

# Compilatore gcc ed editor vi

---

Sistemi Operativi L-A

## Compilatore standard C: gcc

File sorgente: deve avere come suffisso  
`.c` (Es: `HelloWorld.c`)

Compilazione di `HelloWorld.c`:  
`gcc HelloWorld.c`

viene prodotto un eseguibile, che per  
default è chiamato `a.out`

---

Sistemi Operativi L-A

# gcc

- senza opzioni, gcc preprocessa, compila, assembla ed effettua il linking producendo `a.out`
  - `gcc HelloWorld.c # produce a.out`
- Se si vuole produrre un eseguibile che si chiami in un modo diverso da `a.out`, si specifica l'opzione `-o` (output)
  - `gcc -o hw HelloWorld.c # produce hw`
- Se si vuole produrre solo il modulo oggetto e non l'eseguibile, si specifica `-c` (output per default: `.o`)
  - `gcc -c HelloWorld.c # produce HelloWorld.o`

---

Sistemi Operativi L-A

## Editor di testo: vi

**Editor standard per UNIX**, presente in tutte le distribuzioni base. Funziona con qualsiasi terminale a caratteri (per usarlo è sufficiente avere, oltre ai tasti carattere, un tasto `<ESC>` e `<INVIO>/<ENTER>`)

- prevede **diverse modalità operative**: comando, testo (inserimento/sostituzione), editor di linea
- **<ESC> viene utilizzato per passare alla modalità comando**. Per passare dalla modalità comando alla modalità editor di linea, bisogna digitare il **carattere due punti (:)**

---

Sistemi Operativi L-A

## **vi: creazione di un file**

**vi** <nome\_file> apre in ambiente **vi** il file indicato come argomento. Se il file non esiste lo crea. **vi** parte in modalità comando

es: **vi f1.txt**

si può passare **dalla modalità comando alla modalità testo**, digitando **i** (insert), oppure **a** (append), oppure **R** (replace)

## **vi: uscita dall'editor**

Per uscire da **vi** bisogna passare alla modalità riga di comando (<ESC> seguito da due punti) :

**:q!** (quit!) esce **scartando le modifiche fatte dall'ultimo salvataggio** (il punto esclamativo non è necessario se non si sono fatte modifiche)

nota: i comandi di **vi** sono *case-sensitive*

## **vi: uscita dall'editor/salvataggio file**

- **:wq** (write and quit) o anche **:x** (exit) salva il testo con le eventuali modifiche ed esce
- **:w** (write) salva il testo che si sta modificando
- **:w <file>** (write to *file*) salva il contenuto dell'editor in un nuovo file

Se il file esiste già e si intende sovrascriverne il contenuto bisogna aggiungere un punto esclamativo:

**:w! <file>**

## **vi: comandi**

- ESC u (undo): annulla l'ultima modifica /
- ESC U ripristina il contenuto della linea corrente
- ESC o inserisce una linea
- ESC x cancella un carattere
- ESC D cancella una riga a partire dal cursore
- ESC y copia nel buffer la riga corrente
- ESC p incolla (paste) nella posizione corrente il contenuto del buffer
- ESC / ricerca una stringa (poi n: next, per andare avanti nella ricerca)

## vi: comandi

vi prevede una sintassi particolare per ***ripetere n volte un certo comando***

- <ESC> 4d                      cancella 4 linee
- <ESC> 5x                      cancella 5 caratteri

<ESC>:! consente di eseguire un ***comando a livello di SO***

- <ESC>:! ls -l