

Elementi di Architettura e Sistemi Operativi

Bioinformatica - Tiziano Villa

17 Settembre 2015

Nome e Cognome:

Matricola:

Posta elettronica:

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

1. Si descriva brevemente il problema dei cinque filosofi.

Si consideri la seguente soluzione proposta nel libro di testo di Silberschatz et al., dove ogni bacchetta e' rappresentata con un semaforo: un filosofo cerca di afferrare una bacchetta eseguendo un'operazione *wait* su quel semaforo e la depone eseguendo l'operazione *signal* su quel semaforo. I dati condivisi sono il vettore di semafori `semaforo bacchetta[5]`; con tutti gli elementi inizializzati a 1.

La struttura del filosofo *i* e' proposta come segue:

```
do {
    wait (bacchetta[i]);
    wait (bacchetta[(i+1) % 5]);
    ...
    mangia
    signal (bacchetta[i]);
    signal (bacchetta[(i+1) % 5]);
    ...
    pensa
} while (true);
```

E' possibile che si abbia una situazione di stallo ? Si argomenta la risposta.

E' possibile che si abbia una situazione di affamamento (starvation)?

Traccia di soluzione.

Il filosofo $i = 0, \dots, 4$ ha alla sua sinistra la bacchetta i e alla sua destra la bacchetta $i + 1 \% 5$.

Si potrebbe avere una situazione di stallo. Si consideri il caso che ogni filosofo afferri la bacchetta alla sua sinistra, dopo di che' nessun filosofo potrebbe afferrare l'altra bacchetta di destra e mangiare. Questa sarebbe anche una situazione di affamamento. Ma l'assenza di stallo non garantisce l'assenza di affamamento.

2. Siano dati 4 processi con durata dell'esecuzione e tempo d'arrivo espressi dalla seguente tabella:

Processo	Durata	Arrivo
P1	3	0
P2	5	1
P3	2	3
P4	2	9

- (a) Si disegni lo schema GANTT (come nel libro di testo) che illustri l'esecuzione di questi processi con i seguenti algoritmi di schedulazione: FCFS (First Come First Serve), SRTF (Shortest-Remaining-Time-First, cioè Shortest-Job-First con prelazione), RR (Round-Robin) con quanto di tempo = 1.

Traccia di soluzione.

FCFS:

```
P1 P1 P1 P2 P2 P2 P2 P2 P3 P3 P4 P4
0  1  2  3  4  5  6  7  8  9 10 11
```

SRTF:

```
P1 P1 P1 P3 P3 P2 P2 P2 P2 P2 P4 P4
0  1  2  3  4  5  6  7  8  9 10 11
```

RR:

```
P1 P2 P1 P3 P2 P1 P3 P2 P2 P4 P2 P4
0  1  2  3  4  5  6  7  8  9 10 11
```

- (b) Per i tre algoritmi precedenti si calcoli il tempo di completamento di ciascun processo. Si definisca il tempo di completamento come la differenza tra il tempo di terminazione e il tempo d'arrivo.

Traccia di soluzione.

Il tempo di completamento e' la differenza tra il tempo di terminazione e il tempo d'arrivo, cioe' il tempo trascorso da quando il processo arriva nella coda a quando termina. Nella letteratura in inglese e' designato anche come TRT (turnaround time), cioe' tempo trascorso.

Per FCFS i tempi di completamento sono:

P1 3,

P2 7,

P3 7,

P4 3

Per SRTF i tempi di completamento sono:

P1 3,

P2 9,

P3 2,

P4 3

Per RR i tempi di completamento sono:

P1 6,

P2 10,

P3 4,

P4 3

- (c) Per i tre algoritmi precedenti si calcoli il tempo di attesa di ciascun processo. Si definisca il tempo di attesa come la differenza tra il tempo di completamento e la durata.

Traccia di soluzione.

Il tempo di attesa e' la differenza tra il tempo di completamento e la durata, cioe' e' il tempo trascorso in attesa nella coda senza eseguire.

Per FCFS i tempi di attesa sono:

P1 0,

P2 2,

P3 5,

P4 1

Per SRTF i tempi di attesa sono:

P1 0,

P2 4,

P3 0,

P4 1

Per RR i tempi di attesa sono:

P1 3,

P2 5,

P3 2,

P4 1

- (d) Per i tre algoritmi precedenti si calcoli il tempo di attesa medio per l'esecuzione dei quattro processi.

