

# Compilazione con gcc di programmi C

# Alcune opzioni a riga di comando

## gcc [opzioni] sorgenti

- **-o** permette di dare un nome al file prodotto (sia esso eseguibile o oggetto)
- **-c** effettua solo la fase di compilazione e non quella di linking, il file prodotto è un file oggetto
- **-Wall** abilita tutti i principali messaggi di warning del compilatore
- **-Werror** tutti i warning sono trattati come errori e quindi la compilazione si arresta
- **-pedantic** richiede che il codice sorgente sia conforme ad ISO C, altrimenti produce dei warnings
- **-g** abilita le informazioni di debugging, da usare se si vuole utilizzare **gdb**
- **-std=c99** (**-std=c11**) utilizza lo standard ISO C99 (C11) nella compilazione del codice
- **-ansi** utilizza lo standard ANSI C (C89/C90) nella compilazione del codice
- **-I** permette di specificare un percorso di directory dove verranno cercati gli include (es. **-I**. **-I**.. **-I** /usr/local/include **-I** ~/myinclude)
- **-OX** dove X può essere **s,g,0,1,2,3** ottimizza il codice prodotto per la size o per la velocità di esecuzione (spesso si usa **-O2**).  
NOTA: non usare **-OX** quando si vuole usare il debugger (**-Og** è il flag raccomandato) e ricordarsi di aggiungere **-g**.  
Per maggiori informazioni sulle opzioni di ottimizzazione, vedere ad esempio:  
<https://gcc.gnu.org/onlinedocs/gcc-9.2.0/gcc/Optimize-Options.html#Optimize-Options>

# Alcuni esempi

- `gcc prova.c // produce l'eseguibile a.out`
- `gcc prova.c -o prova // produce l'eseguibile con nome prova`
- `gcc -g prova.c -o prova // produce l'eseguibile prova. L'eseguibile contiene le informazioni di debugging indispensabili per utilizzare il debugger gdb`
- `gcc -g prova.c -c // produce il file oggetto prova.o (non effettua la fase di linking)`
- `gcc prova.c -c -o prova.obj // produce il file oggetto prova.obj (non effettua la fase di linking)`
- `gcc file1.c file2.c main.c -c // produce i files file1.o file2.o e main.o`
- `gcc -Wall -g -I /usr/local prova.c -I . -I .. -o prova // produce l'eseguibile prova, cerca gli includes, oltre che nelle directories di sistema, in /usr/local, nella directory corrente (.) nella directory padre (..), abilita tutti i messaggi di warnings e aggiunge le informazioni di debugging`