

ASSEMBLY DIRETTIVE

giovedì 1 dicembre 2022 16:54

SCHELETRO ASSEMBLY!

IL PROGRAMMA HA UN INIZIO E UNA FINE;
INOLTRE DEVE ESSERE CARICATO IN MEMORIA A PARTIRE DA UN INDIRIZZO;

```
.org [INDIRIZZO CARICAMENTO] → dà all'assemblatore quale  
.data → l'indirizzo di memoria iniziale  
      # Dichiarazione costanti e variabili globali  
  
.text  
main:  
      # Corpo del programma  
      hlt # Per arrestare l'esecuzione
```

DIRETTIVE ASSEMBLY!

- **LABEL** = È una direttiva! (`.NAME LABEL:`)

`label`: mnemonico testuale definito dal programmatore ed associato all'indirizzo di ciò che la segue immediatamente

- **LOCATION COUNTER** = Identificato da `“.”`.
Viene visualizzato con il valore dell'indirizzo corrente;
Tiene traccia dell'indirizzo a cui l'assemblatore è arrivato

È SIA direttiva che variabile, cioè lo posso anche impostare esplicitamente il location counter, posso modificare il suo valore per saltare ad un altro indirizzo.

Se ho una stringa "Hello world \n", quanto è grande?

Chiedo all'assemblatore di fare una sottrazione tra l'indirizzo iniziale della mia stringa, che è la label `.MSG`, e il valore corrente del location counter;

```
msg:
```

```
.ascii "Hello, world!\n"
```

```
len = . - msg
```

valore corrente del location counter.

Da ora in poi posso utilizzare `len` al posto di un numero, perché è una costante.

Il location counter lo posso anche usare per modificare la

Il location counter lo posso anche usare per modificare la directive org.

• ORG ADDRESS = impostare il location counter ad un determinato indirizzo.

• .org address, fill: metodo alternativo di impostare il location counter, impostando i byte a fill

ESISTE UN ALTRO MODO PER DEFINIRE COSTANTI:

• EQU symbol, expression

significa che la COSTANTE symbol ha valore expression;

- Metodo alternativo: symbol = expression
- Lo stesso simbolo può essere ridefinito in più parti del codice
- Non si può usare il simbolo prima della sua definizione (one pass scan)

Ad un altro punto dobbiamo definire i byte primari: Byte, Word, Longword e Quadword

• BYTE

• WORD

• LONG

• QUAD

- .byte expressions: riserva memoria (di dimensione byte) per expressions:
var: .byte 0 → VAR è uguale all'indirizzo in cui è scritto 0.
array: .byte 0, 1, 2, 3, 4, 5
- .word expressions: riserva memoria (di dimensione word) per expressions
- .long expressions: riserva memoria (di dimensione longword) per expressions
- .quad expressions: riserva memoria (di dimensione quadword) per expressions

perché var, essendo una label, prende l'indirizzo dell'elemento successivo;

Il mio assemblatore si metterà al valore del location counter zero.

Per generare all'INDIRIZZO del location counter la rappresentazione binaria di un byte che gli possiamo:

CHAR VAR = 0 è uguale.

l'indirizzo associato ad un vettore sarà l'indirizzo del primo elemento del vettore!

Se lo voglio leggere il valore della variabile, devo andare in memoria all'indirizzo vero;

LE LABEL identificano l'indirizzo del prossimo oggetto;

- .ascii "string": riserva memoria per un vettore di caratteri e imposta il valore a string
- .fill repeat, size, value: riserva una regione di memoria composta da repeat celle di dimensione size impostate a value
 - size e value sono opzionali (default: size = 1. value = 0).

.BSS



- `.fill repeat, size, value`: riserva una regione di memoria composta da `repeat` celle di dimensione `size` impostate a `value`
 - `size` e `value` sono opzionali (default: `size = 1, value = 0`).
- `.text`: tutto ciò che compare da qui in poi va nella sezione testo
- `.data`: tutto ciò che compare da qui in poi va nella sezione data
- `.comm symbol, length`: dichiara un'area di memoria con nome (`symbol`) di dimensione `length` nella sezione `.bss`
- `.driver ivn`: identifica l'inizio della routine di servizio associato al codice `ivn`

a de puos della memoria comincia il driver;