

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA  
 PROVA SCRITTA DI SISTEMI OPERATIVI  
 ANNO ACCADEMICO 2020/2021  
 12 luglio 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:** La fase finale di un torneo sportivo coinvolge  $2^N$  squadre numerate  $0, 1, \dots, 2^N - 1$  e si svolge ad eliminazione diretta. Al primo turno la squadra 0 gioca con la squadra 1, la 2 con la 3 e così via. Al secondo turno la vincente fra le squadre 0 e 1 incontra la vincente fra 2, 3 ecc. Il procedimento continua fino ad individuare la sola squadra vincente del torneo. Scrivere il monitor torneo. L'attività delle squadre è la seguente:

```
Squadra: process( i, i ∈ {0, 1, ..., 2N - 1} )
for turno in range(1, N):
    forma = valutaforma(i, turno)
    if torneo.gioca(i, turno, forma) == False:
        print(f"la squadra {i} ha perso al {turno} turno")
    return
print(f"la squadra {i} ha vinto il torneo")
```

La procedura entry gioca di volta in volta confronta il valore del parametro forma delle due squadre e dichiara vincitrice quella con valore massimo, se i valori coincidono decide in modo casuale (rigori). Alla squadra vincente viene restituito il valore True, alla perdente il valore False.

**Esercizio c.2:** Quella che segue vorrebbe essere l'implementazione di un semaforo unfair inizializzato a zero ma è errata. Mostrare un caso nel quale il valore del semaforo (value) non venga correttamente mantenuto. Correggere il codice usando il passaggio del testimone (passing le baton).

```
class: wrongsem
int value = 0, count = 0
semaphore mutex init 1;
semaphore s init 0;

void wV():
    mutex.P()
    if value == 0 && count > 0:
        s.V()
    else:
        value++
        mutex.V()

void wP()
    mutex.P()
    if value == 0:
        count++
        mutex.V()
        s.P()
        mutex.P()
        count--
    else:
        value--
        mutex.V()
```

**Esercizio g.1:** Sia dato l'algoritmo di rimpiazzamento modulo che data una memoria di NF frame, se avviene un page fault in corrispondenza dell'i-mo elemento della stringa di riferimenti modulo sceglie come pagina vittima quella che occupa il frame  $i \% NF$ .

Es. se NF fosse 3, la prima pagina viene messa nel frame 1 perché  $1 \% 3 = 1$ , se l'undicesimo elemento della stringa causa un page fault, si elimina la pagina nel frame 2 ( $11 \% 3 = 2$ ).

A) Considerare la stringa di riferimenti seguente e mostrare il funzionamento di modulo nei casi  $NF = 3$  e  $NF = 4$ :

1 2 3 4 5 3 3 3 1 5

B) modulo è un algoritmo a stack?

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

a) Perché il nome del file non è memorizzato all'interno dell'i-node nei file system tipo UNIX (e.g. ext2/3/4)?

b) Un bug di tipo buffer overflow consente ad un attaccante di spedire più dati di quelli che il buffer di ricezione può contenere, tracimando così nelle aree di memoria di altre variabili. Come è stato possibile in tanti casi che venisse spedito codice macchina e che il programma vittima dell'attacco lo eseguisse?

c) come fa l'allocazione gerarchica ad evitare che si possano verificare casi di deadlock?

d) Per rendere uno scheduler round-robin più pronto a servire i processi interattivi si decide di dimezzare la durata del time slice. Quali effetti collaterali può avere la scelta?