

Elementi di Architettura e Sistemi Operativi

Bioinformatica - Tiziano Villa

18 Giugno 2010

Nome e Cognome:

Matricola:

Posta elettronica:

problema	punti massimi	i tuoi punti
problema 1	8(=4+2+1+1)	
problema 2	10	
problema 3	7	
problema 4	5	
totale	30	

1. Rispondere in modo preciso ma conciso alle seguenti domande.

- (a) Si elenchino i vari livelli di memoria, indicando per ognuno di essi la dimensione tipica, la tecnologia, il tempo di accesso, l'ampiezza di banda, e da chi e' gestito. Si precisino le unita' di misura di ogni quantita'.

Traccia di soluzione.

Si veda la Figura 1.9, p. 25.

- (b) Si consideri un sistema composto da quattro risorse dello stesso tipo condivise da tre processi, ciascuno dei quali necessita di non più di due risorse. Si argomenti se è possibile o meno che si verifichi una situazione di stallo.

Traccia di soluzione

Non è possibile che si verifichi uno stallo.

Per assurdo si supponga che ci sia uno stallo. Per questo è necessario che ogni processo trattiene una risorsa e sia in attesa di un'altra. Poiché ci sono 3 processi e 4 risorse, un certo processo deve poter ottenere due risorse; ma allora questo processo non abbisognerà di più risorse, e perciò restituirà le sue risorse quando avrà finito.

(c) Che cosa s'intende per firmware di una macchina ?

Traccia di soluzione.

Vedi Sec. 1.2.1.

Firmware e' un termine informatico sovraccarico di significati. Qui ci atterremo alle sfumature piu' rilevanti per i sistemi operativi.

Programmi di controlli di dispositivi elettronici che risiedono in memorie di solito inalterabili il cui supporto fisico e' parte integrante della macchina (esempi di tali memorie sono le ROM, memorie a sola lettura, e le EEPROM, memorie programmabili cancellabili elettronicamente). Tali programmi hanno lo scopo di avviare i dispositivi e farli interagire con altri. L'esempio piu' noto di firmware e' il BIOS (Basic Input Output System) che risiede sulla ROM BIOS e svolge soprattutto le funzioni di caricare il sistema operativo dal disco di avvio o da una porta per rete locale, ed eseguire il collaudo dei componenti al momento dell'accensione. Nella letteratura sulla progettazione dell'unita' di controllo di un processore, s'indica con la parola firmware il cosiddetto "microcodice", cioe' le microistruzioni dell'unita' di controllo microprogrammata (in contrapposizione a quella cablata in un circuito). Il termine significa materiale "ware" stabile "firm", per alludere che si tratta di programmi non modificabili direttamente dall'utente.

(d) Si descriva la tecnica di accesso diretto alla memoria (DMA).

Traccia di soluzione.

Vedi Sec. 1.2.3.

Il controllore della memoria trasferisce un intero blocco dalla propria memoria di transito ("buffer") direttamente nella memoria centrale o viceversa, senza alcun intervento del processore centrale. In tal modo, mentre il controllore del dispositivo effettua le operazioni di trasferimento, il processore centrale è libero di occuparsi di altri compiti. Tecnica particolarmente adatta per i trasferimenti da e a disco rigido.

2. Si consideri il seguente codice incompleto per scrivere le primitive di un semaforo P (*Wait*) and V (*Signal*) per un sistema uniprocessore.

```
typedef struct {
    int count;
} semaphore;

P(semaphore *S) {
    while (1) {
        disabilita le interruzioni;
        if (S->count ? ) {
            S->count ?;
            abilita le interruzioni
            return;
        }
        else abilita le interruzioni;
    }
}

V(semaphore *S) {
    disabilita le interruzioni;
    S->count ?;
    abilita le interruzioni
}
```

- (a) Si descriva brevemente che cos'è un semaforo e si mostri lo pseudo-codice della definizione classica delle operazioni P e V .

Traccia di soluzione.

Un semaforo è una variabile intera cui si può accedere, escludendo l'inizializzazione, solo tramite due operazioni atomiche predefinite: P (*Wait*) and V (*Signal*).

Le definizioni classiche di *Wait* e *Signal* in pseudo-codice sono le seguenti:

```
Wait(S) {  
    while (S <= 0)  
        ;  
    S--;  
}
```

```
Signal(S) {  
    S++;  
}
```

- (b) Si completi il codice iniziale sostituendo i "?" con le parti mancanti, in modo che realizzi correttamente le primitive di un semaforo, e lo si commenti con annotazioni per dimostrare che il codice completato e' corretto.

Traccia di soluzione.

