

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

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



Compito (25/07/2019) – 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 `e1A.s`. La funzione calcola il carattere più frequente in una stringa `s` usando come appoggio un array `map` di 256 interi.

```
#include "e1A.h"
int most_freq_char(const char* s, int* map) {
    int i, max = 0;
    clear(map, 256);
    while (*s) map[*s++]++;
    for (i=0; i<256; ++i)
        if (map[i] > map[max]) max = i;
    return max;
}
```

L'unico criterio di valutazione è la correttezza. Generare un file eseguibile `e1A` con `gcc -m32 -g`. Per i test, compilare il programma insieme al programma di prova `e1A_main.c` fornito.

Nota: **non** modificare in alcun modo `e1A_main.c`. Prima di tradurre il programma in IA32 si suggerisce di scrivere nel file `e1A_eq.c` una versione C equivalente più vicina all'assembly.

Parte 2 (programmazione di sistema POSIX)

Nel file `E2/e2A.c`, scrivere una funzione:

```
int contains(char* s, char c, int n)
```

che, data una stringa `s`, un carattere `c` e un numero di processi `n`, verifica se `s` contiene `c`. Usare il seguente algoritmo per realizzare la funzione, assumendo che `len=numero di caratteri in s` e `m=len/n` (divisione intera):

1. Se `m*n<len`, la funzione verifica innanzitutto se `c` è contenuto nella porzione di `s` che va dall'indice `n*m` incluso all'indice `len` escluso (per gestire il caso in cui `len` non sia divisibile per `n`).
2. La funzione crea poi `n` processi figli, dove il processo `i`-esimo verifica se `c` è contenuto nella porzione di `s` che va dall'indice `i*m` incluso all'indice `(i+1)*m` escluso; quando il processo termina, restituisce come codice di terminazione 1 se `c` è stato trovato nella porzione esplorata, e 0 altrimenti.
3. Il processo genitore attende la terminazione di tutti i figli, uno alla volta, e verifica se almeno uno di loro è terminato con codice 1. Se questo avviene, o il carattere era stato trovato al passo 1, la funzione `contains` restituisce 1. Altrimenti restituisce 0.

Per i test, compilare il programma insieme al programma di prova `e2A_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 `e3A.txt`. Una sola risposta è quella giusta. Rispondere E equivale a non rispondere (0 punti).

Domanda 1 (ottimizzazioni)

Supponiamo di essere in grado di ridurre di un fattore 3.5x il tempo di esecuzione di una porzione di un programma mediante delle ottimizzazioni. Che percentuale del tempo complessivo deve prendere quella porzione per ottenere uno speedup complessivo per il programma pari a 1.4x?

A	25%	B	33%
C	40%	D	50%

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

Domanda 2 (cache)

Si consideri una cache associativa a 2 vie con 4 linee da 32 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): 105, 782, 384, 1356, 11, 324, 265.

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	10, 8, 3, -	B	10, 8, -, -
C	0, 10, 8, 42	D	3, -, 10, 8

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

Domanda 3 (processi)

Un sistema operativo time-sharing è progettato per dare l'illusione all'utente che più di un processo progredisca nella computazione sullo stesso core allo stesso tempo. Si supponga che la gestione di un context switch richieda 40 usec (microsecondi). Qual è la minima durata di time slice affinché il tempo speso nei context switch non superi il 2% del tempo totale impiegato dal sistema per eseguire processi in time-sharing?

A	4 msec	B	1.96 msec
C	2.44 msec	D	800 usec

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

Domanda 4 (paginazione)

Si consideri un sistema di calcolo con spazio logico dei processi a 20 bit. Quanto deve essere grande una pagina affinché la tabella delle pagine occupi 2 KB? Si assuma che le entry della tabella delle pagine siano grandi ciascuna 32 bit.

A	1 KB	B	2 KB
C	4 KB	D	8 KB

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