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 2018/2019
15 luglio 2019

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: Scopo di questo esercizio è di scrivere un monitor *pairbuf* che implementi un buffer illimitato ad accoppiamento. Ogni processo che vuole scrivere un elemento nel buffer chiama la funzione di entrata:
`void put(T x)`
mentre ogni processo che vuole leggere un elemento dal buffer chiama la funzione:
`T get(void)`
Gli elementi devono essere consegnati in ordine FIFO. Sia la get sia la put sono bloccanti. Solo quando il numero degli scrittori che hanno chiamato la put è uguale a quello dei lettori che hanno chiamato la get tutti i processi in attesa vengono sbloccati.
Quindi per esempio, se un solo scrittore ha chiamato la put, quando arriva un lettore entrambi vengono sbloccati, e il lettore riceverà il valore passato come parametro dallo scrittore. Se vi fossero 10 lettori in attesa occorrono 10 scrittori perche' i 20 processi continuino l'esecuzione. Chi ha chiamato la get per primo riceverà il dato passato dal primo chiamante della put e così via in ordine FIFO.

Esercizio c.2:

```
fun dilemma(x):
  asend((getpid(), x, ""), serverproc)
  return arecv(serverproc)
```

```
serverproc: process
  while True:
    (pid, x, y) = arecv(None)
    if x == "":
      asend(y, pid)
    else
      asend((pid, x[1:], y + x[0]), getpid())
```

Cosa calcola la funzione dilemma tramite il processo serverproc? Descrivere come avviene il calcolo.
Se più processi chiamano la funzione dilemma in concorrenza il calcolo avviene in modo corretto? Perché?
Note: asend/arecv sono le primitive di un sistema di message passing asincrono. La funzione asend ha due parametri: il messaggio da inviare e l'identità del processo destinatario. Il parametro della funzione arecv indica il mittente atteso, None indica che vengono accettati messaggi da ogni mittente. La funzione getpid restituisce l'identità del processo chiamante.

Esercizio g.1: Sia dato l'algoritmo di rimpiazzamento MINREF che sceglie come pagina vittima quella che compare meno volte nella stringa dei riferimenti dall'inizio dell'esecuzione.
1a) mostrare una sequenza infinita nella quale MINREF si comporti come MIN. Mostrare come è stata costruita la sequenza.
1b) mostrare una sequenza infinita nella quale MINREF si comporti come FIFO. Mostrare come è stata costruita la sequenza.
(NB le sequenze infinite devono generare infiniti page fault).

Esercizio g.2: rispondere alle seguenti domande:
a) Perché una system call non può essere implementata come una normale chiamata di funzione ad un indirizzo del kernel?
b) Il calcolo del working set dipende dall'algoritmo di rimpiazzamento utilizzato? Perché?
c) Discutere la scelta di usare un file system FAT per una partizione contenente file soggetti a frequenti aggiornamenti e variazione di dimensione. E' una scelta appropriata o no? perché?
d) Quando viene richiamato l'algoritmo del Banchiere e cosa succede se il risultato dell'algoritmo mostra che lo stato non è safe?