



Corso di Laboratorio di Architettura degli Elaboratori A.A. 2006-2007

Daniele Nadalutti

Uso della Memoria

SPIM divide spazio indirizzamento in 5 parti:

- $0 - 400000_{hex}$: riservata al sistema operativo.
- $400000_{hex} - 10000000_{hex}$: codice programma
- $10000000_{hex} - xx$: costanti, variabili globali
\$gp (global pointer) facilita accesso
- $xx - yy$: spazio dati dinamici,
syscall sbrk n alloca n byte
- $zz - 7ffffff_{hex}$: stack per chiamate di procedura

Chiamata di procedura

- Jump and link: `jal Label`
salta all'etichetta `Label`, salva nel registro `$ra` (return address) l'indirizzo dell'istruzione successiva (a `jal Label`).
- Uscita dalla procedura: `jr $ra`
- Altre istruzione di salto con link: `jalr`, `bgezal`, `bltzal`

Chiamata di procedura standard

```
main:
sub $sp, 4      #sposta lo stack pointer
sw $ra, 0($sp)  #salva l'indirizzo di ritorno
...            #caricamento eventuali argomenti nei registri $a
jal procedura
lw $ra, 0($sp)  #termina il programma
add $sp, 4
jr $ra

procedura:
...            #istruzioni della procedura
jr $ra         #ritorna al main
```

Chiamata di procedura ricorsiva (1/3)

```
main:
sub $sp, 4      #sposta lo stack pointer
sw $ra, 0($sp)  #salva l'indirizzo di ritorno
li $a0, 10
jal fatt
move $a0, $v0   #stampa il risultato
li $v0, 1
syscall
lw $ra, 0($sp)  #termina il programma
add $sp, 4
jr $ra
```

Chiamata di procedura ricorsiva (2/3)

```
fatt:
sub $sp, 8
sw $ra, 4($sp)      #salva l'indirizzo di ritorno
sw $a0, 0($sp)      #salva $a0 nello stack
bgtz $a0, rec        #caso base
li $v0, 1            #fatt(0)=1
b return

rec:                 #ricorsione
sub $a0, 1
jal fatt             #calcola il fattoriale di n-1
lw $a0, 0($sp)       #recupera $a0 dallo stack
lw $ra, 4($sp)       #recupera l'indirizzo di ritorno
mul $v0, $v0, $a0     #fatt(n)=n*fatt(n-1)
```

Chiamata di procedura ricorsiva (3/3)

```
return:  
add $sp, 8    #dealloca il frame  
jr $ra        #ritorna
```