

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
 PROVA SCRITTA DI SISTEMI OPERATIVI
 ANNO ACCADEMICO 2013/2014
 17 febbraio 2014

Esercizio -1: Essere iscritti su AlmaEsami 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 tre fogli, sei facciate compresa questa. Le soluzioni che si vogliono 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.
 Per svolgere questo compito occorre solo una penna e un documento di identità valido. La consultazione o anche solo la disponibilità di altro materiale comporterà l'annullamento del compito (verrà automaticamente valutato gravemente insufficiente).

Esercizio c.1: Scrivere il monitor altcolbb. Il monitor deve implementare le seguenti procedure entry:

```
void write(color_t color, generic_type val);
generic_type read(void);
```

Esistono elementi di due colori diversi, rosso e blu. Altcolbb deve implementare un buffer limitato con la particolarità che gli elementi devono essere inseriti nel buffer (fino a un numero massimo MAX) a colori alternati: se l'ultimo elemento nel buffer e' rosso il prossimo deve essere blu e viceversa. Quando il buffer e' vuoto il primo elemento inserito può essere sia rosso sia blu.

Esercizio c.2:

Si consideri il programma che ha tre variabili globali:

```
semaphore s1(1);
semaphore s2(1);
semaphore mutex(1)
int n=0;
```

e due thread:

```
Thread A {
  register int i;
  for (i=1;i<3;i++) {
    s1.P();
    mutex.P();
    n+=i;
    mutex.V();
    s2.V();
  }
}
```

```
Thread AB{
  register int j;
  for (j=2;j<4;j++) {
    s2.P();
    mutex.P();
    n*=j;
    mutex.V();
    s1.V();
  }
}
```

Mostrare tutti i possibili valori di n al termine dell'esecuzione, corredando la risposta con opportuna spiegazione del ragionamento seguito per trovare il risultato.

Esercizio g.1:

a) Sia data la seguente stringa di riferimenti: 012345012340123012010.

mostrare il comportamento degli algoritmi MIN e FIFO quando operano su una memoria di 4 frame.

b) Data una memoria di 4 frame contenente le pagine 0,1,2,3 rispettivamente nei frame 0,1,2,3. Mostrare una stringa di riferimenti di un programma che usi 5 pagine (esiste la pagina 4 non ancora mappata in memoria oltre alle 4 cariate nei frame) e che consenta alla fine dell'esecuzione di avere nel frame 0 la pagina nel frame 1 la pagina 2 nel frame 3 la pagina 4 e nel frame 3. L'algoritmo di rimpiazzamento da usare e' MIN.

Esercizio g.2:

Rispondere alle seguenti domande:

1. Quali sono i casi di frammentazione interna ed esterna che possono avvenire usando la paginazione.
2. Per il file system di tipo fat non e' efficiente l'accesso diretto (ad esempio con la system call lseek) a file di grandi dimensioni. Perche'?
3. Per quali tipi di processo e' indicato uno schedule a priorita' statica? Quale problema può essere causato da uno scheduler a priorita' statica? perche'?
4. Cosa e' un knot in un grafo? Quale teorema lega la definizione di knot in un grafo e la presenza di deadlock fra processi?