Secondo Compitino di Architetture 1, prof. Chiola

2/6/1995

Cognome :

Nome :

Matricola :

Regole Generali

Questo compito é composto da 2 diversi tipi di domande: alcune a risposte multiple, altre in cui dovete dare voi la risposta corretta mediante lo svolgimento di un semplice esercizio. Nel secondo tipo di domande si richiede di riportare solo il risultato finale.
Le risposte vanno riportate sul MODULO DI RISPOSTA distribuito a parte. Si consiglia comunque di riportare le risposte anche all'interno del testo in modo che il candidato possa tenere una copia dopo la consegna per la verifica delle risposte ed il calcolo del punteggio. Il calcolo dei punteggi relativi ai due tipi di domande é spiegato più avanti.
NON USARE FOGLI ADDIZIONALI né per la consegna del compito né per la "brutta"! E' ammesso utilizzare il rovescio dei fogli del testo come "brutta copia" (che in ogni caso non sarà presa in considerazione per la correzione).
Usare solo penne e NON MATITE per la compilazione delle risposte, in quanto NON VERRANNO ACCETTATE CONTESTAZIONI DI NESSUN TIPO riguardo eventuali correzioni o cancellazioni (comprese quelle effettuate col bianchetto).
Non é ammessa nessuna forma di comunicazione tra i candidati. Non sono ammessi "prestiti" di materiale di nessun tipo tra i candidati.
E' possibile assentarsi dall'aula solo previo deposito del testo del compito presso il docente "di guardia". Anche in caso di ritiro anticipato dal compito il testo dell'esercizio NON PUÒ essere portato fuori dall'aula prima dello scadere del termine di consegna.

Compilazione delle risposte

Si consiglia di scrivere le risposte inizialmente sul testo e di utilizzare gli ultimi 5–10 minuti per ricopiare le risposte sull'apposito foglio da consegnare.

Domande a risposte multiple

Rispondere segnando sul testo con una crocetta le risposte che ritenete corrette e lasciando in bianco quelle che ritenete errate. Ad ogni domanda corrisponde almeno una risposta corretta (ci possono essere più risposte corrette per una domanda).
Domande di questo tipo hanno associati due valori di punteggio: uno negativo e uno positivo. Ogni risposta errata segnata con una crocetta comporta la detrazione del valore negativo dal punteggio complessivo accumulato. Il valore positivo corrisponde al punteggio globalmente assegnato a tutte le risposte corrette ad una domanda.
Per esempio, se una domanda etichettata coi valori $(-1, 4)$ prevede 4 risposte possibili di cui **a. d.** errate e **b. c.** corrette, allora se il candidato segna le risposte **b. c.** e non segna le risposte **a. d.** ottiene il massimo punteggio 4, mentre se segna le risposte **a. b.** e non **c. d.** allora ottiene il punteggio $-1+2=1$; nel caso segnasse tutte le risposte otterrebbe in questo esempio il punteggio $-1+2+2-1=2$. Notare

che in generale, CONTRARIAMENTE a quanto succede nell'esempio, la somma algebrica dei punteggi ottenuti segnando tutte le risposte é ≤ 0!

Esercizi

In questo caso é il candidato che deve calcolare e scrivere la risposta finale. Ogni domanda é accompagnata da un solo valore positivo che indica il punteggio massimo assegnato per la risposta corretta a tutti i quesiti. In caso di risposte parziali o parzialmente corrette il punteggio viene ridotto in proporzionalità diretta rispetto al numero di risposte corrette (con troncamento al più grande numero intero minore o uguale nel caso di risultato frazionario).

Tattica suggerita

Come nel caso di altri test da completarsi in tempi predefiniti, risulta normalmente conveniente completare il compito in due passate: in una prima fase dedicare una quantità predefinita e limitata di tempo alla lettura di tutte le domande, svolgendo solo gli esercizi più semplici. In un secondo passo (dopo aver risposto rapidamente a tutte le domande semplici) tornate a considerare quelle più difficili in ordine decrescente di punteggio. In questo modo riuscirete a dedicare la maggior quantità di tempo alla soluzione degli esercizi difficili ma più redditizi. Per le domande a risposte multiple con punteggi negativi evitate di segnare risposte di cui non siete assolutamente certi in quanto i valori dei punteggi sono studiati per penalizzare i "tentativi a caso". Per gli esercizi senza punteggi negativi invece le risposte sbagliate non vi penalizzeranno.

