

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

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



Esempio di compito di esonero – 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`:

```
short lshift8(char);

void replica(const char* s, short* v) {
    while (*s) {
        if (*s < 'a' || *s > 'z')          // 'a' == 97, 'z' == 122
            *v = '-' + lshift8(*s);      // '-' == 45
        else *v = *s + lshift8(*s);
        s++;
        v++;
    }
}
```

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 `lshift.s`.

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

Parte 2 (programmazione IA32)

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

```
void init(char*, char*, char*);

int weird(char* d) {
    unsigned char a, b, c;
    init(&a, &b, &c);
    return a + *d < 2*c - b;
}
```

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 `init.s`.

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

Parte 3 (ottimizzazione del work)

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

```
#include "es3A.h"

char piufreq(const char* s) {
    unsigned    max    = 0;
    char        max_c  = *s;
    const char* p      = s;

    while (*p) {
        if (conta(*p, s) > max) {
            max_c = *p;
            max    = conta(*p, s);
        }
        p++;
    }

    return max_c;
}
```

La funzione `piufreq` trova il carattere più frequente in una stringa. *Suggerimento per l'ottimizzazione*: provare a ridurre il numero di iterazioni del ciclo. 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 (assegnamento condizionale)

Si consideri il seguente frammento di codice IA32:

```
f: movl 4(%esp), %ecx
    movl %ecx, %eax
    negl %eax
    cmovll %ecx, %eax
    ret
```

A quale delle seguenti funzioni C corrisponde?

A	<pre>int f(int x) { return x>0 ? -x:x; }</pre>	B	<pre>short f(short x) { return x<0 ? -x:x; }</pre>
C	<pre>int f(int x) { return x<0 ? -x:x; }</pre>	D	<pre>unsigned f(unsigned x) { return x>0 ? -x:x; }</pre>

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

Domanda 2 (comprensione programmi assembly)

Si consideri il seguente programma formato da un modulo C e da uno assembly IA32:

<pre>#include <stdio.h> int f(short* v, int n); int main() { short v[] = { 1, 2, 3, 4 }; printf("%d\n", f(v, 4)); return 0; }</pre>	<pre>f: pushl %esi movl 12(%esp), %ecx xorl %eax, %eax testl %ecx, %ecx jle L2 movl 8(%esp), %edx xorl %eax, %eax L1: decl %ecx movswl (%edx,%ecx,2), %esi addl %esi, %eax testl %ecx, %ecx jg L1 L2: popl %esi ret</pre>
---	--

Cosa stampa il programma?

A	10	B	20	C	15	D	5
----------	----	----------	----	----------	----	----------	---

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

Domanda 3 (convenzioni System V ABI)

Si consideri la seguente funzione C:

```
int f(int x, int y) {
    return x-2*y;
}
```

Una sola delle seguenti traduzioni IA32 è corretta e rispetta le convenzioni della System V ABI. Quale?

A	<pre>f: movl 4(%esp), %eax movl 8(%esp), %ecx addl %ecx, %ecx subl %ecx, %eax ret</pre>	B	<pre>f: movl 8(%esp), %eax movl 12(%esp), %ecx addl %ecx, %ecx subl %ecx, %eax ret</pre>
----------	---	----------	--

C	f: movl 4(%esp), %eax movl 8(%esp), %esi addl %esi, %esi subl %esi, %eax ret	D	f: movl 8(%esp), %eax movl 4(%esp), %edx addl %edx, %edx subl %edx, %eax ret
----------	--	----------	--

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

Domanda 4 (endianness)

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

<i>Indirizzo</i>	0x1000	0x1001	0x1002	0x1003
<i>Contenuto</i>	0xAA	0xBB	0xCC	0xDD

Eseguendo le seguenti istruzioni:

```
movw $0xFFEE, 0x1000
movl 0x1000, %eax
```

Cosa conterrà il registro %eax?

A	0xFFEECCDD	B	0xFFEEAABB
C	0xDDCCFFEE	D	0xDDCCBBAA

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