

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
PROVA **PARZIALE** DI SISTEMI OPERATIVI
ANNO ACCADEMICO 2022/2023
20 gennaio 2023

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 due fogli, quattro 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 fullbuf che abbia le seguenti procedure entry:

```
void add(int value)
int get(void)
```

Le prime MAX chiamate della procedure entry **add** devono bloccare i processi chiamanti. In seguito deve sempre valere $N_a \geq MAX$ indicando con N_a il numero di processi bloccati in attesa di completare la funzione **add**.

La funzione **get** deve attendere che $N_a > MAX$, restituire la somma algebrica dei parametri value delle chiamate **add** in sospenso e riattivare il primo processo in attesa di completare la **add** (se la **get** richiede che $N_a > MAX$, la **get** può riattivare un processo e al completamento della **get** si rimarrà garantito che $N_a \geq MAX$)

Esercizio c.2: Dato un **servizio** di message passing asincrono implementare un **servizio** di message passing sincrono a scambio che prevede una sola chiamata:

```
T xchange(T msg, pid_t dest)
```

dove T è un tipo generico (potrebbe essere int, string, struct mystruct). Il parametro dest non può essere ANY.

Quando il processo P chiama **xchange**(m1, Q) e il processo Q chiama **xchange**(m2, P) entrambi i processi vengono riattivati, la **xchange** chiamata da P restituirà m2 mentre la **xchange** di Q restituirà m1.