priorit  statica 0 e 2 risorse su un massimo di 4.
 I processi devono fare le seguenti operazioni:
 A: calcola per 1ms, chiede una ulteriore risorsa, calcola per 1 ms, rilascia le 2 risorse.
 B: calcola per 16ms, rilascia la risorsa
 C: calcola per 5ms, restituisce le due risorse
 (1) Mostrare che la situazione iniziale e' safe
 (2) Mostrare il diagramma di GANNT con l'indicazione del calcolo delle priorit  e i passi di esecuzione dell'algoritmo del banchiere.
 (3) A e' a massima priorit  pero' la sua esecuzione subisce ritardi? Perche'?