

Architettura dei Calcolatori a.a. 2012/13

Prova di Laboratorio - assembly MIPS: trova_somma_differenza

24 Giugno 2013

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*. Sul desktop troverete una cartella contenente i simulatori QtSpim e Mars. Lanciate ed utilizzate quello che preferite. 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

Dovete scrivere un programma assembly che legga due numeri interi (positivi, negativi o nulli) da tastiera, e stampi la loro somma e la loro differenza. L'individuazione della somma e della differenza va fatta utilizzando due funzioni (`trova_somma`, `trova_differenza`), che ricevono come parametri i due numeri interi, e ne ritornano somma e differenza, rispettivamente, senza effettuare nessuna verifica di eventuali overflow. La chiamata delle due funzioni dovrà rispettare le convenzioni per il salvataggio dei registri. Utilizzate le system call SPIM/MARS per l'I/O e la terminazione del main. ***È fondamentale utilizzare funzioni con scambio dei parametri per individuare somma e differenza.***

Per i più bravi: inserire il codice per verificare la presenza di overflow (sia nella somma, che nella differenza), ed eventualmente, segnalarne l'occorrenza.

Suggerimenti

Potete leggere i due numeri in input da tastiera utilizzando la system call 5. Per la stampa di stringhe e di interi potete usare le syscall 4 e 1, rispettivamente. Il seguente è un output di esempio:

```
Dammi due numeri interi:
3
5

La somma è 8.
la differenza è -2.

-- program is finished running --
Reset: reset completed.

Dammi due numeri interi:
-2000000000
2000000001

La somma è 1.
La differenza è 294967295.
ATTENZIONE: si è verificato un overflow!!!

-- program is finished running --
```

Valutazione

Scrivere un programma funzionante, che faccia uso di due funzioni come richiesto nelle 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.