

Elementi di Architettura e Sistemi Operativi

Bioinformatica - Tiziano Villa

18 Settembre 2018

Nome e Cognome:

Matricola:

Posta elettronica:

problema	punti massimi	i tuoi punti
problema 1	9	
problema 2	6	
problema 3	5	
problema 4	10	
totale	30	

1. (a) Si descrivano concisamente le primitive di sincronizzazione Semaforo e Monitor.

Nel descrivere la primitiva Monitor si spieghi la differenza tra la schedulazione alla Mesa o alla Hoare.

Traccia di soluzione.

Si veda il libro di testo per una descrizione completa.

Mesa: *signal* non passa il lucchetto e il processore al processo risvegliato dall'attesa che e' messo nella coda dei processi pronti e quindi non e' detto che esegua subito.

Hoare: *signal* passa il lucchetto e il processore al processo risvegliato dall'attesa, il quale esegue immediatamente e restituisce il lucchetto e il processore quando esce dalla sezione critica o si mette di nuovo in attesa.

(b) Il Semaforo sia realizzato con le seguenti classi:

```
public class Semaphore {
    public Semaphore (int initialValue) {
        /* Si crea un semaforo con valore iniziale:
           initialValue */
        ...
    }
    public P() {
        /* Si chiama P() sul semaforo */
        ...
    }
    public V() {
        /* Si chiama V() sul semaforo */
        ...
    }
}
```

Il Monitor sia realizzato con le seguenti classi:

```
public class Lock {
    public Lock {
        /* Si crea un nuovo lucchetto Lock */
        ...
    }
    public void Acquire() {
        /* Si acquisisce un lucchetto Lock */
        ...
    }
    public void Release() {
        /* Si rilascia un lucchetto Lock */
        ...
    }
}

public class CondVar {
    public CondVar (Lock lock) {
        /* Si crea una nuova variabile di condizione
```

```
        associata con il lucchetto Lock lock */
        ...
    }
    public void Wait() {
        /* Aspetta sulla variabile di condizione */
        ...
    }
    public void Signal() {
        /* Risveglia un processo (se esiste) */
        ...
    }
    public void Broadcast() {
        /* Risveglia tutti i processi che aspettano su cv */
        ...
    }
}
```

Per dimostrare che si possono realizzare semafori con monitor, si completi il codice seguente che realizza la primitiva Semaforo mediante la primitiva Monitor, cioè' avendo a disposizione le classi Lock e CondVar (si assume la schedulazione alla Mesa).

```
public class Semaphore {
    Lock lock;
    CondVar c;
    int value;

    public Semaphore(int initialValue) {
        value = ?;
        lock = new ?;
        c = new ?;
    }
    public P() {
        lock.?.;
        while (value ?)
            c.?.;
        value = ?;
        lock.?.;
    }
    public V() {
        lock.?.;
        value = ?;
        c.?.;
        lock.?.;
    }
}
```

Traccia di soluzione.

```
public class Semaphore {
    Lock lock;
    CondVar c;
    int value;

    public Semaphore(int initialValue) {
        value = InitialValue;
        lock = new Lock();
        c = new CondVar(lock);
    }
    public P() {
        lock.Acquire();
        while (value == 0)
            c.Wait();
        value = value -1;
        lock.Release();
    }
    public V() {
        lock.Acquire();
        value = value +1;
        c.Signal();
        lock.Release();
    }
}
```

2. (a) Si descriva brevemente lo schema della memoria virtuale, incluso il ruolo della TLB e della tavola delle pagine.

(b) Supponiamo che un programma abbia appena fatto riferimento a un indirizzo nella memoria virtuale. Per ognuno dei seguenti casi ipotetici si descriva uno scenario nel quale tale caso possa verificarsi. Se non e' possibile che tale caso si verifichi, si spieghi il motivo dell'impossibilita'.

- Insuccesso della TLB senza assenza di pagina.

Traccia di soluzione.

