Laboratorio di Architetture degli elaboratori n. 18, 19-20/5/03

ESERCIZIO 1

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.

ESERCIZIO 2

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.)

Ripasso LISTE

La convenzione utilizzata per memorizzare un nodo di una lista è (quella vista a lezione e cioè) la seguente:

puntatore all'elemento successivo (Long)	dato

La dimensione del dato ovviamente dipenderà dal tipo di lista (in caso di liste di interi a 16 bit avremo una word, in caso di liste di stringhe ci saranno un numero variabile di char terminati dal carattere 0, eccetera).

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*.

Si abuserà spesso la notazione indicando una lista il cui indirizzo del nodo iniziale è contenuto in un registro col nome del registro stesso. Si dirà quindi "la lista A0" invece di "la lista il cui indirizzo del primo nodo è contenuto in A0".

Data una lista "A0", il valore del contenuto della testa è 4(A0), mentre il puntatore all'elemento successivo è (A0). Si testino opportunamente i seguenti stralci di codice che operano su liste:

- per caricare il secondo elemento di una lista di byte "A0" in D0:

```
cmpa.l   #0,a0
beq      error
movea.l  (a0),a0
cmpa.l   #0,a0
```

```
beq     error
move.b  4(a0),d0
```

- per cancellare il nodo successivo di “A0”:

```
cmpa.l  #0,a0
beq     error
movea.l (a0),a1
cmpa.l  #0,a1
beq     error
movea.l (a1),a1
move.l  a1,(a0)
```

- per inserire “A1” a seguito di “A0”:

```
cmpa.l  #0,a0
beq     error
cmpa.l  #0,a1
beq     error
movea.l (a0),(a1)
move.l  a1,(a0)
```

- per posizionare A0 sul nodo che contiene D0, se esiste, nella lista di byte “A0”:

```
loop    cmpa.l  #0,a0
        beq     failure
        cmp.b   4(a0),d0
        beq     success
        movea.l (a0),a0
        bra     loop
```

Per **capire** meglio il codice si consiglia di simularlo **manualmente** su una rappresentazione grafica.

Per una **verifica corretta** (col simulatore) si definiscano le liste di test in modo non contiguo, ad esempio per la lista di interi 1,2,3,4:

```
lista1  dc.l  nodo2
        dc.w  1
nodo4   dc.l  0
        dc.w  4
nodo3   dc.l  nodo4
        dc.w  3
nodo2   dc.l  nodo3
        dc.w  2
```

oppure per la lista di stringhe “Laboratorio”, “di”, “Architetture”,””:

```
nodoC   dc.l  nodoD
        dc.b  “Architetture”,0
nodoB   dc.l  nodoC
        dc.b  “di”,0
lista2  dc.l  nodoB
        dc.b  “Laboratorio”,0
nodoD   dc.l  0
        dc.b  0
```