Nome_____ Cognome _____N. di matricola (10 cifre) _____ Riga ____ Col ____

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
PROVA SCRITTA DI SISTEMI OPERATIVI
ANNO ACCADEMICO 2020/2021
15 settembre 2021

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 alrv (at least rendez-vous) che fornisce una sola procedure entry:
`procedure entry void at_least(int n)`
Quando un processo chiama la funzione `at_least` vuol dire che vuole sincronizzarsi con un insieme di almeno n processi (incluso il chiamante).
Esempi:
Se un processo chiama `at_least(1)` non si blocca.
Se il processo p chiama `at_least(2)` si blocca, se poi il processo q chiama `at_least(2)` oppure `at_least(1)` si sbloccano sia p sia q (la richiesta di p è soddisfatta, ne aspettava almeno 2, p e q)
Se il processo p chiama `at_least(2)` si blocca, se poi il processo q chiama `at_least(3)` si blocca anch'esso perché sebbene la richiesta di p possa essere soddisfatta, q non può ancora sbloccarsi: ci sono solo 2 processi in attesa mentre q ne vuole 3. Un terzo processo che chiami `at_least(x)` con x=1,2,3 li sblocca tutti.
Hint: sia $w[k]$ il numero dei processi in attesa di essere in almeno k (quelli che hanno chiamato `at_least(k)` e non hanno completato l'esecuzione della funzione). Sia $s[n]=\sum_{k=1}^{n} w[k]$ (rappresenta il numero di processi soddisfacibili: e.g. se ci sono 4 processi in attesa, potrebbero essere soddisfatte le richieste dei processi che ne aspettano almeno 2, almeno 3 o almeno 4). Preso, se esiste, il massimo indice m tale che $s[m] \geq m$ tutti i processi in attesa di essere in n, per $n \leq m$ possono essere sbloccati.

Esercizio c.2: Dato un servizio di message passing sincrono scrivere, senza fare uso di processi server, un servizio di message passing sincrono concatenato che abbia le seguenti primitive:

```
void chained_send (T msg, list_of_pids dests)
T chained_recv(void)
```

La funzione `chained_send` deve fare in modo che tutti i processi indicati nella lista `dests` ricevano il messaggio. Il processo che chiama la `chained_send` si blocca solo fino a quando il primo processo della lista `dests` non chiama una `chained_recv`, il primo si sblocca quando il secondo chiama la `chained_recv` e così via.
(la funzione `chained_recv` riceve messaggi provenienti da qualsiasi mittente)

Esercizio g.1: Usando l'algoritmo di rimpiazzamento FIFO la stringa di riferimenti 1,2,3,4,1,2,5,1,2,3,4,5 genera più page fault con una memoria di 4 frame di quanti ne vengano generati con una memoria di 3 frame (anomalia di Belady)
i) si generi una stringa di riferimenti che generi più page fault in una memoria di 5 frame rispetto ad una da 4 frame.
ii) si costruisca una regola generale per costruire stringhe di riferimenti che generino più page fault in una memoria di n frame rispetto a quelli generati in una memoria da `n - 1` frame

Esercizio g.2: rispondere alle seguenti domande (motivando opportunamente le risposte):
a) Lo scheduler sceglie fra i processi ready quale porre in esecuzione. Ma cosa succede quando la coda dei processi ready è vuota?
b) Quali vantaggi offre RAID5 rispetto a RAID1?
c) Se per effetto di un guasto o di un bug il numero di hard link di un i-node è errato, quali conseguenze possono esserci?
d) Quali eventi causano la valutazione dell'algoritmo del banchiere? Cosa si fa se lo stato risulta unsafe e cosa invece se è safe?