

Sistemi di Calcolo (A.A. 2023-2024)

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



Compito (13/06/2024) – 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)

Il leet (o anche l33t, 31337 o 1337) è una forma codificata di inglese caratterizzata dall'uso di caratteri non alfabetici al posto delle normali lettere. In questo esercizio, si chiede di tradurre in assembly una funzione che decodifica stringhe in leet. Nella directory E1, si traduca in assembly IA32 la seguente funzione C scrivendo un modulo `e1A.s`:

```
unsigned char* deleetify(unsigned char* v, unsigned len)
{
    unsigned char* res = malloc(len * sizeof(unsigned char));
    unsigned i;

    for (i = 0; i < len; i++) {
        if (v[i] > 57)
            res[i] = v[i];
        else
            res[i] = deleetify_helper(v[i]);
    }
    return res;
}
```

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`.

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. La funzione `deleetify_helper` ritorna il valore `'*'` in caso di input errato; quindi, se nella stringa di output riscontrate il suddetto carattere è per via di un parametro errato.

Parte 2 (programmazione di sistema POSIX)

Si scriva una funzione che aiuti l'utente nella scelta di una possibile destinazione per una villeggiatura. Nello specifico, si scriva nel file `E2/e2A.c` una funzione con il seguente prototipo:

```
void destinazioniDisponibili(const char * filename, const char *
partenza, int budget, struct destinazione ** list)
```

che, dato in input il nome `filename` del file contenente informazioni sulle connessioni disponibili, una stringa `partenza`, e un intero `budget`, restituisce in `list` la lista di tutte le destinazioni raggiungibili da `partenza` con un costo inferiore o uguale a `budget`. Per

raggiungibile si intende tramite una connessione specifica tra località di partenza e di destinazione (ossia non vanno valutate eventuali connessioni attraverso località intermedie).

Il file `filename` tiene traccia delle connessioni disponibili con righe aventi il seguente formato

DEF-UVZ-130

Ogni riga nel file rappresenta una connessione tra un punto di partenza, una destinazione, ed il relativo costo. Le tre informazioni sono separate dal carattere '-'. Nell'esempio, partendo da DEF è possibile raggiungere UVZ con un costo di 130€.

`list` deve essere una lista collegata costituita da elementi rappresentati dalla struttura `destinazione` definita nel file `e2A.h`. L'ordine degli elementi in `list` deve essere lo stesso con cui quei collegamenti compaiono in `filename`.

Per i test, compilare il programma insieme al programma di prova `e2A_main.c` fornito, che **non** deve essere modificato. Nota: non modificare i file `booked*.txt` che riportano esempi di file contenenti alcune connessioni.

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 (assembly)

Dati `%eax=0x12341234` e `%ecx=0xCAFEF00D`, che valore è contenuto nel registro `%ecx` dopo l'esecuzione delle seguenti istruzioni?

1: <code>movzbw %al, %cx</code>
2: <code>sall \$4, %ecx</code>

A	0xCAFE0034	B	0xAFE00340
C	0x0034CAFE	D	0xAFE11340

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

Domanda 2 (paginazione)

Si consideri un sistema di calcolo con spazio logico dei processi a 10 bit. Quanto deve essere grande una pagina affinché la tabella delle pagine occupi 1 KB? Si assuma che le entry della tabella delle pagine siano grandi ciascuna 16 bit

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

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

Domanda 3 (Permessi file system)

Un file ha permessi 606. Quale delle seguenti risposte è corretta?

A	È permessa l'esecuzione del file all'utente proprietario	B	Il gruppo proprietario può scrivere sul file
C	Nessuna delle risposte è corretta	D	Nessuno può leggere il file

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

Domanda 4 (interrupt)

Quale delle seguenti affermazioni sugli interrupt è vera?

A	Ogni processo può definire l'interrupt vector table per l'intero sistema	B	Ogni processo può definire l'interrupt vector table per sé stesso
C	Nessun processo in user space può modificare l'interrupt vector table	D	Un processo padre può modificare l'interrupt vector table solo per i figli

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