

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA  
 PROVA SCRITTA DI SISTEMI OPERATIVI  
 ANNO ACCADEMICO 2023/2024  
 25 giugno 2024

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 choicsem che implementa un semaforo con scelta che ha le seguenti procedure entry:

```
void P(void)
void V(int n)
```

Il semaforo con scelta soddisfa l'invariante dei semafori ma i processi non vengono riattivati in ordine FIFO.

La funzione V se vi sono in attesa al più n processi riattiva il primo altrimenti viene riattivato quello in posizione n + 1 nella sequenza di attesa.

es. se ci sono in attesa i processi A, B e C, la funzione V(0) attiverrebbe A, V(1) attiverrebbe B, V(2) attiverrebbe C, per tutti i valori del parametro maggiori di 2 verrebbe riattivato A.

**Esercizio c.2:** Sia dato un sistema nel quale è presente un meccanismo di message passing asincrono che consente di inviare messaggi di lunghezza massima 256 byte e che ha le seguenti primitive:

```
lasend(byte *msg, pid_t dest)
byte *larecv(pid_t sender)
```

I messaggi che larecv riceve da ogni mittente sono consegnati in ordine FIFO.

In tale sistema scrivere (senza fare uso di processi server) un servizio di message passing asincrono per messaggi di lunghezza arbitraria:

```
asend(msg_t *msg, pid_t dest)
msg_t arecv(pid_t sender)
```

sapendo che sizeof(pid\_t) vale 4 e che esistono funzioni per calcolare la lunghezza di un messaggio e per scomporre e ricomporre sottosequenze dei messaggi (definite per queste funzioni l'API che ritenete più opportuna).

**Esercizio g.1:** Sia dato un sistema monoprocesso con una unità di input output dove sono in esecuzione i processi p1, p2 e p3. Il diagramma di GANNT dell'esecuzione è il seguente.

```
CPU |1|1|1|3|3|2|1|1|1|1|1|3|3|3|1|3|3|2|1|2| |3| | |1| |2|
I/O | | |1|1|1|3|3|3| | |1|1|1| |1|1|1|3|3|3|1|1|1|2|2| |
```

CI sono casi di preemption? Quale algoritmo di scheduling di CPU appare sia stato utilizzato e quale scheduler per l'input-output? Ricostruire la sequenza delle richieste dei tre processi.

**Esercizio g.2:** rispondere alle seguenti domande (*motivando opportunamente le risposte!*):

- a) L'algoritmo di rimpiazzamento LeastUsed sceglie come pagina vittima quella che ha avuto il minor numero di riferimenti da quando è in memoria. Quali funzionalità dovrebbe avere la MMU per poter implementare questo algoritmo di rimpiazzamento? Possono esistere MMU che forniscono le funzionalità richieste?
- b) Perché è bene che l'area di memoria dove viene memorizzato lo stack di valutazione non sia eseguibile?
- c) RAID0 non tollera guasti. Quali sono i casi di utilizzo di RAID0?
- d) Perché è necessario usare spinlock in sistemi multiprocesso per implementare kernel di tipo simmetrico (SMP)?