Calcolo del punteggio

Il voto del secondo compito espresso in quindicesimi sarà calcolato applicando la seguente formula, nella quale s rappresenta la somma dei punti totalizzati per le varie domande:

$$\text{Voto} = c1 + \min(s, c2) \times c3 + (s - c2) \times c4$$

I coefficienti $c1$, $c2$, $c3$ e $c4$ sono valori costanti che verranno determinati a posteriori tenendo conto della statistica dei valori s ottenuti dai vari candidati che avranno consegnato in modo da ottenere una distribuzione equa dei voti tra 8/15 e 16/15 per quelli che avranno risposto correttamente ad una porzione sufficiente di domande (questo a titolo di regola transitoria, in macanza di dati storici sul grado di difficoltà dell'esame; per gli scritti successivi i coefficienti saranno sempre calcolati con criteri statistici ma sulla base dei risultati degli scritti precedenti, e quindi resi noti a priori; l'unica condizione garantita a priori per questo secondo compito e' che le costanti $c2$ $c3$ e $c4$ saranno maggiori di 0 e che $c3 > c4$, in modo da garantire voti monotonicamente crescenti al crescere di s con una saturazione rispetto alla soglia $s = c2$). Il voto finale in trentesimi della prova scritta verrà ottenuto sommando il voto in quindicesimi del primo e del secondo compitino e arrotondando il risultato (a patto che entrambi i compiti siano sufficienti). Punteggi complessivi superiori a 30 verranno poi considerati equivalenti alla "lode". Il voto finale verrà poi stabilito a partire dal voto dello scritto con un riaggiustamento di tale valore fino ad una variazione massima di 3/30 in più o in meno, a seconda dell'esito della prova orale.

Domande

1. (punti: -1, 4)
Il registro Frame Pointer dell'architettura SPARC:

   (a) é destinato a contenere un valore costante fino al termine dell'esecuzione della procedura o funzione in corso;

   (b) é un registro globale dell'architettura contenente una costante predefinita (come "`g0`");

   (c) é un nome diverso per accedere al registro Stack Pointer (un "alias" per SP, come "`g2`" é un alias per "`r2`");

   (d) é un nome diverso per accedere al registro "`r30`";

   (e) contiene l'indirizzo (fisico) della tabella di paginazione per la definizione degli indirizzi virtuali del programma in esecuzione;

   (f) nessuna delle affermazioni precedenti é corretta.

2. (punti: -1, 4)
Una codifica ad espansione su 4, 8 e 12 bit (ossia con codifica base su 4 bit e possibilità di due livelli di espansione, ciascuno con aggiunta di 4 bit):

   (a) consente di rappresentare $2^4 + 2^8 + 2^{12} = 2320$ configurazioni diverse;

   (b) consente di rappresentare $2^4 + 2^4 + 2^4 = 48$ configurazioni diverse;

   (c) consente di rappresentare $2^3 + 2^3 + 2^4 = 32$ configurazioni diverse;

   (d) permette di codificare su 12 bit 32 istruzioni con operando codificato su 4 bit (oltre al codice operativo);

   (e) permette di codificare su 16 bit 20 istruzioni con operando codificato su 8 bit (oltre al codice operativo);

   (f) nessuna delle affermazioni precedenti é corretta.

3. (punti: -1, 4)
Data una architettura RISC con organizzazione a Pipeline in 5 stadi non superscalare e frequenza di clock di 75 MHz.:

   (a) il tempo di esecuzione di ogni istruzione é di 13.33 ns;

   (b) la velocità massima di completamento delle istruzioni é di 75 MIPS;

   (c) nel caso di programmi puramente sequenziali (senza istruzioni di salto) consente di eseguire "in parallelo" fasi diverse di 5 istruzioni consecutive, quintuplicando la velocità rispetto alla frequenza di clock (375 MIPS con clock a 75 MHz.);

   (d) la velocità minima di completamento delle istruzioni é di 15 MIPS (75 MHz/5 stadi), a meno di ulteriori rallentamenti dovuti alla memoria;

   (e) richiede una memoria RAM con tempo di accesso di non oltre 13.33 ns (per consentire il completamento di un fetch per ogni periodo di clock);

   (f) nessuna delle affermazioni precedenti é corretta.

4. (punti: -1, 4)
La microarchitettura MIC-2 descritta sul libro di Tanenbaum:

(a) é preferibile rispetto a MIC-1 per la realizzazione di MAC-2 perché permette di economizzare sul numero di parole della Control Store a parità di istruzioni della macchina convenzionale;

(b) é preferibile rispetto a MIC-1 per la realizzazione di MAC-2 perché permette di economizzare sul dimensione della Control Store a parità di velocità di esecuzione delle istruzioni della macchina convenzionale;

(c) é preferibile rispetto a MIC-1 per la realizzazione di MAC-2 perché permette di semplificare l'insieme di istruzioni della macchina convenzionale;

(d) non é competitiva rispetto a MIC-1 in termini di prestazioni a causa dell'introduzione del metodo di codifica delle microistruzioni che richiede una ulteriore fase di decodifica a livello firmware;

(e) non é utilizzabile per la realizzazione di una macchina convenzionale con insieme di istruzioni ridotto a causa della sua organizzazione a microarchitettura verticale;

(f) nessuna delle affermazioni precedenti é corretta.

