

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

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

B

Compito di esonero (14/11/2017) – Durata 1h 30'

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

Parte 1 (programmazione IA32)

Nella directory `es1B`, si traduca in assembly IA32 la seguente funzione C scrivendo un modulo `es1B.s`:

```
int minore(short x, short y);

int confronta(short* a, short* b, unsigned n) {
    short* end = a + n;
    while (a < end)
        if (!minore(*a++, *b++)) return 0;
    return 1;
}
```

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 `es1B` con `gcc -m32 -g`. Per i test, compilare il programma insieme al programma di prova `es1B-main.c` e al modulo `minore.s`.

Nota: non modificare in alcun modo `es1B-main.c` e `minore.s`.

Parte 2 (programmazione IA32)

Nella directory `es2B`, si traduca in assembly IA32 la seguente funzione C scrivendo un modulo `es2B.s`:

```
void loads(char**);

char* get(int i) {
    char* b;
    loads(&b);
    return b + 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 `es2B` con `gcc -m32 -g`. Per i test, compilare il programma insieme al programma di prova `es2B-main.c`, e al modulo `loads.s`.

Nota: non modificare in alcun modo `es2B-main.c` e `loads.s`.

Parte 3 (ottimizzazione del work)

Nella directory `es3B`, si crei una versione ottimizzata del seguente modulo `es3B.c` scrivendo un modulo `es3B-opt.c`:

```
#include <math.h>

int radice(int a);

int sumRangeSqrt(int a, int b) {
    int i, sum = 0;
    for (i = a; i < b; i++) {
        int c = i % 2;
        sum += a + b + radice(a + c);
    }
    return sum;
}
```

Verificare che la soluzione ottimizzata stampi gli stessi risultati di quella originaria!

Per compilare, usare **sempre alla fine del comando** le opzioni `-m32 -O1 -lm` e il programma di prova `es3B-main.c`.

Ai fini dell'ottimizzazione:

1. usare `gprof` per identificare le porzioni più onerose computazionalmente
2. esaminare il modulo `es3B.s` generato a partire da `es3B.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. `es3B`, eseguibile ottenuto da `es3B.c`
2. `es3B-pg`, eseguibile ottenuto da `es3B.c` con l'opzione `-pg`
3. `gmon.out`, report binario generato da `gprof`
4. `es3B-pg.txt`, report testuale generato da `gprof`
5. `es3B-opt`, eseguibile ottenuto da `es3B-opt.c`

Rispondere alle seguenti domande nel file `es3B-risposte.txt`:

1. descrivere le ottimizzazioni applicate e dire perché si ritiene che siano efficaci
2. riportare i tempi di esecuzione real di `es3B` ed `es3B-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, 0x1002      # scrive all'indirizzo 0x1002
movl 0x1000, %eax     # legge dall'indirizzo 0x1000
```

Cosa conterrà il registro %eax?

A	0x1122DDCC	B	0xCCDD2211
C	0xDDCC3344	D	0xDDCC1122

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

Domanda 2 (estensione bit)

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

```
movl $0xAABBCCDD, %eax
movb $95, %cl
movsbw %cl, %ax
```

Cosa conterrà il registro %eax?

A	0xFFFFFFFF	B	0xAABB005F
C	0x0000005F	D	0xAABBFF5F

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 file oggetto relativo al modulo C 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 1):

<pre>int f(int x) { int k, a = 7; int b = 10, s = 0; for (k = 0; k < x; k++) { if (a < b) s += k; else s += a; } }</pre>	<pre>f: movl 4(%esp), %ecx testl %ecx, %ecx jle .L4 movl \$0, %eax movl \$0, %edx .L3:</pre>
--	---

<pre> return s; } </pre>	<pre> addl %edx, %eax addl \$1, %edx cmpl %edx, %ecx jne .L3 rep ret .L4: movl \$0, %eax ret </pre>
--------------------------	---

Quale tra le seguenti ottimizzazioni è stata effettuata dal compilatore?

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

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