

## Laboratorio di Architetture degli elaboratori n. 16, 9-10/5/02

### NOTAZIONE

Analogamente a quanto fatto per vettori e matrici solitamente utilizzeremo il termine *indirizzo base* di una lista per denotare l'indirizzo del primo nodo di una lista. Il primo nodo di una lista viene anche detto testa, quindi l'indirizzo della testa di una lista  $L$  coincide con l'indirizzo base di  $L$ .

### ESERCIZIO

Si progetti una subroutine assembler 68000 che accetti in ingresso l'indirizzo base di una lista di interi a 16 bit mediante il registro A0 e produca in uscita nella word meno significativa del registro D0 la lunghezza della lista. Con lunghezza di una lista si intende il numero di nodi di una lista. **N.B.** la lunghezza di una lista vuota (cioè di indirizzo base 0 long) è 0 word. Ad esempio nel caso della lista `lista` di seguito dobbiamo rientrare con il valore 4 nel registro D0.

```
lista    dc.l  nodo2
         dc.w  1
nodo4    dc.l  0
         dc.w  1
nodo3    dc.l  nodo4
         dc.w  3
nodo2    dc.l  nodo3
         dc.w  2

length  movem.l  a0,-(sp)
        clr.w    d0
loop    cmpa.l   #0,a0
        beq     exit
        addq.w  #1,d0
        movea.l (a0),a0
        bra     loop
exit    movem.l  (sp)+,a0
        rts
```

### ESERCIZIO

Si progetti una subroutine assembler 68000 che accetti in ingresso:

1. un numero naturale  $k$  a 16 bit nel registro D0
2. l'indirizzo base di una lista  $L$  di valori su  $[0,k]$  nel registro A0
3. l'indirizzo base di un vettore  $V$  di naturali a 16 bit di lunghezza  $k+1$  nel registro A1.

La subroutine deve assegnare ad ogni elemento  $V[i]$  il numero di occorrenze del valore  $i$  nella lista  $L$ . Ad esempio con  $k=5$  e la precedente lista `lista` dovremo produrre il vettore

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 0 | 2 | 1 | 1 | 0 | 0 |
|---|---|---|---|---|---|

(Visto che non è specificato altrimenti, il contenuto iniziale del registro  $V$  deve essere assunto indefinito.)

```
count  movem.l   a0/a2/d0,-(sp)
        movea.l  a0,a2
zeros  clr.w     (a2)+
        dbra    d0,zeros
loop   cmpa.l   #0,a0
        beq     exit
        move    4(a0),d0
        addq.w  #1,(a1,d0.w)
        movea.l (a0),a0
        bra    loop
exit   movem.l  (sp)+,a0/a2/d0
        rts
```