

# Elementi di Architettura e Sistemi Operativi

Bioinformatica - Tiziano Villa

29 Giugno 2017

Nome e Cognome:

Matricola:

Posta elettronica:

problema	punti massimi	i tuoi punti
problema 1	7	
problema 2	8	
problema 3	5	
problema 4	10	
totale	30	

1. Si consideri il seguente paradigma di sincronizzazione, nel caso di due processi che competono per entrare in una sezione critica. Si assuma che:

- (a) Una sezione critica e' protetta se vi puo' accedere un solo processo per volta;
- (b) la sincronizzazione e' equa se ogni processo ha le medesima possibilita' di accedervi.
- (c) All'inizio vale

```
flagA = false;  
flagB = false;
```

Si supponga che si abbia un codice diverso per ogni processo:

**Processo A**

```
1A. flagA = true;  
2A. while (flagB == true)  
3A.     non si fa nulla;  
4A. si entra nella sezione critica;  
5A. flagA = false;
```

**Processo B**

```
1B. flagB = true;  
2B. while (flagA == true)  
3B.     non si fa nulla;  
4B. si entra nella sezione critica;  
5B. flagB = false;
```

Si risponda alle seguenti domande.

- (a) Questo meccanismo di sincronizzazione garantisce la protezione della sezione critica ?

Se si, si argomenta il motivo. Se no, si dia un esempio di esecuzione dei due processi che viola la protezione.

Traccia di soluzione.

Si. Il processo A entra nella sezione critica solo quando *flagB* e' falsa (il che succede solo mentre il processo B sta eseguendo prima di 1B o dopo 5B). Il processo B entra nella sezione critica solo quando *flagA* e' falsa (il che succede solo mentre il processo A sta eseguendo prima di 1A o dopo 5A).

- (b) Con questo codice puo' prodursi una situazione di stallo ? Se no, si argomenta il motivo. Se si, si dia un esempio di esecuzione dei due processi che produce una situazione di stallo.

Traccia di soluzione.

Si. Si supponga che il processo A esegua l'istruzione 1A (ponendo la variabile *flagA* a vero), e che poi sia interrotto e che il processore passi al processo B che esegue a sua volta 1B ((ponendo la variabile *flagB* a vero). A questo punto il sistema e' bloccato.

- (c) Si consideri la seguente variante del codice precedente e si risponda per tale variante alle due domande precedenti. Si noti che *yield()* e' una procedura per cui un processo rilascia volontariamente il processore che puo' quindi essere assegnato a un altro processo.

Processo A

```
1A. flagA = true;
2A. while (flagB == true) {
3A.     flagA = false;
4A.     yield();
5A.     flagA = true;
6A. }
7A. si entra nella sezione critica;
8A. flagA = false;
```

Processo B

```
1B. flagB = true;
2B. while (flagA == true) {
3B.     flagB = false;
4B.     yield();
5B.     flagB = true;
6B. }
7B. si entra nella sezione critica;
8B. flagB = false;
```

Traccia di soluzione.

Questa variante garantisce la protezione della sezione critica e previene lo stallo. [Si potrebbe immaginare una situazione di ciclo che non progredisce mai - livelock - in cui i due processi eseguono il corpo del ciclo while e rimettono a vero le loro bandierine per poi ritornare ad eseguire il ciclo while etc., ma prima o poi la situazione si sparglierebbe ed uno dei due processi entrerebbe nella sezione critica, in ogni caso non si tratterebbe tecnicamente di stallo.]

2. Si consideri il seguente scenario di memoria organizzata a due livelli con segmenti e pagine:

- (a) il numero di segmento virtuale e' rappresentato con 2 cifre binarie;
  - (b) il numero di pagina virtuale e' rappresentato con 8 cifre binarie;
  - (c) lo scostamento ("offset") e' rappresentato con 12 cifre binarie;
  - (d) un elemento della tavola delle pagine (PTE) e' rappresentato con 8 cifre binarie
  - (e) il numero di pagina fisica e' rappresentato con 8 cifre binarie.
- (a) Si spieghi il meccanismo di traduzione degl'indirizzi logici (22 cifre binarie) in indirizzi fisici (20 cifre binarie), aiutandosi con uno schema grafico se conveniente.

Traccia di soluzione.

Si vedano il libro di testo e le dispense per una spiegazione dello schema d'indirizzamento a due livelli con segmenti e pagine.

(b) Date la tavola dei segmenti e la memoria fisica allegate, si traducano i seguenti indirizzi logici nei corrispondenti indirizzi fisici:

- $0x\ 1103DB$

Traccia di soluzione.

Indirizzo non valido: il numero della pagina logica eccede la dimensione del segmento.

Si noti che dalla tavola dei segmenti (seconda riga, poichè il primo campo dell'indirizzo logico ha il valore 1) si ottiene l'indirizzo  $0x0000$  che deve essere incrementato di  $0x0010$  (secondo campo dell'indirizzo logico) dando l'indirizzo  $0x0010$  il cui contenuto è  $1E$ , il che farebbe ottenere l'indirizzo fisico  $0x1E3DB$ . Ma bisogna controllare anche la dimensione del segmento  $0x10$  (sempre dalla seconda riga della tavola dei segmenti). che corrisponde a 16 parole di memoria e termina all'indirizzo  $0x000F$  (la prima parola è all'indirizzo  $0x0000$  e la sedicesima è all'indirizzo  $0x000F$ ), che è l'indirizzo precedente al numero della pagina logica  $0x0010$  di nostro interesse, il quale quindi eccede di un'unità la dimensione del segmento.

- $0x\ 010350$

Traccia di soluzione.

$0x\ 16350$

Per entrambe le traduzioni, si veda l'allegato che mostra graficamente il processo di traduzione da indirizzo logico a fisico.

3. (a) Si spieghi che cosa sono i codici di condizione.

Si scriva una successione di istruzioni LC-3 che abbia come risultato che alla fine dell'esecuzione di quelle istruzioni i codici di condizione abbiano il valore  $N = 1, Z = 1, P = 0$ .

Traccia di soluzione.

I codici di condizione sono i valori di tre bistabili. posti a 1 o 0 a seconda del valore (positivo, negativo o nullo) dell'ultimo registro modificato. Ovviamente uno solo dei tre è posto a 1 e gli altri due sono posti a 0 in modo esclusivo.

Non è possibile che si produca la situazione indicata, perché si richiederebbe che il contenuto di un registro fosse al contempo negativo ( $N = 1$ ) e zero ( $Z = 1$ ).

- (b) Si scriva un'istruzione LC-3 che azzeri il contenuto del registro R2.

Traccia di soluzione.

AND R2, R2, #0

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

- C'è una variabile binaria in ingresso  $X$ , e una variabile binaria in uscita  $Z$ .
- L'uscita  $Z$  è asserita a 1 ogni volta che si rileva la sequenza in ingresso 01 oppure la sequenza in ingresso 10 (cioè gli ultimi due ingressi sono 01 o 10), altrimenti vale 0.
- **Si richiede la realizzazione come macchina di Mealy.**

(a) Si disegni il grafo delle transizioni di una macchina a stati finiti che realizza la specifica. S'indichi lo stato iniziale.

Traccia di soluzione.

Si vedano le dispense del corso per la soluzione dell'esercizio.



- (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.