

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2008/2009
CONCORRENZA – 10 luglio 2009

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: Un servizio prevede N tipi di diverse attività. Le attività dello stesso tipo possono venir elaborate in modo concorrente mentre attività di tipo diverso debbono venir gestite in modo indipendente. Non c'è un limite massimo al numero di attività dello stesso tipo in fase di elaborazione.

Prima di iniziare una attività di tipo k un processo chiama:

```
nact.begin(k)
```

e al termine:

```
nact.end(k)
```

La soluzione deve:

- ammettere realmente esecuzioni concorrenti (la soluzione banale di mera mutua esclusione fra processi sarà considerata errore)
- essere immune da starvation.

Esercizio 2: Sia data la seguente funzione atomica:

```
atomic int F(int x) {  
    static int y;  
    int tmp;  
    tmp=y; y=x;  
    return tmp;  
}
```

(dove static ha il significato previsto dal linguaggio C).

(a) E' possibile implementare una mutua esclusione usando la funzione F in modo simile alla Test&Set. Come?

(b) E' invece possibile implementare un servizio di mutua esclusione dove piu' sezioni critiche possano essere accedute in modo indipendente?

Esercizio 3: Siano dati due processi che operano su un semaforo S che ha valore iniziale 2.

A: process: S.P(); S.P(); print("P"); S.V(); S.V();

B: process: S.P(); S.P(); print("Q"); S.V(); S.V();

(la funzione print e' considerata atomica per lo scopo di questo esercizio).

Descrivere (con rappresentazione grafica) tutte i possibili stati di esecuzione dell'insieme di processi {P,Q}.

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
 CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2008/2009
 PARTE GENERALE – 10 luglio 2009

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: Sia data una memoria di tre frame e due processi.

Il processo P viene attivato al tempo 0 e ha la seguente stringa di riferimenti: 1, 2, 3, 4, 12

Il processo Q viene attivato al tempo 5ms e ha la seguente stringa di riferimenti, 10, 11, 20.

Gli accessi sono in lettura e le pagine si trovano su di un disco che impiega 2 ms per leggere dalla traccia corrente e 2ms a traccia per la seek. Le pagine da 0 a 9 sono sulla traccia 0, quelle da 10 a 19 nella traccia 1, da 20 a 29 nella 2 e così via. Il disco è inizialmente sulla traccia 0 e usa l'algoritmo LOOK (ascensore).

Mostrare come avviene l'esecuzione dei due processi.

Esercizio 2: Si confrontino le esecuzioni degli algoritmi SJN (non preemptive) e SRTF (preemptive) nell'esecuzione dei seguenti programmi.

```
P1: for (i=0;i<3;i++) {
  read(1ms);
  compute(3ms);
  write(2ms);
}
```

```
P2: for (i=0;i<3;i++) {
  read(1ms);
  compute(1ms);
  write(2ms);
}
```

Le letture e le scritture avvengono su un unico device (FIFO).

I tempi dei context switch e di valutazione del for si intendono trascurabili ai fini didattici.

Esercizio 3: Se N è il numero di matricola calcolare $M=N\%5$. Prendere l'oggetto M e descrivere:

- a) a cosa serve
- b) esempi reali di utilizzo

1)

oggetto0: sottosistema di ambiente

oggetto1: autotools (autoconf, automake)

oggetto2: buddy list

oggetto3: linker dinamico

oggetto4: sistemi di controllo di versione distribuiti (ad esempio: cvs (concurrent version system), svn (subversion), git (fast version control system))



