

[Sessione 2] Uso delle syscall

[s2.1] Intero successivo

nome del file sorgente: intsuccessivo.asm

Si scriva codice assembly che:

- chieda all'utente di inserire un intero (messaggio su terminale);
- acquisisca un intero da terminale;
- calcoli l'intero successivo;
- mostri all'utente il risultato (messaggio su terminale).

[s2.2] Intero e successivo in array

nome del file sorgente: successivoarray.asm

Si scriva codice assembly che:

- chieda all'utente di inserire un intero (messaggio su terminale);
- acquisisca un intero da terminale;
- calcoli l'intero successivo;
- memorizzi l'intero ed il successivo in un array di dimensione 2 in memoria;
- mostri all'utente i due numeri (messaggio su terminale).

[s2.3] Spilling di registri

nome del file sorgente: spilling1.asm

Si supponga di poter usare soltanto i registri \$s0 e \$t0.

Si scriva il codice assembly che:

- calcoli la somma dei primi tre numeri interi positivi (1, 2 e 3), ciascuno moltiplicato per 3;
- non si utilizzi la pseudo-istruzione mul.

[s2.4] Semplice slotmachine con syscall

WARNING : Questo sorgente è eseguibile solo con l'emulatore MARS

nome del file sorgente: slotmachine.asm

Utilizzando le syscall rese disponibili dall'emulatore MARS implementare un programma che:

- Richieda all'utente attraverso una finestra di dialogo l'inserimento di un numero intero `NUM`
- Estragga un numero casuale `R` nel range `[-NUM, NUM]`, (il seed del generatore di numeri casuali può essere inizializzato con un qualsiasi numero intero)
- Sommi `R` al numero inserito `NUM` : `RESULT = NUM + R`
- Mostri all'utente attraverso una nuova finestra di dialogo il nuovo credito dell'utente dopo la scommessa (`RESULT`).

[s2.5] Cosa fa?

nome del file sorgente: cosa_fa.asm

Eseguire e analizzare il codice assembly riportato nel file `cosa_fa.asm` e descrivere, in linguaggio di alto livello e aggiungendo commenti al file, le operazioni che esegue.