

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

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

D

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 `es1D.s`:

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

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 `es1D-main.c` e il modulo `es1D-media.s`, entrambi inclusi.

Generare un file eseguibile `es1D` compilato con `gcc -m32`.

Parte 2 (programmazione IA32)

Si traduca in assembly IA32 la seguente funzione C scrivendo un modulo `es2D.s`:

```
void arrayProdWithDeref(int** x, int** y, int* z, int n) {
    int** end = x+n;
    while (x < end) {
        if (**x && **y) *z = (**x) * (**y);
        else *z = 0;
        x++, y++, z++;
    }
}
```

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 `es2D-main.c` incluso.

Generare un file eseguibile `es2D` compilato con `gcc -m32`.

Parte 3 (ottimizzazione work)

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

```
#include "anagrams.h"

int count(char c, const char* s) {
    int cnt = 0;
    while (*s) cnt += c == lower(*s++);
    return cnt;
}

int anagrams(const char* a, const char* b) {
    int cnt = 0;
    const char* p = a;
    while (*p) {
        cnt += count(lower(*p), a) !=
            count(lower(*p), b);
        p++;
    }
}
```

```
    return cnt == 0;
}
```

Il modulo implementa una funzione `anagrams` che **verifica se due stringhe sono una l'anagramma dell'altro**. Ragionare sia in termini algoritmici che di ottimizzazioni del codice.

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 `anagrams`;
2. ottimizzata manualmente: eseguibile `anagrams-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 `anagrams-pg` e il report del profiler `anagrams-pg.txt`;
2. esaminare il modulo `anagrams.s` generato a partire da `anagrams.c` con `gcc -S -O1` (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 `anagrams` e di `anagrams-opt` su tre run usando il comando `time` e dire di quante volte è più veloce l'**eseguibile** `anagrams-opt` rispetto a `anagrams` (speedup).
3. Riportare il flat profile del programma `anagrams` usando `gprof`.

Inserire le risposte nel file `es3D.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 `es4D.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 leggere le ultime 10 righe di un file testuale chiamato `file.txt`:

A	<code>cat 10 .././././file.txt</code>	B	<code>head -n 10 ./file.txt</code>
C	<code>tail -n 10 .././file.txt</code>	D	<code>grep 10 file.txt</code>

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

Domanda 2 (permessi)

Il comando `"chmod 412 file"` è equivalente al comando:

A	<code>chmod u=w,g=x,o=r file</code>	B	<code>chmod u=r,g=x,o=w file</code>
C	<code>chmod u=rx,g=w,o=x file</code>	D	<code>chmod u=x,g=r,o=w file</code>

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 60% una sua porzione che

richiede il 25% del tempo di esecuzione?

A	1.22x	B	1.18x
C	1.67x	D	1.33x

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