

# Elementi di Architettura e Sistemi Operativi

Bioinformatica - Tiziano Villa

19 Settembre 2011

Nome e Cognome:

Matricola:

Posta elettronica:

problema	punti massimi	i tuoi punti
problema 1	8	
problema 2	6	
problema 3	8	
problema 4	8	
totale	30	

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

(a) 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.

(b) Perché certi sistemi memorizzano il sistema operativo nel firmware, altri sul disco ?

Traccia di soluzione.

Vedi Sec. 2.11

Alcuni sistemi, come i telefoni cellulari, i palmari (PDA - personal digital assistant) e le console per videogiochi, memorizzano l'intero sistema operativo nella ROM. La scelta di memorizzare nella ROM il sistema operativo è indicata per sistemi semplici (sia come hardware che come software) e di piccole dimensioni (ad es. per certi dispositivi come i palmari o i telefoni cellulari può darsi che non sia disponibile un disco con un file system). In questo caso il sistema operativo deve essere memorizzato nel firmware.

Un limite di questa soluzione è la necessità di modificare la ROM per modificare il codice del programma di avvio. Una soluzione è utilizzare una ROM a sola lettura programmabile e cancellabile (EPROM).

Per sistemi operativi di grandi dimensioni, il caricatore di avvio è memorizzato nel firmware e il sistema operativo risiede su disco. In tal caso, il codice di avviamento e il sistema operativo stesso possono essere modificati facilmente scrivendone nuove versioni sul disco. Una volta caricato il programma di avvio, esso localizza nel file system il sistema operativo e lo carica in memoria per eseguirlo.

(c) Si descriva la procedura di avvio del sistema.

Traccia di soluzione.

Vedi Sec. 2.10

All'avviamento si esegue un piccolo programma ("bootstrap program") (di qualche centinaio di byte) che risiede in una ROM e poi carica un sistema operativo nella memoria, di solito da un disco rigido. L'esecuzione di tale programma di avvio prevede una serie di controlli su quali unita' (come il disco rigido) siano utilizzabili, poi carica il sistema operativo cui passa il controllo della macchina.

Per sistemi operativi di grandi dimensioni il programma di avvio risiede solo in parte in una ROM (tale parte conosce la lunghezza e l'indirizzo sul disco del codice residuo del programma d'avvio), mentre il resto e il sistema operativo risiedono nel disco. Un disco che contenga una partizione di avvio si chiama disco di avvio ("boot disc") o disco di sistema.

- (d) Come si potrebbe progettare un sistema in modo da permettere la scelta di quale sistema operativo far partire (tra alcuni messi a disposizione) ?  
Che cosa dovrebbe fare il programma d'avviamento ?

Traccia di soluzione.

Si consideri un sistema su cui si volesse far girare sia Windows XP sia quattro distribuzioni diverse di Linux (es., RedHat, Debian, Mandrake e Ubuntu). Ciascun sistema operativo sarà memorizzato su disco. Al momento d'avvio uno speciale programma di gestione dell'avviamento (boot manager) determinerà quale sistema operativo si vuol far partire. Questo vuol dire che invece di far partire subito un certo sistema operativo, al momento d'avvio del sistema si eseguirà per primo il gestore dell'avviamento, che è responsabile di quale sistema operativo far partire. Tipicamente i gestori dell'avviamento devono essere memorizzati in locazioni fisse del disco rigido per essere riconosciuti quando si fa partire il sistema. I gestori dell'avviamento di solito offrono all'utente il menu dei sistemi operativi tra cui scegliere e hanno un caso di default se non si esprime una scelta.

2. Si consideri il seguente codice per risolvere il problema dei lettori e scrittori mediante semafori (*OKToRead*, *OKToWrite*, *Mutex*) e variabili di stato (*AR* numero lettori attivi, *WR* numero lettori in attesa, *AW* numero scrittori attivi, *WW* numero scrittori in attesa). Le specifiche sono:

- Più lettori possono accedere al codice contemporaneamente, ma gli scrittori devono avere accesso esclusivo.
- I lettori possono procedere solo se non ci sono scrittori attivi o in attesa.
- Gli scrittori possono procedere solo se non ci sono lettori attivi o scrittori attivi o in attesa.
- Solo un processo alla volta manipola le variabili di stato con il semaforo *Mutex*.

Si completino le operazioni di tipo *P* e *V* con gli argomenti mancanti.

