

Elementi di Architettura e Sistemi Operativi

Bioinformatica - Tiziano Villa

1 Luglio 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. Processi (in inglese "processes") e fili esecutivi (in inglese "threads").

- (a) Che cos'è un filo esecutivo ?

Traccia di soluzione.

È un percorso di controllo, cioè l'unità autonoma di esecuzione del processore, e comprende un identificatore, un contatore di programma, un insieme di registri, e una pila.

Si vedano le sezioni 4.1.1 e 4.1.2.

- (b) Si descrivano brevemente le differenze tra i processi a singolo filo esecutivo (in inglese "single-threaded process") e i processi a più fili esecutivi (in inglese "multi-threaded process") e s'illustrino i vantaggi del secondo modello.

Traccia di soluzione.

Un processo tradizionale, chiamato anche processo pesante, è composto da un solo filo esecutivo. Un processo a più fili esecutivi può lavorare a più compiti in modo concorrente; i fili esecutivi che appartengono allo stesso processo condividono parte delle risorse (come la sezione del codice, la sezione dei dati e altre risorse del sistema, come i file aperti e i segnali), mentre altre sono private dei singoli fili esecutivi.

I vantaggi del secondo modello riguardano il tempo di risposta, la condivisione ed economia delle risorse, la scalabilità.

Si vedano le sezioni 4.1.1 e 4.1.2. La Fig. 4.1 riassume il tutto con evidenza grafica.

- (c) Quali risorse sono condivise e quali sono private nell'ambito dei fili esecutivi di un processo a più fili esecutivi ?

Traccia di soluzione.

Risorse condivise: sezione del codice, la sezione dei dati e altre risorse del sistema, come i file aperti e i segnali.

Risorse private: registri del processore (incluso contatore di programma e puntatore alla pila), contenuto della pila.

- (d) Che cosa si deve salvare quando si opera una commutazione di contesto tra fili esecutivi (in inglese "context switching") ?

Traccia di soluzione.

I registri del processore (incluso il contatore di programma e il puntatore alla pila), contenuto della pila.

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 consideri il seguente scenario di memoria organizzata a pagine:

- (a) lunghezza indirizzo logico = 4B;
- (b) dimensione di una pagina = 4 KB;
- (c) dimensione di un elemento della tavola delle pagine = 4 B;
- (d) una tavola delle pagine occupa esattamente una pagina di memoria.

Si mostri il meccanismo di traduzione degli indirizzi logici in indirizzi fisici, definendo la divisione in campi dell'indirizzo logico (specificando la lunghezza e il significato di ogni campo). Quanti livelli d'impaginazione sono necessari ?

Traccia di soluzione.

Una tavola delle pagine ha dimensione 4 KB e quindi contiene $1K = 1024 = 2^{10}$ elementi (4 KB / 4 B), ciascuno dei quali punta a una pagina, per un totale di 2^{10} pagine puntate, che indirizzano $2^{10} \times 2^{12} = 2^{22}$ B di memoria.

Aggiungendo un altro livello d'indirizzamento s'introducono altre 2^{10} pagine puntate, ciascuna delle quali e' una tavola delle pagine di secondo livello contenente 2^{10} elementi, e s'indirizzano in totale $2^{10} \times 2^{10} \times 2^{12} = 2^{32}$ B di memoria.

Perciò servono due livelli d'indirizzamento, e l'indirizzo logico di 32 cifre binarie e' organizzato come segue:

- (a) un primo campo di 10 cifre binarie che seleziona un elemento della tavola delle pagine di primo livello il quale punta a una pagina che e' una tavola delle pagine di secondo livello;
- (b) un secondo campo di 10 cifre binarie che seleziona un elemento della tavola delle pagine di secondo livello (selezionata nel passo precedente) il quale punta a una pagina fisica;
- (c) un terzo campo di 12 cifre binarie che seleziona una parola di memoria nella pagine fisica selezionata nel passo precedente.

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 progetti un circuito combinatorio che realizza la seguente variabile binaria d'uscita F in funzione delle variabili binarie d'ingresso A, B, C, D, E :

$$F = (AB) \rightarrow ((CD) \rightarrow E),$$

dove \rightarrow e' l'operatore logico d'implicazione.

- (a) Si scriva la tavola di verita' dell'operatore d'implicazione logica.

Traccia di soluzione.

$$F \equiv X \rightarrow Y$$

X	Y	F
0	0	1
0	1	1
1	0	0
1	1	1

Da cui $X \rightarrow Y$ equivale a $\overline{X} + Y$.

(b) Si scriva la tavola di verità della funzione F .

Traccia di soluzione.

Per semplicità si sostituisce nella F l'implicazione con la sua disgiunzione equivalente ($X \rightarrow Y$ equivale a $\overline{X} + Y$) ottenendosi $F = \overline{A} + \overline{B} + \overline{C} + \overline{D} + E$, che è sempre vero tranne quando $A = B = C = D = 1, E = 0$. Si noti che è equivalente alle forme $F = (AB) \rightarrow (C \rightarrow (D \rightarrow E))$.

- (c) Si minimizzi la rappresentazione a due livelli (somma di prodotti) della funzione F usando il metodo delle mappe di Karnaugh.

Traccia di soluzione.

Applicando il metodo delle mappe di Karnaugh si ottiene $F = \overline{A} + \overline{B} + \overline{C} + \overline{D} + E$.

Una scorciatoia sarebbe negare l'unico termine dove la funzione vale 0, cioè $\overline{F} = ABCDE$ e quindi $F = \overline{A} + \overline{B} + \overline{C} + \overline{D} + E$.

Si sottolinea che la domanda richiede d'impostare e calcolare correttamente il minimo con il metodo delle mappe di Karnaugh.

- (d) Si realizzi il circuito combinatorio corrispondente con porte NAND. Si etichettino con chiarezza i segnali.

Traccia di soluzione.

Conviene rappresentare la negazione della negazione della funzione F . La negazione della funzione F si esprime naturalmente con una porta AND con gl'ingressi A, B, C, D, \overline{E} , e negata diventa una porta NAND.