

Sistemi di Calcolo (A.A. 2017-2018)

Corso di Laurea in Ingegneria Informatica e Automatica
Sapienza Università di Roma

A

Compito di esonero (9/11/2017) – Durata 1h 30'

Inserire nome, cognome e matricola nel file `studente.txt`.

Parte 1 (programmazione IA32)

Nella directory `es1A`, si traduca in assembly IA32 la seguente funzione C scrivendo un modulo `es1A.s`:

```
void update(short* v, unsigned n) {
    int i;
    for (i=0; i<n; i++)
        if (v[i] < 0) assoluto(&v[i]);
}
```

L'unico criterio di valutazione è la correttezza, cioè l'equivalenza semantica tra il programma tradotto e quello C di partenza. Generare un file eseguibile `es1A` con `gcc -m32 -g`. Per i test, compilare il programma insieme al programma di prova `es1A-main.c` e al modulo `assoluto.s`.

Nota: non modificare in alcun modo `es1A-main.c` e `assoluto.s`.

Parte 2 (programmazione IA32)

Nella directory `es2A`, si traduca in assembly IA32 la seguente funzione C scrivendo un modulo `es2A.s`:

```
int get(int* x) {
    int z;
    load(&z);
    return x[0] <= z && z <= x[1] && !(x[0]<0);
}
```

L'unico criterio di valutazione è la correttezza, cioè l'equivalenza semantica tra il programma tradotto e quello C di partenza. Generare un file eseguibile `es2A` con `gcc -m32 -g`. Per i test, compilare il programma insieme al programma di prova `es2A-main.c`, e al modulo `load.s`.

Nota: non modificare in alcun modo `es2A-main.c` e `load.s`.

Parte 3 (ottimizzazione del work)

Nella directory `es3A`, si crei una versione ottimizzata del seguente modulo `es3A.c` scrivendo un modulo `es3A-opt.c`:

```
static unsigned anno_bisestile(unsigned short anno) {
    if (anno % 4 != 0) return 0;
    else if (anno % 100 != 0) return 1;
    else if (anno % 400 != 0) return 0;
    else return 1;
}

int quanti_bisestile(unsigned short anno) {
    unsigned count = 0;
    unsigned short k;
    for (k = 0; k <= anno; k++)
        count += anno_bisestile(k);
    return count;
}
```

La funzione `es3A` implementa un algoritmo per calcolare il numero di anni bisestili presenti tra l'anno zero e l'anno passato come argomento (lo short anno). **Suggerimento per l'ottimizzazione:** l'argomento della funzione `quanti_bisestile` può assumere un valore massimo pari a `MAX_YEAR` (costante definita in `es3A.h`). **Verificare che la soluzione ottimizzata stampi gli stessi risultati di quella originaria!**

Per compilare, usare **sempre** le opzioni `-m32 -O1` e il programma di prova `es3A-main.c`.

Ai fini dell'ottimizzazione:

1. usare `gprof` per identificare le porzioni più onerose computazionalmente
2. esaminare il modulo `es3A.s` generato a partire da `es3A.c` con `gcc -S -O1` (e già fornito) per capire quali ottimizzazioni siano già state effettuate dal compilatore

Alla fine del compito, la directory **dovrà contenere** i seguenti file non presenti in origine:

1. `es3A`, eseguibile ottenuto da `es3A.c`
2. `es3A-pg`, eseguibile ottenuto da `es3A.c` con l'opzione `-pg`
3. `gmon.out`, report binario generato da `gprof`
4. `es3A-pg.txt`, report testuale generato da `gprof`
5. `es3A-opt`, eseguibile ottenuto da `es3A-opt.c`

Rispondere alle seguenti domande nel file `es3A-risposte.txt`:

1. descrivere a parole le ottimizzazioni applicate
2. riportare i tempi di esecuzione real di `es3A` ed `es3A-opt` misurati con il comando `time` e mediati su tre esecuzioni distinte (trial)
3. riportare lo speedup ottenuto (rapporto dei tempi medi calcolati al punto 2)

Parte 4 (quiz)

Si risponda ai seguenti quiz, inserendo le risposte (A, B, C, D o E per ogni domanda) nel file `es4A.txt`. **Una sola risposta è quella giusta.** Rispondere E equivale a non rispondere (0 punti).

Domanda 1 (endianness)

Si assuma di operare in un'architettura IA32 sul seguente frammento di memoria:

<i>Indirizzo</i>	0x1000	0x1001	0x1002	0x1003
<i>Contenuto</i>	0x11	0x22	0x33	0x44

Eseguendo le seguenti istruzioni:

```
movl $0xAABBCCDD, %eax
movw %ax, 0x1001
movl 0x1000, %eax
```

Cosa conterrà il registro `%eax`?

A	0x11CCDD44	B	0x11AABB44
C	0x44CCDD11	D	0x44BBAA11

Motivare la risposta nel file `M1.txt`. **Risposte non motivate saranno considerate nulle.**

Domanda 2 (estensione segno)

Si assuma di eseguire in un'architettura IA32 il seguente frammento di codice:

```
movl $0xAABBCCDD, %eax
movb $0xAF, %cl
movsbw %cl, %ax
```

Cosa conterrà il registro %eax?

A	0xFFFFFFFFAF	B	0xAABB00AF
C	0x000000AF	D	0xAABBFFAF

Motivare la risposta nel file M2.txt. **Risposte non motivate saranno considerate nulle.**

Domanda 3 (uso del compilatore)

Quale dei seguenti comandi permette di ottenere il codice assembly generato dal compilatore gcc per programma contenuto in f.c?

A	gcc -c f.c	B	gcc -S f.c
C	gcc -g f.c	D	gcc -o f.c

Motivare la risposta nel file M3.txt. **Risposte non motivate saranno considerate nulle.**

Domanda 4 (ottimizzazione)

Si consideri il seguente programma C e il corrispettivo codice assembly emesso dal compilatore (livello ottimizzazione 2):

<pre>int f(int * x, int * v, unsigned n){ int i, s, a = 0xF; for (i = 0; i < n; i++) { s += a + *x + v[i]; } return s; }</pre>	<pre>f: pushl %esi pushl %ebx movl 20(%esp), %ecx testl %ecx, %ecx je .L2 movl 12(%esp), %edx movl (%edx), %ebx movl 16(%esp), %edx leal (%edx,%ecx,4), %esi addl \$15, %ebx .L3: movl (%edx), %ecx addl \$4, %edx addl %ebx, %ecx addl %ecx, %eax cmpl %edx, %esi jne .L3 .L2: popl %ebx popl %esi ret</pre>
---	---

Quale tra le seguenti ottimizzazioni è stata effettuata dal compilatore?

A	augmentation	B	loop invariant code motion
C	common subexpression elimination	D	cortocircuitazione

Motivare la risposta nel file M4.txt. **Risposte non motivate saranno considerate nulle.**