

Sistemi di Calcolo (A.A. 2016-2017)

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

C

Compito di esonero – Durata 1h 30'

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

Parte 1 (programmazione IA32)

Si traduca in assembly IA32 la seguente funzione C scrivendo un modulo `es1C.s`:

```
int media(int, int);
int test(const int* a, const int* b, const int* c) {
    return media(*a,*b) >= media(*b,*c);
}
```

L'unico criterio di valutazione è la correttezza, cioè l'equivalenza semantica tra il programma tradotto e quello C di partenza. Per i test, usare il programma di prova `es1C-main.c` e il modulo `es1C-media.s`, entrambi inclusi.

Generare un file eseguibile `es1C` compilato con `gcc -m32`.

Parte 2 (programmazione IA32)

Si traduca in assembly IA32 la seguente funzione C scrivendo un modulo `es2C.s`:

```
void arrayCrossMean(unsigned* x, unsigned* y,
                    unsigned* z, unsigned n) {
    unsigned* end = x+n;
    y = y+n-1;
    while (x < end) {
        *z = *x + *y;
        if (*z) *z = (*z) >> 1;
        x++, z++, y--;
    }
}
```

L'unico criterio di valutazione è la correttezza, cioè l'equivalenza semantica tra il programma tradotto e quello C di partenza. Per i test, usare il seguente programma di prova `es2C-main.c` incluso.

Generare un file eseguibile `es2C` compilato con `gcc -m32`.

Parte 3 (ottimizzazione work)

Si crei nel file `freq-opt.c` una versione **ottimizzata** del seguente modulo `freq.c`:

```
#include <string.h>
#include "freq.h"

// verifica se b inizia per a
static int prefix(const char* a, const char* b) {
    int c = 0;
    while (*a && *b) c += *a++ != *b++;
    return !c && !*a;
}

// conta quante volte sub è in text
static int substring_count(const char* sub, const char* text) {
    int i, c = 0, n = strlen(text)-strlen(sub);
```

```

    for (i=0; i<=n; i++)
        if (prefix(sub, text+i)) c++;
    return c;
}

int has_token(const char* text) {
    return substring_count("continue", text) ||
           substring_count("break", text) ||
           substring_count("goto", text) ||
           substring_count("return", text) ||
           substring_count("(", text) ||
           substring_count(")", text);
}

```

Il modulo implementa una funzione `has_token` che **verifica se un testo che rappresenta il codice sorgente di un programma contiene determinate parole chiave o separatori**.

Compilare due versioni del programma, usando `gcc` a **32 bit** con livello di ottimizzazione 1 e lo stesso modulo `main.c`:

1. non ottimizzata manualmente: eseguibile `freq`;
2. ottimizzata manualmente: eseguibile `freq-opt`.

Ai fini dell'ottimizzazione:

1. usare `gprof` per identificare le porzioni più onerose computazionalmente. Per evitare confusione, chiamare l'eseguibile usato per la profilazione `freq-pg` e il report del profiler `freq-pg.txt`;
2. esaminare il modulo `freq.s` generato a partire da `freq.c` con `gcc -S -O1` (e già fornito) per capire quali ottimizzazioni siano già state effettuate dal compilatore.

Rispondere alle seguenti domande:

1. Descrivere le ottimizzazioni applicate **manualmente** e dire perché si ritiene che siano efficaci.
2. Riportare la media dei tempi di esecuzione (real) di `freq` e di `freq-opt` su tre run usando il comando `time` e dire di quante volte è più veloce l'**eseguibile** `freq-opt` rispetto a `freq` (speedup).
3. Riportare il flat profile del programma `freq` usando `gprof`.

Inserire le risposte nel file `es3C.txt`. Alla fine del compito, **non cancellare** `gmon.out` e gli altri eseguibili creati.

Parte 4 (quiz)

Si risponda ai seguenti quiz, inserendo le risposte (A, B, C, D o E per ogni domanda) nel file `es4C.txt`. **Una sola risposta è quella giusta**. Rispondere E equivale a non rispondere (0 punti).

Domanda 1 (file system)

Quale dei seguenti comandi permette di redirigere lo `stdout` del comando `"cat file"` nello `stdin` del comando `"grep pippo"`?

A	<code>grep pippo <& cat file</code>	B	<code>cat file grep pippo</code>
C	<code>cat file > grep pippo</code>	D	<code>cat file & grep pippo</code>

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

Domanda 2 (permessi)

Il comando "chmod 451 file" è equivalente al comando:

A	chmod u=w,g=rw,o=r file	B	chmod u=xw,g=r,o=w file
C	chmod u=x,g=rx,o=w file	D	chmod u=r,g=rx,o=x file

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

Domanda 3 (analisi delle prestazioni del software)

Qual è lo speedup ottenibile per un programma se riduciamo del 25% una sua porzione che richiede il 50% del tempo di esecuzione?

A	1.25x	B	1.40x
C	1.50x	D	1.14x

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