



**UNIVERSITÀ
DEGLI STUDI
DI TRIESTE**

Struttura di un programma Assembly

Prof.ssa Giulia Cisotto

giulia.cisotto@units.it

Trieste, 10 marzo 2026

AGENDA DI OGGI

- *Recap (ISA, formati R-type, J-type)*
- *Formati istruzioni MIPS32*
- *Indirizzamento alla memoria*
- *Catena programmatica (parte 1)*
- **Struttura di un programma in Assembly**

STRUTTURA DI UN PROGRAMMA ASSEMBLY

```
# Title:                               Filename:
# Author:                               Date:
# Description:
# Input:
# Output:
```

preambolo

```
##### Data segment #####
```

```
.data
```

```
. . .
```

Dati utente

```
##### Code segment #####
```

```
.text
```

```
.globl main
```

```
main:                               # main program entry
```

```
. . .
```

```
li $v0, 10                           # Exit program
```

```
syscall
```

*Istruzioni del
programma*

PROGRAMMARE IN ASSEMBLY: GENERALE

- Il programma è organizzato in linee
- **Ogni linea** non bianca può essere una (pseudo)istruzione, una direttiva o un commento
- Spazi, tab e virgole (,) fungono da separatori
- Gli **identificatori** sono sequenze di caratteri alfanumerici (lettere, cifre, \$), underbar (_) e dot (.) che non cominciano con una cifra
- Ogni linea può contenere un **commento** che parte dal carattere # e si estende fino alla fine della linea

PROGRAMMARE IN ASSEMBLY: GENERALE

- Lettere **minuscole** e **maiuscole** **non** sono equivalenti
- Le **etichette** (**label**) si definiscono scrivendole all'inizio della linea, seguite da due punti (:)
- I mnemonici di istruzioni, pseudoistruzioni, direttive e registri (\$0-\$31, \$zero, \$at, ...) sono riservati
- I numeri si indicano in **decimale** (default) o, se preceduti da **0x**, in **esadecimale**
- Le stringhe si racchiudono tra doppi apici ("..."), e:
 - a capo (newline) \n
 - tabulazione (tab) \t
 - virgolette (quote) \"

PROGRAMMARE IN ASSEMBLY: DIRETTIVE

Le principali **direttive** all'assemblatore sono:

.data → dice all'assembler di passare alla **sezione dati dell'utente** (es. per dichiarare variabili globali)

.kdata → Passa alla sezione **dati del kernel**.

.text, .ktext → Passa alla **sezione testo**.

.space → Riserva un certo **numero di byte** nella sezione corrente, ma **non li inizializza**

.ascii, .asciiz, .byte, .double, .float, .half, .word

Materiale per la lezione

- *Hennessy-Patterson, cap. 1 pp.14-16*
- *Appendix A.2, A.3, A.4, A.5*

Prossima lezione: 11 marzo, h.14:00, aula 4C