La disabilitazione e riabilitazione delle interruzioni sono sufficienti per ottenere l'atomicita' delle operazioni in caso di processore singolo.

```
typedef struct {
    int count;
} semaphore;

P(semaphore *S) {
    while (1) {
        disabilita le interruzioni;
        if (S->count > 0 ) {
            S->count -= 1;
            abilita le interruzioni
            return;
        }
        else abilita le interruzioni;
    }
}

V(semaphore *S) {
    disabilita le interruzioni;
    S->count += 1;
    abilita le interruzioni
}
```


(c) Che problemi ha il codice precedente ?

Traccia di soluzione.

Problema fondamentale: attesa attiva in P (per il ciclo $while(1)$).

Non e' equo in P .

(Stallo in P per certe priorit').

- (d) Si modifichi il codice come segue: si aggiunga al semaforo una coda di processi in attesa; se P fallisce si aggiunga il processo alla coda, con V si rimuova il processo dalla coda se essa non e' vuota.

Si commenti la soluzione risultante rispetto a quella iniziale e ai problemi del punto precedente.

Traccia di soluzione.

```
typedef struct {
    int count;
    queue q;
} semaphore;

P(semaphore *S) {
    disabilita le interruzioni;
    if (S->count > 0) {
        S->count -= 1;
        abilita le interruzioni
        return;
    }
    else {
        aggiungi il processo alla coda S->q;
        abilita le interruzioni;
    }
}

V(semaphore *S) {
    disabilita le interruzioni;
    S->count += 1;
    if (la coda S->q non e' vuota) {
        risveglia il primo processo di S->q
    }
    abilita le interruzioni
}
```

Questa soluzione migliora quella iniziale perche' l'introduzione della coda di processi in attesa permette di risolvere i problemi del punto precedente, ad es. non presenta attesa attiva.

3. (a) Si consideri la seguente successione di riferimenti a pagine in un sistema con memoria virtuale:

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1

Si supponga di avere a disposizione tre pagine fisiche ("frame"). Si applichino i seguenti algoritmi di sostituzione delle pagine:

- i. Sostituzione delle pagine secondo l'ordine di arrivo (FIFO).
- ii. Sostituzione ottimale delle pagine (MIN, si sostituisce la pagina che non si userà per il periodo di tempo più lungo).
- iii. Sostituzione delle pagine usate meno recentemente (LRU).

In particolare, si mostri per ogni algoritmo: il contenuto nel tempo delle pagine fisiche e il numero di assenze di pagina.

Traccia di soluzione.

FIFO: 15 assenze di pagina

7	7	7	2	2	2	2	4	4	4	0	0	0	0	0	0	0	7	7	7
	0	0	0	0	3	3	3	2	2	2	2	2	1	1	1	1	1	0	0
		1	1	1	1	0	0	0	3	3	3	3	3	2	2	2	2	2	1
^	^	^	^		^	^	^	^	^	^			^	^			^	^	^

MIN: 9 assenze di pagina

7	7	7	2	2	2	2	2	2	2	2	2	2	2	2	2	2	7	7	7
	0	0	0	0	0	0	4	4	4	0	0	0	0	0	0	0	0	0	0
		1	1	1	3	3	3	3	3	3	3	3	1	1	1	1	1	1	1
^	^	^	^		^		^			^			^				^		

LRU: 12 assenze di pagina

7	7	7	2	2	2	2	4	4	4	0	0	0	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	3	3	3	3	3	3	0	0	0	0	0
		1	1	1	3	3	3	2	2	2	2	2	2	2	2	2	7	7	7
^	^	^	^		^		^	^	^	^			^		^		^		

(b) Quali degli algoritmi precedenti soffrono dell'anomalia di Belady ?

Traccia di soluzione

Solo l'algoritmo FIFO soffre dell'anomalia di Belady.

4. Considerate la seguente tabella dei segmenti:

Segmento	Base	Lunghezza
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

Si spieghi lo schema d'indirizzamento basato su segmenti.

Si calcolino gl'indirizzi fisici corrispondenti ai seguenti indirizzi logici:

(a) $\langle 0, 430 \rangle$

(b) $\langle 1, 10 \rangle$

(c) $\langle 2, 500 \rangle$

(d) $\langle 3, 400 \rangle$

(e) $\langle 4, 112 \rangle$

Traccia di soluzione.

(a) $219+430=649$

(b) $2300+10=2310$

(c) indirizzo illegale

(d) $1327+400=1727$

(e) indirizzo illegale

5. (Problema per chi non abbia sostenuto l'esame di Elementi di Architettura).

Si progetti un circuito sequenziale che funziona come contatore crescente a 3 cifre binarie.

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

Si minimizzi il numero degli stati della macchina proposta.

- (b) Si scriva la tavola delle transizioni con gli stati futuri e le uscite e la si codifichi.

- (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 (a 2, 3, o 4 ingressi). Si etichettino con chiarezza i segnali.