Si spieghi la soluzione proposta.

```
Semaphore OKToRead = 0; OKToWrite = 0; Mutex = 1;
AR = WR = AW = WW = 0;
```

```
Lettore {
    P(?);
    if ((AW + WW) == 0) {
        V(?);
        AR = AR + 1;
    } else WR = WR + 1;
    V(?);
    P(?);

    leggere i dati;

    P(?);
    AR = AR - 1;
    if (AR == 0 && WW > 0) {
        V(?);
        AW = AW + 1;
        WW = WW - 1;
    }
}
```

```

    V(?);
}

Scrittore {
    P(?);
    if ((AW + AR + WW) == 0) {
        V(?);
        AW = AW + 1;
    } else WW = WW + 1;
    V(?);
    P(?);

    scrivere i dati;

    P(?);
    AW = AW - 1;
    if (WW > 0) {
        V(?);
        AW = AW + 1;
        WW = WW - 1;
    } else while (WR > 0) {
        V(?);
        AR = AR + 1;
        WR = WR - 1;
    }
    V(?);
}

```

## Traccia di soluzione.

```
Semaphore OKToRead = 0; OKToWrite = 0; Mutex = 1;  
AR = WR = AW = WW = 0;
```

```
Lettoore {  
    P(Mutex);  
    if ((AW + WW) == 0) {  
        V(OKToRead);  
        AR = AR + 1;  
    } else WR = WR + 1;  
    V(Mutex);  
    P(OKToRead);  
  
    leggere i dati;  
  
    P(Mutex);  
    AR = AR - 1;  
    if (AR == 0 && WW > 0) {  
        V(OKToWrite);  
        AW = AW + 1;  
        WW = WW - 1;  
    }  
    V(Mutex);  
}
```

```

Scrittore {
    P(Mutex);
    if ((AW + AR + WW) == 0) {
        V(OKToWrite);
        AW = AW + 1;
    } else WW = WW + 1;
    V(Mutex);
    P(OKToWrite);

    scrivere i dati;

    P(Mutex);
    AW = AW - 1;
    if (WW > 0) {
        V(OKToWrite);
        AW = AW + 1;
        WW = WW - 1;
    } else while (WR > 0) {
        V(OKToRead);
        AR = AR + 1;
        WR = WR - 1;
    }
    V(Mutex);
}

```

3. Si consideri uno schema d'indirizzamento a due livelli basato su pagine per indirizzi di memoria virtuale a 24 cifre binarie, con il seguente formato:

cifre 23-16	cifre 15-8	cifre 7-0
# pagina virtuale	# pagina virtuale	spiazzamento

Gl'indirizzi virtuali sono tradotti in indirizzi fisici di 16 cifre binarie, con il seguente formato:

cifre 15-8	cifre 7-0
# pagina fisica	spiazzamento

Gli elementi delle tavole delle pagine hanno 16 cifre binarie, con il seguente formato:

cifre 15-8	cifre 7 6 5 4 3 2 1 0
# pagina fisica	informazione varia

dove l'informazione varia e' definita come segue:

**posizione 7** riservato sistema operativo

**posizione 6** irraggiungibile

**posizione 5** 0

**posizione 4** 0

**posizione 3** modificato in scrittura

**posizione 2** usato

**posizione 1** accessibile in scrittura

**posizione 0** valido

Una traduzione da virtuale a fisico puo' fallire per uno di due motivi: pagina invalida (la pagina non e' presente in memoria fisica), oppure violazione d'accesso (la pagina e' presente in memoria fisica, ma l'accesso e' illegale).

- (a) Spiegate testualmente e illustrate con un disegno lo schema proposto di risoluzione degli indirizzi da virtuale a fisico.
- (b) Spiegate il meccanismo di "copia su scrittura" ("copy on write").

Traccia di soluzione.

Si veda la Sez. 9.3 del libro di testo.

L'idea è che processi genitori e figli inizialmente condividano le pagine. Se un processo genitore o figlio scrive su una pagina condivisa il sistema crea una copia di tale pagina.

(c) Si consideri il contenuto della memoria fisica nello schema allegato. Si assuma che il puntatore alla tavola di primo livello (Table Base Pointer) punti all'indirizzo  $0x002000$ .

Si calcolino i valori prodotti dalle seguenti istruzioni di lettura (Load) e scrittura (Store) eseguite in modalita' utente.

- i. Store[0x010310],
- ii. Load[0x020102],
- iii. Store[0x020731].

Gl'indirizzi indicati sono virtuali e devono essere tradotti in indirizzi fisici.

