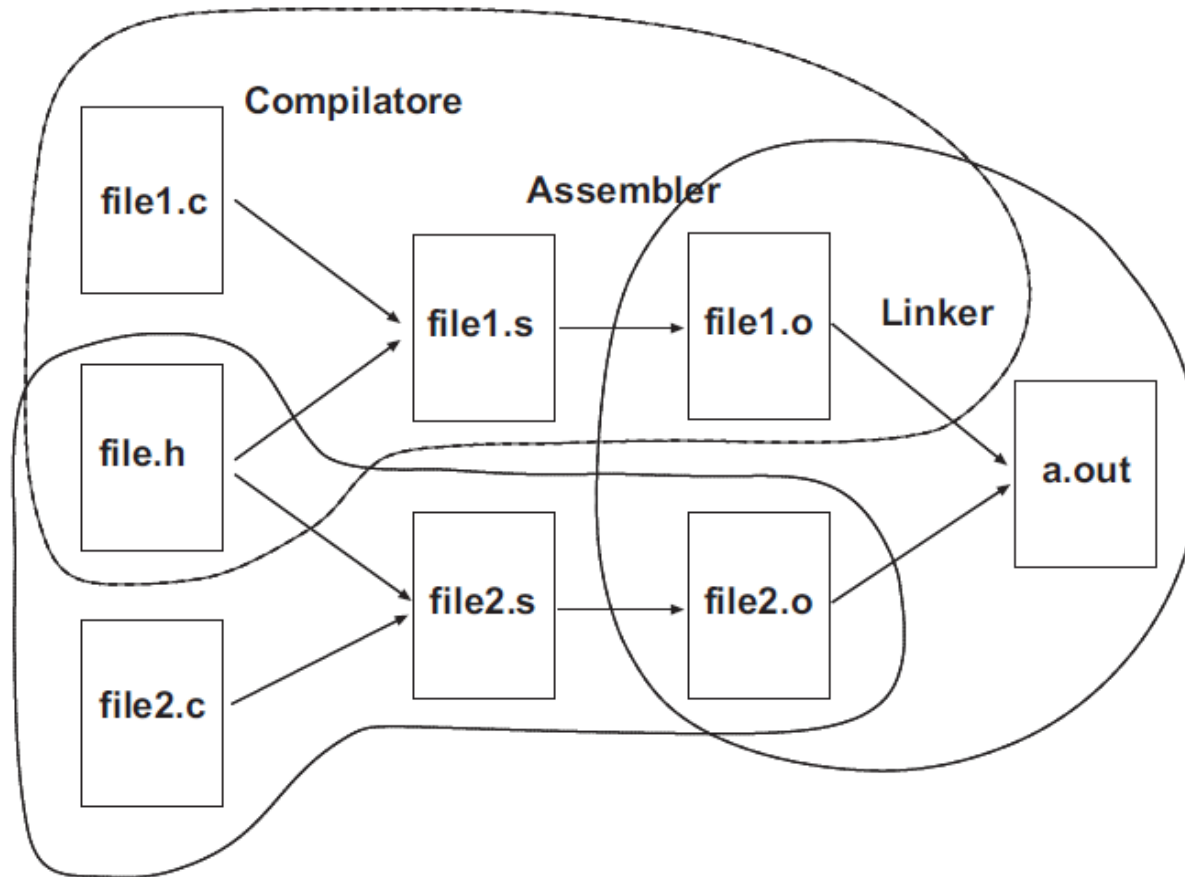


# Compilazione con GCC

- Semplice e affidabile, ma
  - Progetti complessi? Diversi file sorgente e (potenzialmente) diversi file eseguibili di output
  - Dipendenze? **Cosa** e **quando** ricompilare?
- I **makefiles** permettono di automatizzare il processo di compilazione e linking del progetto

