

Architettura dei Calcolatori - a.a. 2016/17

Prova di Laboratorio - assembly MIPS: stupid_decrypt

19 Settembre 2017

Introduzione

Lo scopo di questa prova di laboratorio è lo sviluppo di un semplice programma nel linguaggio assembly del processore MIPS. Non è richiesta una particolare base di conoscenze algoritmiche, ma semplicemente un minimo di dimestichezza con la programmazione assembly.

Istruzioni

Cominciate facendo il login sulla macchina del laboratorio che vi è stata assegnata. Per il login occorre usare matricola e password dello *student portal*. Nel disco Z: troverete una cartella (*mips*) contenente il simulatore Mars. Tutto il vostro codice (sia esso costituito da un singolo file, o da file multipli) andrà salvato nella cartella "mips" da creare sul drive H: .

Create un file *student-info.txt* con incluso il vostro nome e cognome e numero di matricola nella cartella "mips". Per maggior sicurezza, includete anche nome, cognome e matricola come commento, in testa ad ogni file sorgente. Alla fine della prova, i file saranno prelevati automaticamente dalla directory. Tutto quello che lascerete nella cartella *mips* sarà utilizzato per la valutazione. Salvare i vostri file altrove, o non indicare nome e cognome, porterà inevitabilmente all'annullamento della vostra prova. *Tutti i file all'esterno della cartella verranno cancellati automaticamente!!!*

Le specifiche

Un metodo molto semplice per criptare un testo è fare l'XOR di ogni carattere con una chiave di 1 byte (un numero tra 0 e 255). Conoscendo la chiave e rifacendo l'XOR sul testo criptato si ottiene il testo originale (il metodo è abbastanza stupido, perché un esperto riesce facilmente a decriptare il testo senza fare 256 tentativi). Dovete scrivere un programma assembly che legga un testo criptato assegnandolo ad una stringa, chieda la chiave per decriptare il testo (verificando che sia un numero tra 0 e 255) e stampi il testo in chiaro mediante una funzione con parametri (la chiave e la stringa da decriptare) *decrypt*. La chiamata della funzione dovrà rispettare le convenzioni per il salvataggio dei registri. Utilizzate le system call SPIM/MARS per l'I/O e la terminazione del main.

Il seguente è un output di esempio:

```
Testo da decriptare: 5%34/`#/-0)4/`??`42/00/`&!#),%aJ
```

```
Chiave (0-255): 64
```

```
Testo in chiaro: Questo compito è troppo facile!
```

```
-- program is finished running --
```

Valutazione

Scrivere un programma funzionante secondo le specifiche, e che segua le convenzioni di salvataggio dei registri è strettamente necessario per essere ammessi a sostenere l'orale. In quella sede, si entrerà nel dettaglio della struttura del codice, dando una valutazione migliore a soluzioni "pulite" e ben commentate.