Il valore prodotto da una lettura e' un dato di 8 cifre binarie, se la lettura e' andata a buon fine, o una segnalazione d'errore. Il valore prodotto da una scrittura e' un SI, se la lettura e' andata a buon fine, o una segnalazione d'errore.

Nel caso di errore se ne specifichi il tipo: pagina invalida oppure violazione d'accesso (per una scrittura senza il privilegio per scrivere, o per una lettura/scrittura da pagina riservata al sistema operativo).

Si mostrino in dettagli tutti i passi del meccanismo di traduzione degli indirizzi per arrivare alla risposta.

Traccia di soluzione.

- i. Store[0x010310] produce SI (si scrive nella locazione di memoria  $0x0010$  contenente  $0x81$ ),
- ii. Load[0x020102] produce  $0x01$  (si legge dalla locazione di memoria  $0x0002$  contenente  $0x01$ ),
- iii. Store[0x020731] produce segnalazione d'errore: violazione d'accesso per una scrittura senza il privilegio di scrivere.

Nota 1: ogni elemento della tavola occupa 2 bytes di memoria, percio' l'indice  $i$  punta alle posizioni di memoria  $2i$  e  $2i+1$  a partire dall'indirizzo di base (nel primo passo dato dall'inizio della tavola); si veda la figura allegata.

Nota 2: l'indirizzo fisico ottenuto indica la casella (byte) di memoria. Ad es. nel caso di Load[0x020102] alla fine della conversione da indirizzo virtuale a indirizzo fisico si perviene all'indirizzo fisico  $0x0002$  che e' la terza casella di memoria contenente  $0x01$ .

4. Si consideri la seguente rete logica combinatoria a piu' livelli:

$$\begin{aligned}V_1 &= cd' + b' \\V_2 &= V_1d + a'c' + b' \\V_3 &= V_2' + ab \\Y &= cd' + V_1d \\Z &= c'a' + V_3' + cd \\y &= Y \\z &= Z\end{aligned}$$

dove  $a, b, c, d$  sono gl'ingressi,  $y, z$  sono le uscite.

(a) Si realizzi tale rete combinatoria con una matrice logica programmabile del tipo PLA, supponendo di avere a disposizione tutti i termini prodotto necessari. Si indichino esplicitamente i termini prodotto del piano AND e le espressioni del piano OR e si disegni lo schematico della PLA con le interconnessioni da programmare.

Traccia di soluzione.

I termini prodotto sono:

$$\begin{aligned}P_1 &= cd' \\P_2 &= b' \\P_3 &= V_1d \\P_4 &= a'c' \\P_5 &= V_2' \\P_6 &= ab \\P_7 &= V_3' \\P_8 &= cd\end{aligned}$$

Le disgiunzioni sono:

$$\begin{aligned}V_1 &= P_1 + P_2 \\V_2 &= P_3 + P_4 + P_2 \\V_3 &= P_5 + P_6 \\Y &= P_1 + P_3 \\Z &= P_4 + P_7 + P_8\end{aligned}$$

Si noti che i termini  $V_1$ ,  $V_2$  e  $V_3$  sono retroazionati per realizzare una funzione a piu' livelli con logica a due livelli.

- (b) Si realizzi tale rete combinatoria con una matrice logica programmabile del tipo PAL, supponendo di avere a disposizione tutti i termini prodotto necessari e che il piano OR sia costituito solo da OR a 2 ingressi. S'indichino esplicitamente i termini prodotto del piano AND e le espressioni del piano OR e si disegni lo schematico della PAL con le interconnessioni da programmare.

Traccia di soluzione.

I termini prodotto sono:

$$P_1 = cd'$$

$$P_2 = b'$$

$$P_3 = V_1d$$

$$P_4 = a'c'$$

$$P_5 = b'$$

$$P_6 = V_2'$$

$$P_7 = ab$$

$$P_8 = cd'$$

$$P_9 = V_1d$$

$$P_{10} = a'c'$$

$$P_{11} = V_3'$$

$$P_{12} = cd$$

$$P_{13} = V_{21}$$

$$P_{14} = Z_1$$

Le disgiunzioni sono:

$$V_1 = P_1 + P_2$$

$$V_{21} = P_3 + P_4$$

$$V_2 = P_{13} + P_5$$

$$V_3 = P_6 + P_7$$

$$Y = P_8 + P_9$$

$$Z_1 = P_{10} + P_{11}$$

$$Z = P_{14} + P_{12}$$

Si noti che i termini  $V_1, V_{21}, V_2, V_3, Z_1$  sono retroazionati per realizzare una funzione a piu' livelli con logica a due livelli.