

Laboratorio di Architetture degli elaboratori n. 17, 25-27/4/04

ESERCIZIO 1

Si progetti una subroutine per stampare a terminale una lista secondo il formato [*elemento*₁, *elemento*₂, ..., *elemento*_{*n*}]. Ad esempio [1, 2, 3, 1]. Si faccia attenzione a stampare bene la lista vuota (stampare []) e la lista di un solo elemento (stampare [1] non [1,]).

Si consiglia di definire nel segmento dati le seguenti stringhe:

```
prefix:      .asciiz "["
separator:   .asciiz ", "
suffix:      .asciiz "]"
```

ESERCIZIO 2

Si progetti una subroutine che accetti in ingresso:

1. l'indirizzo base di una lista *L* di numeri interi a 32 bit nel registro A0,
2. l'indirizzo base di un blocco di memoria libera di 2 word in A1.
3. un numero positivo *n* (a 32 bit) in A2,

La subroutine deve copiare *n* nel campo dati del nodo di indirizzo A1 e poi inserire detto nodo come ultimo elemento della lista *L*. Ad esempio, se *L* = [3,6,7,8,11] e *n* = 9 la subroutine deve modificare *L* in [3,6,7,8,11,9].

Si assuma *L* **non vuota**. Poi si ragioni su cosa comporta rimuovere tale assunzione.

Nel main di test si imposti A2 staticamente o, se si preferisce, si usi pure la seguente funzione di allocazione dinamica che restituisce in V0 un puntatore ad un blocco di memoria libera di dimensione A0 words (param ingresso)

```
alloc:      lw      $v0,heap_pntr      # heap_pntr è una var glob
            sll     $t0,$a0,2         # dim words -> dim bytes
            add     $t1,$v0,$t0
            la      $t2,heap_pntr
            bgt    $t1,$t2,noalloc
            sw      $t1,heap_pntr
            jr      $ra
noalloc:    move    $v0,$zero          # se non ho spazio rit. null
            jr      $ra
            .data
heap:       .space  400                # multiplo di 4
heap_pntr:  .word   heap              # inizializzo, poi cambia
```

(Ovviamente se si "sfora" nell'uso della memoria assegnata heap_pntr viene corrotto)