

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

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

C

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

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

Parte 1 (programmazione IA32)

Nella directory `es1C`, si traduca in assembly IA32 la seguente funzione C scrivendo un modulo `es1C.s`:

```
unsigned char eq(short * x, short * y);

int is_reversed(short* a, short* b, unsigned n) {
    int k;
    short * pa = a;
    short * pb = b + n - 1;
    for (k = 0; k < n; k++)
        if (!eq(pa + k, pb - k)) 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 `es1C` con `gcc -m32 -g`. Per i test, compilare il programma insieme al programma di prova `es1C-main.c` e al modulo `eq.s`.

Nota: non modificare in alcun modo `es1C-main.c` e `eq.s`.

Parte 2 (programmazione IA32)

Nella directory `es2C`, si traduca in assembly IA32 la seguente funzione C scrivendo un modulo `es2C.s`:

```
int fetch_elem_ptr(int** buf_ptr, int* v, int i) {
    if (!buf_ptr) return -1;
    *buf_ptr = v + i;
    return 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 `es2C` con `gcc -m32 -g`. Per i test, compilare il programma insieme al programma di prova `es2C-main.c`.

Nota: non modificare in alcun modo `es2C-main.c`.

Parte 3 (ottimizzazione del work)

Nella directory `es3C`, si crei una versione ottimizzata del seguente modulo `es3C.c` scrivendo un modulo `es3C-opt.c`:

```
static int contain(int * x, unsigned xn, int v) {
    int k = 0;
    while (k < xn) if (x[k++] == v) return 1;
    return 0;
}
```

```
// Dato l'array x e l'array ordinato sorted_v, ritorna
// l'indice del piu' grande elemento di sorted_v che è anche
// presente in x, o -1 se i due array non hanno elementi
// in comune
int search_max(int * x, unsigned xn, int * sorted_v, unsigned vn) {
    int k;
    int k_max = -1;
    for (k = 0; k < vn; k++) {
        if (contain(x, xn, sorted_v[k]))
            k_max = k;
    }
    return k_max;
}
```

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 `es3C-main.c`.

Ai fini dell'ottimizzazione:

1. usare `gprof` per identificare le porzioni più onerose computazionalmente
2. esaminare il modulo `es3C.s` generato a partire da `es3C.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. `es3C`, eseguibile ottenuto da `es3C.c`
2. `es3C-pg`, eseguibile ottenuto da `es3C.c` con l'opzione `-pg`
3. `gmon.out`, report binario generato da `gprof`
4. `es3C-pg.txt`, report testuale generato da `gprof`
5. `es3C-opt`, eseguibile ottenuto da `es3C-opt.c`

Rispondere alle seguenti domande nel file `es3C-risposte.txt`:

1. descrivere le ottimizzazioni applicate e dire perché si ritiene che siano efficaci
2. riportare i tempi di esecuzione real di `es3C` ed `es3C-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          # scrive all'indirizzo 0x1001
movw 0x1000, %ax          # legge dall'indirizzo 0x1000
```

Cosa conterrà il registro `%ax`?

A	0x11DD	B	0x22CC
C	0xDD11	D	0xCC22

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 $0xF, %cl
movsbw %cl, %ax
```

Cosa conterrà il registro %eax?

A	0xFFFFFFFF	B	0xAABB000F
C	0x0000000F	D	0xAABBFFFF

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

Domanda 3 (uso del compilatore)

Quale dei seguenti comandi permette di generare un eseguibile con simboli di debugging per il programma C 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 1):

<pre>int f(unsigned x) { int k, a = 7, s = 0; for (k = 0; k < x; k++) { if (x > 0) s += a + k; else s += 25; } return s; }</pre>	<pre>f: movl 4(%esp), %ecx testl %ecx, %ecx je .L4 addl \$7, %ecx movl \$7, %edx movl \$0, %eax .L3: 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	constant folding
C	common subexpression elimination	D	dead code elimination

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