E' possibile. La pagina e' stata portata da disco in memoria, ma e' stata rimossa dalla TLB.

- Insuccesso della TLB con assenza di pagina.

Traccia di soluzione.

E' possibile. La pagina non e' mai stata portata in memoria ne' nella TLB.

- Successo della TLB senza assenza di pagina

Traccia di soluzione.

E' possibile. La pagina e' in memoria e nella TLB. Probabilmente un accesso recente.

- Successo della TLB con assenza di pagina.

Traccia di soluzione.

E' impossibile. La TLB e' una cache della tavola delle pagine; se un dato non e' nella tavola delle pagine, esso non puo' essere nella TLB.

3. Alle istruzioni dell'architettura LC-3 sono state aggiunte due nuove istruzioni Push e Pop:

- *Push Rn* aggiunge alla cima della pila un elemento con il valore contenuto nel registro *Rn*;
- *Pop Rn* toglie il primo elemento dalla cima della pila e ne memorizza il valore nel registro *Rn*.

Sia dato il seguente contenuto dei registri:

R0 x0000
R1 x1111
R2 x2222
R3 x3333
R4 x4444
R5 x5555
R6 x6666
R7 x7777

Si supponga di eseguire le seguenti 6 istruzioni:

Push R4
Push ?
Pop ?
Push ?
Pop R2
Pop ?

e che alla fine si abbia il seguente contenuto dei registri:

R0 x1111
R1 x1111
R2 x3333
R3 x3333
R4 x4444
R5 x5555
R6 x6666
R7 x4444

Si sostituiscano i punti di domanda con i registri corrispondenti. consistenti con il codice e con i contenuti dei registri prima e dopo le operazioni.

Traccia di soluzione

```
    Push R4
*   Push R1
*   Pop  R0
*   Push R3
    Pop  R2
*   Pop  R7
```

4. Si progetti un circuito sequenziale che realizza un addizionatore seriale con la seguente specifica:

- Ci sono due variabili binarie in ingresso x e y , e una variabile binaria in uscita z .
- Le variabili in ingresso rappresentano due operandi binari da sommare letti a partire dalla cifra binaria meno significativa.
- La variabile in uscita rappresenta la somma binaria prodotta a partire dalla cifra binaria meno significativa.
- Si noti che non c'è un limite prefissato al numero di cifre binarie degli addendi letti in ingresso.
- Per semplicità si trascuri il riporto finale, quindi se i due ingressi hanno n cifre binarie, il circuito genera un'uscita con n cifre binarie (senza generare in uscita il riporto finale che corrisponderebbe alla $n + 1$ -esima cifra binaria dell'uscita).

- (a) Si progetti la macchina a stati finiti (tipo **Moore**) che modella la specifica disegnando il grafo delle transizioni. S'indichi lo stato iniziale.

Traccia di soluzione.

Tabella delle transizioni della macchina a stati finiti di tipo Moore:

00	NR0	NR0	0
01	NR0	NR1	0
10	NR0	NR1	0
11	NR0	R0	0

00	NR1	NR0	1
01	NR1	NR1	1
10	NR1	NR1	1
11	NR1	R0	1

00	R0	NR1	0
01	R0	R0	0
10	R0	R0	0
11	R0	R1	0

00	R1	NR1	1
01	R1	R0	1
10	R1	R0	1
11	R1	R1	1

- (b) Si minimizzi il numero degli stati della macchina proposta, applicando l'algoritmo di minimizzazione degli stati.
- (c) Si scriva la tavola delle transizioni con gli stati futuri e le uscite e la si codifichi.

- (d) Supponendo di usare bistabili di tipo D, si derivino le equazioni minimizzate di eccitazione degl'ingressi dei bistabili e le equazioni minimizzate delle uscite. Si esegua e mostri la minimizzazione con le mappe di Karnaugh.

- (e) Si realizzi il circuito sequenziale corrispondente con bistabili di tipo D campionati sul fronte di salita, invertitori e porte NAND. Si etichettino con chiarezza i segnali.