

Sistemi di Calcolo (A.A. 2018-2019)

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

B

Compito (24/06/2019) – Durata 1h 45'

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

Parte 1 (programmazione IA32)

Nella directory E1, si traduca in assembly IA32 la seguente funzione C scrivendo un modulo `e1B.s`. La funzione verifica se un array `v` di `n` `short` contiene elementi duplicati.

```
#include "e1B.h"
int has_duplicates(short* v, unsigned n) {
    unsigned i, j;
    for (i=0; i<n; ++i)
        for (j=i+1; j<n; ++j)
            if (v[i] == v[j]) return 1;
    return 0;
}
```

L'unico criterio di valutazione è la correttezza. Generare un file eseguibile `e1B` con `gcc -m32 -g`. Per i test, compilare il programma insieme al programma di prova `e1B_main.c` fornito.

Nota: **non** modificare in alcun modo `e1B_main.c`. Prima di tradurre il programma in IA32 si suggerisce di scrivere nel file `e1B_eq.c` una versione C equivalente più vicina all'assembly.

Parte 2 (programmazione di sistema POSIX)

Si vuole scrivere nel file E2/es2B una funzione `make_files` con il seguente prototipo:

```
void make_files(int n,
                void (*f)(int i, char name[64], char buf[256]))
```

Dato un intero `n` e un puntatore a funzione `f`, la funzione `make_files`:

1. crea `n` processi. Il processo `i`-esimo (con `i` in `[0, n-1]`) invoca `f(i, name, buf)`, dove `name` è un array di 64 caratteri e `buf` è un array di 256 caratteri, crea un file con nome `name` e contenuto `buf`, e termina;
2. attende la terminazione dei processi creati.

Creare i file con flag `O_CREAT`, `O_WRONLY`, `O_TRUNC` e permessi `0664`. Per i test, compilare il programma insieme al programma di prova `e2B_main.c` fornito, che **non** deve essere modificato.

Parte 3 (quiz)

Si risponda ai seguenti quiz, inserendo le risposte (A, B, C, D o E per ogni domanda) nel file `e3B.txt`. Una sola risposta è quella giusta. Rispondere E equivale a non rispondere (0 punti).

Domanda 1 (permessi)

Che permessi (in notazione ottale) dovrebbe avere un file per essere accessibile in lettura, scrittura ed esecuzione dall'utente proprietario, in lettura ed esecuzione dal gruppo proprietario, e solo in lettura per tutti gli altri utenti?

A	0764	B	0754
C	0642	D	0741

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

Domanda 2 (memoria virtuale)

Si consideri un sistema di memoria virtuale con uno spazio di indirizzi a 16 bit, pagine da 4 KB, e la seguente tabella delle pagine: {0xA, 0x3, 0x2, 0xE, 0x7, 0xB, 0x6, 0xD, 0x9, 0xC, 0x1, 0xF, 0x8, 0x5, 0x0, 0x4}. A quali indirizzi fisici corrispondono i seguenti indirizzi logici: 0xA3CD, 0x2B3F, 0x0CB9?

A	0xA3C5, 0x2B34, 0x0CBC	B	0x1E80, 0x2FE4, 0xA8FC
C	0x1ECD, 0x2F3F, 0xA8BC	D	0x13CD, 0x2B3F, 0xACB9

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

Domanda 3 (pipelining)

Si consideri la seguente sequenza di istruzioni:

```
addl $1, %eax
movl %edx, %ebx
subb %bl, %cl
movl %eax, %esi
```

Quanti cicli di clock vengono richiesti da una semplice pipeline a 5 stadi (Fetch, Decode, Execute, Memory, Write-Back) per completare tutte le istruzioni assumendo che gli hazard vengano risolti con stalli?

A	8	B	9
C	11	D	14

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

Domanda 4 (cache)

Si consideri una cache associativa a 2 vie con 4 linee da 128 byte ciascuna e politica di rimpiazzo LRU, inizialmente vuota. Potendo scegliere fra più linee vuote, si usa la linea con indice più basso. Si ha inoltre un processo che accede in sequenza ai seguenti indirizzi di memoria (senza interruzioni): 1446, 422, 908, 409, 665, 345, 991.

Alla fine della sequenza di accessi, quali sono gli indici dei blocchi contenuti nelle 4 linee di cache? Il trattino indica che la linea di cache rimane vuota.

A	2, -, 5, 7	B	2, 3, 7, 5
C	-, -, 7, 2	D	2, -, 7, 5

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