

Università di Firenze – Laurea Triennale in Informatica

Corso di Architetture degli Elaboratori – A.A. 2018/2019

Procedure

Attenzione: si raccomanda di far attenzione all'uso delle convenzioni di chiamate a procedura viste a lezione e riportate brevemente di seguito.

Registri riservati da MIPS alle chiamate di procedura:

- `$a0, $a1, $a2, $a3`: per il passaggio di parametri.
- `$v0, $v1`: per la restituzione dei valori.
- `$ra`: per ritornare al punto di origine del chiamante.
- istruzione jump and link: `jal indirizzo` (salta all'istruzione memorizzata ad indirizzo e pone in `$ra` l'indirizzo dell'istruzione successiva a `jal indirizzo`).
- Istruzione per il ritorno: `jr $ra` (esegue l'istruzione il cui indirizzo è contenuto in `$ra`).

Utilizzo della pila (stack):

- Per ogni procedura non ancora terminata esistono dei dati locali e un indirizzo di ritorno della procedura.
- Per il passaggio dei parametri al chiamato si usano i registri `$a0 . . . $a3` (4 registri in tutto). Se ho più di 4 parametri da passare o più chiamate innestate occorre una struttura dati per memorizzare le variabili locali del chiamato, gestire le chiamate annidate e i parametri dal quinto in poi.

Convenzioni per la chiamata di Procedura: Chiamante

- passa gli argomenti se ce ne sono. Per i primi 4 si usano i registri `$a0, . . . , $a3`. I successivi vanno memorizzati nella pila.
- salva nel proprio stack i registri `$a0, . . . , $a3, $v0, $v1` e `$t0, . . . , $t9` se intende usarli dopo la chiamata (registri caller-saved).
- esegue `jal`: salta alla prima istruzione del chiamato e salva in `$ra` l'indirizzo di ritorno.

Convenzioni per la chiamata di Procedura: Chiamato

- alloca memoria per il proprio stack. Es., se lo stack è di 32 byte, `$sp = $sp - 32`
- salva nello stack i registri `$s0, . . . , $s7, $fp, $ra` se intende alterarli (registri callee-saved)

Convenzioni per il ritorno da Procedura: Chiamato

- mette i valori risultato in `$v0` e `$v1` (se esistono)
- ripristina dallo stack i registri callee-saved che aveva salvato prima dell'esecuzione della procedura (compreso, eventualmente, `$ra`)
- de-alloca lo stack: `$sp = $sp + 32`
- ritorna al programma chiamante: `jr $ra`

Convenzioni per la il ritorno da Procedura: Chiamante

- usa i valori risultato in `$v0` e `$v1` (se esistono)
- ripristina dallo stack i registri caller-saved

Esempio:

Prendere dimestichezza col programma *fattoriale.s* .