

Elementi di Architettura e Sistemi Operativi

Bioinformatica - Tiziano Villa

28 Settembre 2012

Nome e Cognome:

Matricola:

Posta elettronica:

| problema | punti massimi | i tuoi punti |
|------------|---------------|--------------|
| problema 1 | 6 | |
| problema 2 | 7 | |
| problema 3 | 7 | |
| problema 4 | 10 | |
| totale | 30 | |

1. Si consideri il seguente programma che crea processi invocando la chiamata di sistema `fork`.

```
#include <stdio.h>
#include <unistd.h>

int main()
{
    /* crea mediante fork un processo figlio */
    fork();

    /* crea mediante fork un altro processo figlio */
    fork();

    /* crea mediante fork un altro processo figlio */
    fork();

    return 0;
}
```

(a) Si spieghi la semantica della chiamata di sistema `fork`.

Si analizzi il codice precedente spiegandone il funzionamento.

Traccia di soluzione.

La chiamata di sistema `fork()` crea un nuovo processo figlio che avrà una copia dello spazio degli indirizzi del processo genitore. Entrambi i processi genitore e figlio continuano l'esecuzione dall'istruzione successiva alla chiamata di sistema `fork()`, con una differenza: la chiamata di sistema `fork()` restituisce il valore 0 nel processo figlio, ma restituisce l'identificatore del processo figlio (il PID diverso da 0) nel processo genitore.

(b) Considerando anche il processo padre iniziale, quanti processi sono creati dal programma precedente ?

Si argomenti la risposta e si mostri con uno schema grafico l'albero della creazione di nuovi processi.

Traccia di soluzione.

I processi sono 8 includendo quello iniziale. Il processo1 padre crea successivamente i tre processi: processo2, processo3, processo4. Il processo2 crea successivamente i due processi: processo5 e processo6. Il processo5 crea successivamente il processo7. Il processo3 crea successivamente il processo8.

2. Si consideri uno schema di traduzione da indirizzi logici a indirizzi fisici basato sull'impaginazione semplice.

Le pagine abbiano 4 bytes ciascuna. Siano date tre pagine della memoria logica, con indirizzo logico rispettivo: 0x00, 0x04 e 0x08.

La tavola delle pagine abbia tre elementi con i seguenti contenuti: l'elemento d'indice 0 ha contenuto 4 (in binario 000100), l'elemento d'indice 1 ha contenuto 3 (in binario 000011), l'elemento d'indice 2 ha contenuto 1 (in binario 000001).

- (a) Si mostri graficamente lo schema d'indirizzamento.

(b) Si calcolino gl'indirizzi delle pagine fisiche corrispondenti alle pagine logiche indicate, in particolare si calcolino:

- i. l'indirizzo fisico del primo elemento della pagina logica d'indirizzo $0x00$,
- ii. l'indirizzo fisico del primo elemento della pagina logica d'indirizzo $0x04$,
- iii. l'indirizzo fisico del primo elemento della pagina logica d'indirizzo $0x08$.

Traccia di soluzione.

- i. indirizzo logico $0x00 = 0000\ 0000_2$, dove le ultime due cifre sono lo spiazzamento 00 e le prime 6 cifre indicano l'elemento 0 della tavola delle pagine (indirizzo pagina fisica) che contiene $4_{10} = 000100_2$ (indirizzo pagina logica);
concatenando 000100 con 00 si ha $0001\ 0000_2 = 16_{10} = 10_{16}(0x10)$
indirizzo fisico
- ii. indirizzo logico $0x04 = 0000\ 0100_2$, dove le ultime due cifre sono lo spiazzamento 00 e le prime 6 cifre indicano l'elemento 1 della tavola delle pagine (indirizzo pagina fisica) che contiene $3_{10} = 000011_2$ (indirizzo pagina logica);
concatenando 000011 con 00 si ha $0000\ 1100_2 = 12_{10} = C_{16}(0x0C)$
indirizzo fisico
- iii. indirizzo logico $0x08 = 0000\ 1000_2$, dove le ultime due cifre sono lo spiazzamento 00 e le prime 6 cifre indicano l'elemento 2 della tavola delle pagine (indirizzo pagina fisica) che contiene $1_{10} = 000001_2$ (indirizzo pagina logica);
concatenando 000001 con 00 si ha $0000\ 0100_2 = 4_{10} = 4_{16}(0x04)$
indirizzo fisico

3. (a) Si descriva brevemente la gestione della memoria mediante l'impaginazione, spiegando il ruolo della tavola delle pagine e della TLB.

Traccia di soluzione.

Si vedano in particolare le sezioni 8.4 e 9.2 del libro di testo.

Nell'impaginazione si divide la memoria fisica in pagine fisiche e la memoria logica in pagine logiche (in entrambi i casi della stessa dimensione costante).

Ogni indirizzo è composto da un numero di pagina e da uno scostamento. Il numero di pagina serve come indice nella tavola delle pagine per ottenere l'indirizzo (dell'inizio) della pagina fisica nella memoria centrale. Tale indirizzo combinato con lo scostamento dà l'indirizzo fisico nella memoria centrale.

La TLB è una cache associativa che contiene una piccola parte degli elementi della tavola delle pagine. Quando il processore genera un indirizzo logico si verifica prima se esso è presente nella TLB, nel qual caso essa contiene l'indirizzo della pagina fisica, altrimenti si ha un insuccesso della TLB e si cerca nella tavola delle pagine (che è nella memoria centrale). Quando la TLB è piena bisogna sostituire qualche suo elemento con altri.

Lo schema precedente si combina nella memoria virtuale con la politica dell'impaginazione su richiesta per gestire la migrazione di pagine fisiche dal disco alla memoria centrale (e viceversa in caso di pagine riscritte). La tavola delle pagine ha un cifra binaria di validità che indica se la pagina è valida e presente in memoria centrale, o se è invalida o non presente in memoria centrale. Se l'accesso è valido ma la pagina fisica corrispondente si trova su disco, essa deve essere portata in memoria centrale, e si devono aggiornare la tavola delle pagine e la TLB. Le fasi di tale meccanismo sono spiegate nella sezione 9.2.1.

(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.

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

- Ci sono due segnali binari d'ingresso X_1X_2 e un segnale binario d'uscita Z .
- Se $X_2 = 0$, l'uscita Z nel generico istante t vale $Z(t) = X_1(t - 1)$, se $X_2 = 1$, l'uscita Z nel generico istante t vale $Z(t) = X_1(t - 2)$.

Sia 00 00 la sequenza di azzeramento (cioe', alla partenza del circuito si danno gl'ingressi 00 00).

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

Traccia di soluzione.

```
00 s00 s00 0
01 s00 s00 0
10 s00 s10 0
11 s00 s10 0
```

```
00 s01 s00 0
01 s01 s00 1
10 s01 s10 0
11 s01 s10 1
```

```
00 s10 s01 1
01 s10 s01 0
10 s10 s11 1
11 s10 s11 0
```

```
00 s11 s01 1
01 s11 s01 1
10 s11 s11 1
11 s11 s11 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.

- (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.