

Calcolatori Elettronici

Esercitazione 5

M. Sonza Reorda – M. Monetti

M. Rebaudengo – R. Ferrero

L. Sterpone – E. Vacca

Politecnico di Torino

Dipartimento di Automatica e Informatica

Esercitazione 5 - Obiettivi

- Stack
- Algoritmi

Stack – Esercizio esempio

```
.data
.text
.globl main
.ent main

main:
    li    $s0, 0xF0
    li    $s1, 0xF1
    li    $s2, 0xF2

    subu   $sp, $sp, 12
    sw     $s0, ($sp)
    sw     $s1, 4($sp)
    sw     $s2, 8($sp)

    lw     $s0, ($sp)
    lw     $s1, 4($sp)
    lw     $s2, 8($sp)
    addu   $sp, $sp, 12

    li $v0, 10
    syscall
.end main
```

Stack

```
R28 [gp] = 10008000
R29 [sp] = 7ffff7a8
R30 [s8] = 0
R31 [ra] = 0
```

User Stack [7ffff7a8]..[80000000]

[7ffff7a8]	00000001	7ffff86b		
[7ffff7b0]	00000000	7fffffe1	7fffffba	7fffff83
[7ffff7c0]	7fffff47	7fffff16	7ffffef9	7ffffed5
[7ffff7d0]	7ffffea3	7ffffe72	7ffffe4a	7ffffe3d
[7ffff7e0]	7ffffe27	7ffffdfd	7ffffddf	7ffffdc8

Stack

```
subu    $sp, $sp, 12      # $SP= 7FFFF79C
sw      $s0, ($sp)        # $SP= 7FFFF79C << $S0
sw      $s1, 4($sp)       # $SP= 7FFFF7A0 << $S1
sw      $s2, 8($sp)       # $SP= 7FFFF7A4 << $S2
```

User Stack [7ffff79c] .. [80000000]

[7ffff79c]	000000f0			
[7ffff7a0]	000000f1	000000f2	00000001	7ffff86b
[7ffff7b0]	00000000	7fffffe1	7fffffba	7fffff83
[7ffff7c0]	7fffff47	7fffff16	7ffffef9	7ffffed5

7FFFF79C - 4	3^ Elemento
7FFFF7A0 - 4	2^ Elemento
7FFFF7A4 - 4	1^ Elemento
7FFFF7A8	Indirizzo iniziale \$SP

Esercizio 1

- La *system call* 1 scrive in output un numero intero con segno, compreso fra -2^{31} e $2^{31} - 1$.
- Volendo stampare un intero *unsigned* su 32 bit, non è possibile utilizzare tale system call
 - Che valore è visualizzato se il numero è un intero senza segno compreso fra 2^{32} e $2^{32} - 1$?
- Data una variabile di tipo *word* in memoria inizializzata a 3141592653, si realizzi un programma che ne stampi il valore in output.
- Il programma deve scrivere le singole cifre tramite la system call 11.

Implementazione

- Si utilizza un algoritmo in due passi:
 1. Scomposizione del numero nelle sue cifre tramite divisioni successive per 10, salvando i resti e ripetendo l'operazione sul quoziente sino a che questo è diverso da zero
 2. Visualizzazione dei resti in ordine inverso a quello di generazione, utilizzando lo *stack*
- N.B.: le cifre devono essere convertite in caratteri ASCII prima della stampa.

Esercizio 2

- Si scriva un programma che verifichi se la stringa introdotta dall'utente è palindroma.
- La lettura dell'input avviene un carattere alla volta tramite la system call 12 e termina quando l'utente introduce '\n'.
- Il numero di caratteri introdotto dall'utente non è noto a priori, quindi si utilizzi lo *stack* per memorizzarli invece di allocare una quantità di memoria costante.

Esercizio 3

- Si scriva un programma in linguaggio MIPS che dica se un'equazione di secondo grado nella forma

$$ax^2 + bx + c = 0$$

abbia o meno soluzioni reali.

- a , b e c sono interi con segno introdotti dall'utente.
- Per i salti condizionati, si utilizzino soltanto le istruzioni `slt`, `beq` e `bne`.
- Sia lecito assumere che i calcoli non diano *overflow*.

Esercizio 4

- Sia data una matrice quadrata di *word* memorizzata per righe (numero di righe pari a DIM, con DIM dichiarato come costante).
- Si scriva un programma che sia in grado di valutare se la matrice quadrata è simmetrica o diagonale. Il programma dovrà stampare a video un valore pari a:
 - 2 se la matrice è diagonale
 - 1 se la matrice è simmetrica
 - 0 se la matrice non è simmetrica.

Esercizio 4 [cont.]

- Si ricorda che in una matrice diagonale solamente i valori della diagonale principale possono essere diversi da 0, mentre una matrice simmetrica ha la proprietà di essere la trasposta di se stessa

- Esempio di matrice diagonale:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 5 \end{bmatrix}$$

- Esempio di matrice simmetrica:

$$\begin{bmatrix} 1 & 4 & 5 & 6 & 7 \\ 4 & 2 & 8 & 6 & 4 \\ 5 & 8 & 3 & 2 & 9 \\ 6 & 6 & 2 & 4 & 4 \\ 7 & 4 & 9 & 4 & 5 \end{bmatrix}$$