Sistemi di Calcolo (A.A. 2022-2023)
Corso di Laurea in Ingegneria Informatica e Automatica
Sapienza Università di Roma

A

Compito (19/06/2023) – Durata 1h 30'

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

ISTRUZIONI PER STUDENTI DSA: svolgere a scelta due parti su tre.

Parte 1 (programmazione IA32)

RC4 è un noto algoritmo di cifratura simmetrica, utilizzato ampiamente in passato in protocolli quali SSL e WEP, ma ormai non più usato a causa della sua debolezza. In questo esercizio, viene richiesto di tradurre in assembly un'implementazione semplificata del RC4 (NOTA: non è richiesta alcuna conoscenza di crittografia per svolgere l'esercizio). Nella directory E1, si traduca in assembly IA32 la seguente funzione C scrivendo un modulo `e1A.s`:

```
#include "e1A.h"

void rc4_encrypt(unsigned char *sbox, unsigned char *pt,
                                          unsigned char *ct) {
    unsigned int n = 0, i = 0;
    unsigned char j = 0, rnd;

    while (*pt) {
        i = (i + 1) & 255;              // & e' l'operatore and
        j = j + sbox[i];
        rc4_helper(sbox, i, j, &rnd);
        ct[n++] = rnd ^ *pt++;          // ^ e' l'operatore xor
    }
}
```

La funzione esegue la cifratura di una stringa `pt` e scrive il risultato nell'array (preallocato) `ct`. 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)

Si scriva nel file `E2/es2A.c` una funzione `examStats` con il seguente prototipo:

```
int examStats(const char* fname, int * min, int * max, float * avg)
```

Dato il nome di un file `fname` contenente i risultati della valutazione di un esame, la funzione calcola il voto minimo, il voto massimo e la media dei voti di tutti gli studenti che hanno superato l'esame (cioè quelli con voto maggiore o uguale a 18). I valori calcolati devono essere scritti nelle tre variabili `min`, `max` e `avg` passate come parametro per riferimento. Al termine, la funzione restituisce il numero di studenti che hanno superato l'esame. In caso di errore la funzione deve restituire il valore `-1`.

Il file contiene una riga per ogni studente valutato, con la seguente struttura:

cognome-nome-voto

Ad esempio, una riga potrebbe contenere "`Rossi-Mario-22`". Si ipotizzi che tutti i voti siano espressi con un valore compreso tra 0 e 30, e che nessuna linea del file contenga più di 256 caratteri.

Per i test, compilare il programma con `gcc -lm` insieme al programma di prova `e2A_main.c` fornito, che **non** deve essere modificato. Porre attenzione anche a non modificare i file `.txt` usati per i test.

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 (cache)

Si consideri una cache associativa a 2 vie con 4 linee da 64 byte ciascuna e politica di rimpiazzo LRU, inizialmente vuota. I blocchi pari sono mappati sulle prime due linee, ed i blocchi dispari sulle seconde due. 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): `3200, 670, 253, 189, 5439, 1915, 946`.

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	3, 29, 2, 10	**B**	14, 84, 3, 29
C	2, -, -, 29	**D**	50, -, 3, -

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

Domanda 2 (pipelining)

Si consideri la seguente sequenza di istruzioni:

```
movl $10, %eax
addl %esi, %edx
subl $5, %eax
incl %esi
movl $3, %ebx
```

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	9	**B**	11
C	14	**D**	10

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

Domanda 3 (Analisi delle prestazioni del software)

Di quanto è necessario ridurre una porzione di un programma che richiede il 30% del tempo di esecuzione per ottenere uno speedup sul programma di ~1.21?

A	20%	**B**	40%
C	60%	**D**	80%

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

Domanda 4 (Allineamento)

Si consideri la seguente `struct` C:

```
typedef struct S {
  short x;
  char *y;
  char z;
  int w;
} S;
```

Qual è l'offset del campo `w`?

A	8	**B**	10
C	12	**D**	14

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