5/6 cfu

Ingegneria degli Algoritmi (A.A. 2011-2012)

Corsi di Laurea in Ingegneria Informatica e Automatica, Ingegneria dei Sistemi Informatici, e Laurea Magistrale in Ingegneria Informatica

Sapienza Università di Roma

Primo appello (20/06/2012) – Durata 2 ore – 5/6 cfu

Cognome: _____ Nome: _____ Matricola: _____	Autorizzo la pubblicazione del voto di questo esame sul sito web http://www.dis.uniroma1.it/~demetres/didattica/ae, secondo quanto prevede il decreto legislativo 196/2003 (codice in materia di protezione dei dati personali) che dichiaro di conoscere. In fede, _____

Rispondere alle seguenti domande, **motivando le risposte** (risposte non motivate saranno considerate nulle).

Domanda 1 [7 punti]

Si consideri il seguente modulo C e il relativo codice x86 generato da gcc 4.2.1:

```
static int f(int a, int b) {
    return (a+b)*(a+b);
}

int g(int x) {
    int y = 10;
    return 2*f(x+10,y);
}
```

```
_g: pushq %rbp
    movq  %rsp, %rbp
    addl  $20, %edi
    imull %edi, %edi
    movl  %edi, %eax
    addl  %eax, %eax
    popq  %rbp
    ret
```

Il codice generato è per una piattaforma a 32 o 64 bit? Quali ottimizzazioni sono state applicate dal compilatore? C'è un'ulteriore ottimizzazione che potrebbe essere effettuata dal compilatore su richiesta dello sviluppatore (supponendo di non dover usare un debugger)?

Domanda 2 [6 punti]

Scrivere una funzione C che restituisce un nuovo blocco allocato dinamicamente contenente una stringa ottenuta mediante la concatenazione delle due stringhe date in ingresso. La funzione deve avere il seguente prototipo:

```
char* concat(const char*, const char*);
```

Domanda 3 [5 punti]

Si consideri una routine `f` di un programma `P` e si assuma che `f`, nella sua versione

non ottimizzata, richieda il 60% del tempo di esecuzione totale di `P` su un dato input `X`. Se volessimo dimezzare il tempo richiesto dal programma `P` su input `X`, di quanto dovremmo velocizzare `f` (speedup)? E se la premessa fosse stata che `f` non ottimizzata richiede il 40% invece del 60% del tempo totale del programma `P`?

Domanda 4 [7 punti]

Si consideri la tecnica dell'*interval counting* usata dai sistemi operativi per stimare il tempo di CPU speso da un processo in modalità utente (user) e kernel (system). Si assuma che il timer usato dal sistema operativo per la misurazione scatti ogni D millisecondi e si considerino due processi single-threaded A e B che si alternano nell'uso di una CPU single-core. Sia $S_A = D \cdot C_A$ il tempo totale in millisecondi (user+system) misurato per il processo A mediante interval counting e sia T_A tempo effettivo in millisecondi in cui il processo A ha detenuto il possesso della CPU in modalità utente o kernel. Assumere che T_A abbia precisione al microsecondo.

1. Illustrare due scenari ipotetici estremi in cui:
    a. S_A è il più grande possibile rispetto a T_A (max errore di sovrastima)
    b. S_A è il più piccolo possibile rispetto a T_A (max errore di sottostima)

    Per descrivere gli scenari, usare diagrammi Gantt che mostrano l'alternarsi dei processi A e B rispetto agli scatti del timer.

2. Trascurando i tempi richiesti dai context switch e dai gestori degli interrupt, per ciascuno dei due scenari si calcoli l'errore relativo $| S_A - T_A | / T_A$.

Domanda 5 [7 punti]

Si consideri un dizionario realizzato in C mediante albero binario di ricerca con nodi allocati dinamicamente e strutturati come segue:

```
struct Node {
    int b;
    struct Node *left, *right, *parent;
}
```

Si assuma che l'allocatore di memoria fornito dalla libreria standard C abbia blocchi allineati a indirizzi multipli di 16 byte, header di 4 byte e minima dimensione di blocco di 8 byte. Assumendo di non effettuare alcuna eliminazione dal dizionario e alcuna deallocazione di nodi, quanti byte verrebbero richiesti a un sistema operativo a 64 bit per creare un dizionario contenente n interi a 32 bit? Calcolare il fattore di carico (utilizzo) del dizionario, dell'allocatore, e della coppia dizionario+allocatore in questo scenario.

Domanda 6 (facoltativa) [3 punti]

Si discutano le differenze fondamentali fra le CPU e le GPU. Si illustrino inoltre i vantaggi e le limitazioni delle GPU rispetto alle CPU come unità di calcolo general-purpose.