# Compilazione con più sorgenti

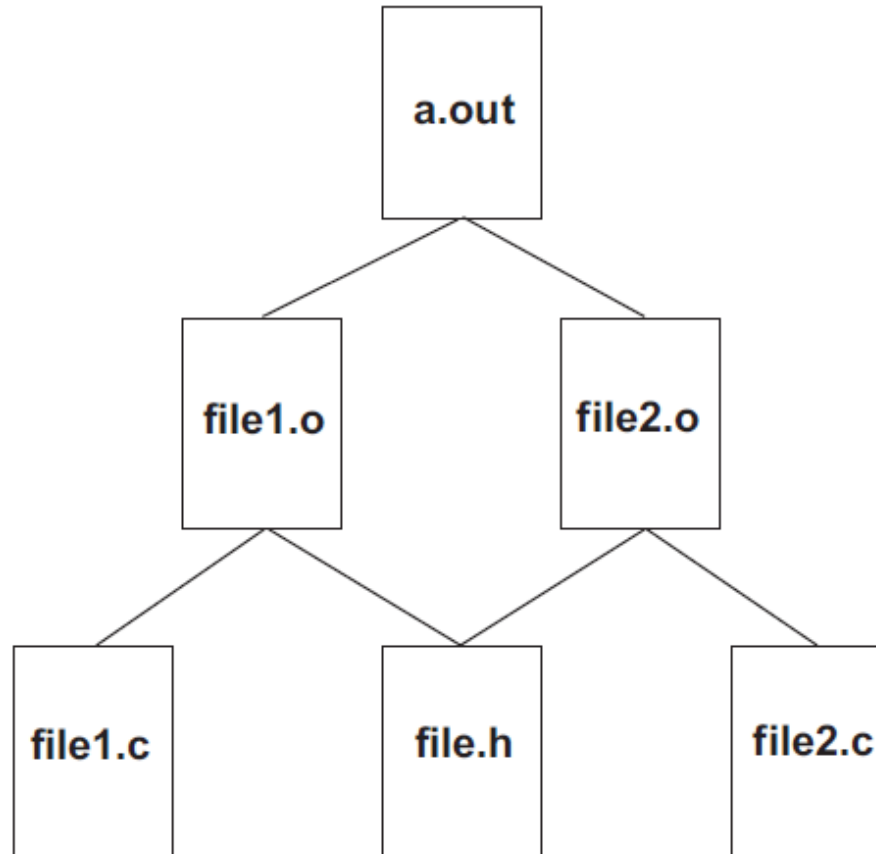


`gcc -c file1.c`

`gcc -c file2.c`

`gcc [-o a.out] file1.o file2.o`

# Grafo delle dipendenze



- Dipendenze tra sorgenti, il grafo si percorre dal basso verso l'alto

# Makefiles

- Un Makefile è un file di testo che codifica le dipendenze tra i vari file sorgente che compongono un'applicazione
- Il comando **make** legge il file chiamato **Makefile** e valuta se un programma deve essere ricompilato oppure no
  - Data di modifica
- I file che dipendono da un file modificato vengono ricompilati

# Struttura Makefiles

- Un Makefile è composto da una lista di regole composte nel modo seguente

**target: dipendenze**

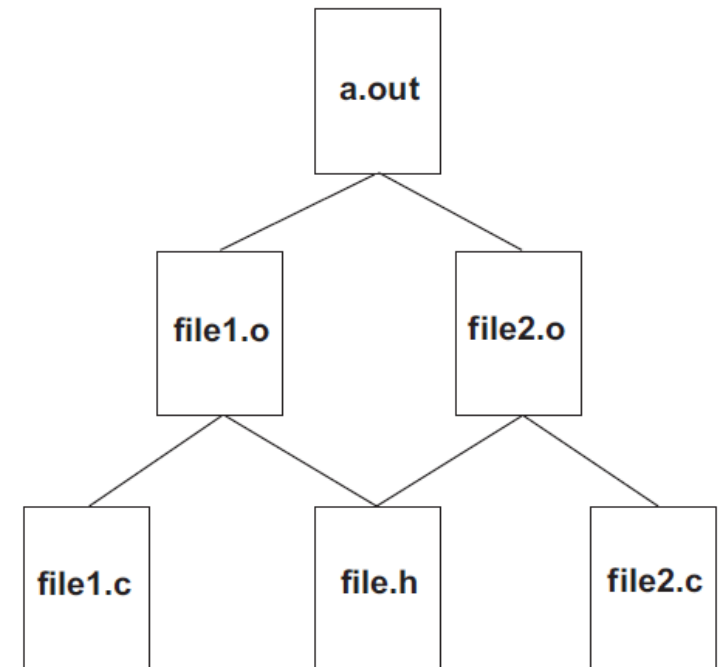
**comandi**

TAB  
↑

NEW LINE  
↑

# Esempio

- Una modifica al file **file1.c** provoca:
  - **file1.o** viene ricreato (*gcc -c*)
  - **a.out** viene ricreato assemblando il (vecchio) **file2.o** e il (nuovo) **file1.o**



**PROCESSO AUTOMATICO**

# Dipendenze vuote

- Se un target non ha dipendenze, i comandi associati vengono sempre eseguite (e.g. *make clean*)
- Ma se esiste un file chiamato *clean* nella directory corrente, il comando verrà eseguito **solamente la prima volta** (ai controlli successivi il file c'è e quindi non deve essere ricompilato)
  - Soluzione: **.PHONY** targets