

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2008/2009
CONCORRENZA – 02 febbraio 2010

Esercizio -1: essersi iscritti correttamente 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 vuole 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.

Esercizio 1: Siano dati cinque processi "stampa" come segue:

stampa(x):(x=A,B,C,D,R) process

```
abracadabra.init(x)
while (true) {
    print(x);
    abracadabra.synchro(x);
}
```

scrivere il monitor abracadabra con due procedure entry `init(x)` `synchro(x)` che consenta di avere in output una sequenza infinita di ABRACADABRA (i.e. ABRACADABRAABRACADABRA ABRACADABRAABRACADABRA

Esercizio 2: Nel compito di gennaio si chiedeva di implementare uno scheduler cooperativo. Ogni processo chiama `init` all'inizio della sua esecuzione e `fini` al termine. Un solo processo alla volta e' in esecuzione. La funzione `yield` consente l'avvicendamento fra processi (quando il processo in esecuzione la chiama viene sospeso, il controllo viene ceduto al prossimo processo attivo). La soluzione era la seguente:

```
semaphore s0 init 1;
Init(): so.P()
Yield(): s0.V(); s0.P()
Fini(): s0.V();
```

Si chiede ora di modificare la soluzione di gennaio per fare in modo che lo scheduler cooperativo gestisca le operazioni di Input-Output. Esistono nel sistema 10 unita' di I-O. Dopo che un processo ha richiesto una operazione di input output sulla unita' n ($0 \leq n < 10$) richiama la funzione `ioreq(n)`, quando l'hardware genera un interrupt per l'unita' m ($0 \leq m < 10$) l'interrupt handler richiama la funzione `interrupt(m)`.

Scrivere le funzioni `ioreq` e `interrupt`. `ioreq` deve sospendere il processo chiamante, consentire l'esecuzione ad altri processi ready e attendere l'interrupt. Controllate che sempre al piu' un processo sia running (i.e. non bloccato nella `yield` o nella `ioreq`).

Esercizio 3: Siano date le seguenti funzioni:

- i. `foo(a,b) = < a >>= b; b >>= a >` (dove `x >>= y` significa `x = x >> y`, shift a destra di y posizioni)
- ii. `bar(a,b) = < a ^= b; b ^= a >` (dove `^` e' l'operatore di or esclusivo)

Quali fra queste funzioni possono essere utilizzate per realizzare meccanismi supporto di sezioni critiche come la Test&Set? (Motivare le risposte).

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2008/2009
PARTE GENERALE – 02 febbraio 2010

Esercizio -1: essersi iscritti correttamente per svolgere questa prova.

Esercizio 0: Scrivere correttamente nome, cognome, matricola e posizione prima di svolgere ogni altro esercizio.

Esercizio 1: Considerare I seguenti processi gestiti mediante uno scheduler round robin con time slice di 3ms:

P1: cpu 10ms; I-O 10 ms; cpu 10 ms; I-O 10ms; cpu 5ms

P2: cpu 2ms; I-O 10 ms; cpu 10 ms; I-O 10ms; cpu 5ms

l'Input-Output avviene su unita' indipendenti fra I due processi (non vi e' alcuna contesa di accesso).

Mostrare I due diagrammi di gannt che si ottengono se la ready queue contiene inizialmente la sequenza P1,P2 oppure la sequenza P2,P1.

Esercizio 2: Trovare una stringa infinita di riferimenti ad una memoria di 3 frame per la quale MIN e FIFO si comportino esattamente allo stesso modo (la stringa deve generare infiniti page fault e essere relativa a un numero finito di pagine).

Esercizio 3: (obbligatorio) Dato N e' il numero di matricola del candidato, calcolare $M=N\%5$. Dato il capitolo del corso di Sistemi Operativi contrassegnato dal numero M nella lista che segue, indicarne in modo schematico il contenuto e I concetti principali.

0: Gestione della Memoria Secondaria

1: Gestione della Memoria Principale

2: Scheduling della CPU

3: Gestione delle Risorse

4: File System