5. (punti: 5)
La connessione di una memoria cache di 16 KByte organizzata in linee di 4 parole ad un sistema con parole di memoria da 32 bit e indirizzamento a livello di byte espresso su 24 bit, al variare del livello di associatività richiede la presenza dei seguenti numeri di bit per il campo "tag":

associatività	N. bit tag
corrispondenza dir.	
2 insiemi	
4 insiemi	
8 insiemi	
completamente assoc.	

6.
Considerare un sistema di traduzione da indirizzamento virtuale a indirizzamento fisico realizzato mediante la tecnica di paginazione e segmentazione. Ogni indirizzo logico può far riferimento ad un massimo di 4 segmenti. Ciascun segmento può essere costituito da un massimo di 32 parole di memoria, suddivise in pagine da 8 parole. La memoria fisica é costituita da 64 parole.

(a) (punti: 2)
Da quanti bit sono costituiti gli indirizzi virtuali e gli indirizzi fisici?

(b) (punti: 6)

Date le seguenti tabelle di traduzione:

tab. segmenti

n. segm	n. pag.	tab. pag.
0	3	tab0
1	4	tab1
2	1	tab2
3	0	—

tab0

n.pag.log.	n.pag.fis
0	2
1	1
2	7

tab1

n.pag.log.	n.pag.fis
0	5
1	3
2	4
3	6

tab2

n.pag.log.	n.pag.fis
0	0

Calcolare i valori $n1$ e $n2$ rispettivamente come numero d'ordine della prima lettera del vostro cognome, e come somma dei numeri d'ordine delle prime lettere dei vostri nome e cognome modulo 22. Interpretando $n1$ ed $n2$ come indirizzi virtuali, tradurli in indirizzi fisici.

7. (punti: 10)
Considerare un programma in assembler SPARC per il trattamento di dati di tipo "stringhe di caratteri di lunghezza ≤ 9", memorizzate sotto forma di vettori di 10 byte consecutivi (con codifica ASCII) e terminate da un byte di valore 0.

In particolare, una stringa individuata dall'indirizzo simbolico "`cn`" verrà inizializzata dal programma principale con la sequenza di caratteri (minuscoli) costituenti il vostro cognome seguito (senza spazi intermedi) dal vostro nome (troncando dal 10^o carattere in poi se la stringa così ottenuta supera i 9 caratteri).

L'indirizzo simbolico "`sn`" rappresenta un vettore di 5 stringhe consecutive (ciascuna espressa su 10 byte) inizializzate con i nomi dei giorni lavorativi della settimana da "`lunedi`" (elemento 0 del vettore) a "`venerdi`" (elemento 4 del vettore).

Il programma é costituito dalla procedura "`proc`" che richiama la funzione "`funz`", la quale ritorna un carattere nel registro "`%o0`" alla procedura chiamante:

```
proc:  save %sp,-104,%sp
       mov 0,%l2
       sethi %hi(cn),%l0  ! %hi(x) indica i bit piu' signif. della rappr. bin. di x
       sethi %hi(sn),%l1
       or %l1,%lo(sn),%l1 ! %lo(x) indica i bit meno signif. della rappr. bin. di x
.LL15: or %l0,%lo(cn),%o0
       call funz
       mov %l1,%o1
       cmp %o0,0
       bne,a .LL14
       call stampacarattere ! stampa il car. il cui codice ASCII si trova in %o0
       call stampaintero  ! stampa il valore che si ritrova in %o0
       mov %l2,%o0
.LL14: add %l2,1,%l2
       cmp %l2,4
       ble,a .LL15
       add %l1,10,%l1
       ret
       restore
```

La funzione "`funz`" é terminale (quindi lavora nel contesto della procedura chiamante), e riceve in ingresso 2 parametri nei registri `%o0, %o1`; gli altri registri `%o` sono usati come variabili locali:

```
funz:  mov %o0,%o3
.LL5:  ldub [%o3],%o0
       cmp %o0,0
       be .LL3
       mov %o1,%o2
.LL4:  ldub [%o2],%o4
       cmp %o4,0
       be,a .LL5
```

```
        add %o3,1,%o3
        cmp %o4,%o0
        bne,a .LL4
        add %o2,1,%o2
.LL3:   retl
        nop
```

scrivere i valori interi e/o i caratteri stampati dall'esecuzione della procedura `proc` sui dati sopra descritti (nell'ordine seguendo l'esecuzione del programma).