Traccia di soluzione.

Tempo di attesa medio per FCFS: $(0+2+5+1)/4 = 2$.

Tempo di attesa medio per SRTF: $(0+4+0+1)/4 = 1,25$.

Tempo di attesa medio per RR: $(3+5+2+1)/4 = 2,75$.

Si noti che SRTF ottiene il tempo di attesa medio minimo come dalla teoria.

3. Si assuma che un sistema di calcolo abbia un indirizzo virtuale di 32 cifre binarie con una dimensione della pagina di 4KB, Dato l'indirizzo virtuale in decimale 19986, si determini il numero di pagina e lo scostamento (offset).

Traccia di soluzione.

$$4KB = 4 * 1024B = 4096B;$$

$19986B / 4096B = 4$ numero di pagina (quoziente della divisione, le pagine si contano da 0);

$$19986B \% 4096B = 3602 \text{ scostamento (resto della divisione).}$$

In alternativa si puo' notare che $19986_{10} = 4E12_{16}$, e quindi su 32 cifre binarie: $x00004E12$, dove le ultime tre cifre esadecimali sono lo scostamento (la dimensione di pagina e' $4KB = 2^{12}B$, che corrisponde alle 3 cifre esadecimali meno significative, cioe' alle 12 cifre binarie meno significative) e la cifra 4 precedente rappresenta il numero di pagina.

4. Si consideri il seguente programma scritto nel linguaggio macchina LC-3.

```
          .ORIG    x3000
          LD       R1, NUM
          LDR      R1, R1, #0
          BRzp     SALTA
          NOT      R1, R1
          ADD      R1, R1, #1
SALTA     ADD      R0, R1, #0
          HALT
NUM       .FILL    x4000
          .END
```

Si spieghi con chiarezza il funzionamento di tale programma, in base alle istruzioni del suo codice.

Traccia di soluzione

Calcola il valore assoluto del numero alla locazione di memoria d'indirizzo $x4000$ e lo scrive nel registro $R0$.

Si noti che la semantica di `LD` è $R1 \leftarrow mem[NUM]$ e di `LDR` è $R1 \leftarrow mem[R1 + 0]$, per cui le due istruzioni `LD`, `LDR` hanno l'effetto di scrivere in $R1$ il contenuto della cella di memoria all'indirizzo $x4000$, in altri termini il loro effetto cumulativo corrisponde a $R1 \leftarrow mem[mem[NUM]] = mem[x4000]$. Se il contenuto della cella a tale indirizzo risulta positivo, si salta a `SALTA` trascrivendolo in $R0$, altrimenti lo si nega (con le istruzioni `NOT`, `ADD` che eseguono la negazione in complemento a 2, invertendo le cifre binarie ed aggiungendo 1) e poi si continua con l'istruzione etichettata da `SALTA` (`ADD R0, R1, #0`) che copia il contenuto di $R1$ in $R0$.

5. Si consideri la macchina a stati finiti (di tipo Moore) completamente specificata a due ingressi (x_1, x_2) e due uscite $Z = (z_1, z_2)$ rappresentata dalla seguente tabella delle transizioni:

	00	01	11	10	Z
st0	st1	st3	st7	st4	00
st1	st7	st2	st4	st5	10
st2	st3	st4	st1	st0	11
st3	st0	st4	st2	st1	10
st4	st5	st6	st3	st0	11
st5	st7	st6	st6	st3	10
st6	st1	st6	st5	st0	11
st7	st5	st3	st0	st2	00

- (a) Applicando l'algoritmo di minimizzazione degli stati (algoritmo di Paull-Unger basato sulla tabella delle implicazioni) si minimizzi il numero degli stati della macchina proposta, e si mostri la tabella delle transizioni della macchina minimizzata.

Traccia di soluzione.

Si veda la soluzione dell'esercizio 2) nella sezione 6.8 del libro di Progettazione digitale.

- (b) Si codifichi la macchina minimizzata e si mostri la tavola delle transizioni codificata.

- (c) Supponendo di usare bistabili di tipo D, si derivino le equazioni minimizzate di eccitazione degl'ingressi dei bistabili e le equazioni minimizzate delle uscite.

- (d) 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.