

PCAD a.a. 2021/22 - Scritto del 21 giugno 2021

Esercizio 1 (4 punti)

Assumiamo che "++" sia atomica e che lo scheduler assegni una parte di tempo di CPU ad ogni thread non bloccato. Discutere i possibili risultati (effetto su array A, errori, terminazione/non terminazione) del seguente programma:

INIZIALIZZAZIONE (sequenziale prima di lanciare i thread)

array A[3]={0,0,0}; // gli indici di A partono da 0 cioè A[0]=A[1]=A[2]=0

i,j=0;

THREAD P:

```
while (i<2) {  
    A[i]=1;  
    i++;  
}
```

THREAD Q:

```
A[i]=2;  
A[j]=3;
```

THREAD R:

```
while (j<2) {  
    A[j]=3;  
    j++;  
}
```

Esercizi 2 e 3 nella pagina seguente

Esercizio 2 (6 punti)

Considerate un programma P costituito dai seguenti 3 thread

T1

X=1;

Y=1;

T2

Z=1;

r1=X-1;

T3

while (Y != 1) { }

while (Z != 1) { }

r1 = X+1

Senza particolari assunzioni di atomicità negli assegnamenti e assumendo che tutte le variabili siano =0 ad inizio dell'esecuzione, spiegare i possibili risultati del programma P con particolare attenzione per il valore finale in r1 nei due casi seguenti:

- un modello di esecuzione che garantisce consistenza sequenziale
- un modello di esecuzione che non garantisce consistenza sequenziale

Esercizio 3 (10 punti)

Considerate due tipologie di processi:

Lettori: leggono il valore di una risorsa condivisa (es una variabile)

Scrittori: leggono o sovrascrivono il valore della risorsa

Usando semafori scrivere un algoritmo concorrente per definire il comportamento di un generico processo Lettore e di un generico processo Scrittore per garantire le seguenti proprietà:

- Più Lettori possono accedere simultaneamente alla risorsa condivisa
- Gli Scrittori possono accedere alla risorsa solo in mutua esclusione rispetto sia a Lettori che Scrittori