8. (punti: 12)
   Il seguente codice assembler MAC-1 rappresenta una possibile traduzione della procedura `proc` in assembler SPARC della domanda precedente:

```
proc:  LOCO sn
I1:
       LOCO 0
       PUSH
ciclo: LOCO cn
       PUSH
I2:
       PUSH
       CALL funz
       INSP 1
       STOL 0
       JZER qua
       CALL stampacarattere
       JUMP la
qua:   LODL 1
       STOL 0
       CALL stampaintero
la:    INSP 1
       LOCO 4
       SUBL 0
       JPOS sotto
I3:
       RETN
sotto: LOCO 1
       ADDL 0
       STOL 0
       LOCO 10
I4:
       STOL 1
       JUMP ciclo
```

   In questo caso le stringhe sono rappresentate da 10 parole consecutive (da 16 bit) contenenti la codifica ASCII dei caratteri. La funzione "`funz`" ritorna il valore intero nel registro accumulatore. Completare il codice con le 4 istruzioni mancanti.

Risposte Compitino di Architetture 1, 2/6/1995

Cognome : **Nome:** **Matricola:**

1. (punti: -1, 4) **X a** **b** **c** **X d** **e** **f**
2. (punti: -1, 4) **a** **b** **c** **d** **X e** **f**
3. (punti: -1, 4) **a** **X b** **c** **X d** **e** **f**
4. (punti: -1, 4) **a** **b** **c** **d** **e** **X f**
5. (punti: 5)

associatività	N. bit tag
corrispondenza dir.	10
2 insiemi	11
4 insiemi	12
8 insiemi	13
completamente assoc.	20

6.
   (a) (punti: 2) 7 bit indir. virtuale, 6 bit indir. fisico

   (b) (punti: 6)
   I numeri da 0 a 7 vengono incrementati di 16;
   I numeri da 8 a 15 rimangono invariati;
   I numeri da 16 a 21 vengono incrementati di 40.

   Esempio: Chiola Giovanni $\rightarrow$ $n1 = 3$, $n2 = 10$

	n virtuale	n fisico
$n1 =$	3	19
$n2 =$	10	10

7. (punti: 10)
   Per ogni giorno della settimana identifica e stampa il primo carattere (da sinistra verso destra) di "cognome-nome" uguale ad un carattere del nome del giorno; se le stringhe non hanno caratteri in comune, stampa il numero associato al giorno della settimana.

   Esempio: "chiolagio"

	intero/carattere
stampa 1	i
stampa 2	i
stampa 3	c
stampa 4	i
stampa 5	i

8. (punti: 12)

```
proc:  LOCO sn
I1:    PUSH
       LOCO 0
       PUSH
ciclo: LOCO cn
       PUSH
I2:    LODL 2
       PUSH
       CALL funz
       INSP 1
       STOL 0
       JZER qua
       CALL stampacarattere
       JUMP la
qua:   LODL 1
       STOL 0
       CALL stampaintero
la:    INSP 1
       LOCO 4
       SUBL 0
       JPOS sotto
I3:    INSP 2
       RETN
sotto: LOCO 1
       ADDL 0
       STOL 0
       LOCO 10
I4:    ADDL 1
       STOL 1
       JUMP ciclo
```

Risposte Compitino di Architetture 1, 2/6/1995

Cognome : **Nome:** **Matricola:**

1. (punti: -1, 4) **a** **b** **c** **d** **e** **f**
2. (punti: -1, 4) **a** **b** **c** **d** **e** **f**
3. (punti: -1, 4) **a** **b** **c** **d** **e** **f**
4. (punti: -1, 4) **a** **b** **c** **d** **e** **f**
5. (punti: 5)

associatività	N. bit tag
corrispondenza dir.	
2 insiemi	
4 insiemi	
8 insiemi	
completamente assoc.	

6.

(a) (punti: 2) bit indir. virtuale, bit indir. fisico

(b) (punti: 6)

	n virtuale	n fisico
$n1 =$		
$n2 =$		

7. (punti: 10)

	intero/carattere
stampa 1	
stampa 2	
stampa 3	
stampa 4	
stampa 5	

8. (punti: 12)

```
proc:  LOCO sn
I1:
       LOCO 0
       PUSH
ciclo: LOCO cn
       PUSH
I2:
       PUSH
       ...
       JPOS sotto
I3:
       RETN
       ...
       LOCO 10
I4:
       STOL 1
       JUMP ciclo
```