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

B

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 l'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 `e1B.s`:

```
#include "e1B.h"

void rc4_encrypt_rev(unsigned char *sbox, unsigned char *pt,
                                          unsigned char *ct)
{
  unsigned int i = 0;
  unsigned char j = 0, rnd;

  unsigned char *pt_aux = pt + strlen(pt) - 1;

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

La funzione esegue la cifratura di una stringa `pt` (scorrendo i caratteri dalla fine all'inizio) 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 `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 scriva nel file `E2/es2B.c` una funzione `getBalanceMovements` con il seguente prototipo:

```
int getBalanceMovements(const char* fname, int min, int max, int *
bal)
```

Dato il nome di un file `fname` contenente l'elenco delle operazioni fatte su un conto corrente bancario durante un mese, la funzione calcola il valore complessivo dei movimenti (accrediti, addebiti) compresi tra i giorni indicati dai parametri `min` e `max`. Al termine, la funzione restituisce il numero di operazioni considerate per il calcolo del valore complessivo. Il valore

complessivo viene invece scritto (intero con segno) nella variabile `bal` passata per riferimento. In caso di errore la funzione deve restituire il valore `-1`.

Il file contiene una riga per ogni operazioni svolta sul conto corrente, con la seguente struttura:

giorno`|`*[+,-]*`|`*valore*

Ad esempio, una riga potrebbe contenere "`12|+|22`" per indicare che il giorno 12 è stato fatto un accredito di € 22. Si ipotizzi che tutti i valori monetari siano espressi con un valore intero positivo, e che il valore del giorno sia un numero compreso tra 1 e 31.

Per i test, compilare il programma con `gcc -lm` insieme al programma di prova `e2B_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 (paginazione)

Si consideri un sistema di memoria virtuale con uno spazio di indirizzi a 24 bit, pagine da 1 MB, e la seguente tabella delle pagine: {`0x6, 0x3, 0x1, 0x8, 0xA, 0xE, 0x9, 0x2, 0xC, 0xF, 0x0, 0x5, 0xB, 0x4, 0x7, 0xD`}. A quali indirizzi fisici corrispondono i seguenti indirizzi logici: `0x1AE420, 0xFD3AB4, 0x27CD8D`?

A	`0x6AE4A0, 0xFD3A51, 0x12CD8D`	**B**	`0x3AE426, 0xDD8AB4, 0x12CDC4`
C	`0x3AE420, 0xDD3AB4, 0x17CD8D`	**D**	`0x1AE426, 0xFD3ABA, 0x27C4C6`

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

Domanda 2 (Ottimizzazioni)

Si consideri il seguente frammento di codice:

```
void f(int* v, int n, int x){
    int i = 0;
    for(; i < n; i++)
          v[i] = x * x;
}
```

Quale delle seguenti ottimizzazioni può essere applicata dal compilatore?

A	Dead Code Elimination	**B**	Constant Propagation
C	Loop-Invariant Code Motion	**D**	Function Inlining

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

Domanda 3 (Permessi)

Un file ha permessi `0745`. Quale di queste risposte è **falsa**:

A	Il proprietario può scrivere il file	**B**	Il gruppo proprietario può leggere il file
C	Gli altri utenti (diversi dall'utente	**D**	Gli altri utenti (diversi dall'utente

	proprietario e non membri del gruppo proprietario) possono eseguire il file		proprietario e non membri del gruppo proprietario) non possono leggere il file

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

Domanda 4 (Assembly)

Si assuma di operare in una architettura IA32. Eseguendo le seguenti istruzioni:

```
movl $0xA1B2C3D4, %eax
movb $0xFE, %dl
movsbw %dl, %ax
movl %eax, %ecx
```

Cosa conterrà il registro `%ecx`?

A	`0xFEFFB2A1`	**B**	`0xA1B2FFFE`
C	`0xFFFEB2A1`	**D**	`0xFEFFA1B2`

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