

# Riassunto Makefile

Luca Parolari

22/05/2018

## 1 Regole

Un makefile è composto da *regole* che specificano come costruire alcuni file, ed in particolare specificano come creare o aggiornare file in modo algoritmico.

Una regola è composta da:

- *target*, ovvero i/il file che deve essere creato o aggiornato,
- *prerequisiti*, ovvero il/i file che devono essere usati per creare o aggiornare target,
- *recipe*, ("ricetta") ovvero una specifica di aggiornamento/creazione dei file, se ce n'è la necessità.

Intuitivamente il makefile permette di esprimere cosa deve fare il sistema per aggiornare il target se uno dei file delle dipendenze (prerequisiti) è stato modificato.

Nel makefile vanno elencate tutte le regole<sup>1</sup> che specificano come aggiornare i vari file, il sistema provvederà a farlo se necessario.

La *sintassi* di una regola è la seguente:

```
target... : prerequisiti...  
          recipe1  
          recipe2  
          recipen
```

---

<sup>1</sup>Makefile è in grado di fare delle deduzioni, quindi non sempre è necessario elencare tutte le regole (si veda la sezione 4).

## 2 Esecuzione

Per processare un make file bisogna invocare il comando **make** che legge il file *makefile* ed esegue le regole in esso contenute. Se una regola ha bisogno di prerequisiti, prima verranno eseguiti i prerequisiti i quali possono richiedere a loro volta dei prerequisiti ricorsivamente, successivamente le dipendenze verranno risolte e i file man mano aggiornati.

Vi sono poi delle regole fittizie che non coinvolgono nessun file, e prendono il nome di *phony rules*. Per esempio la regola *clean* può essere invocata con **make clean**, e può essere contrassegnata (è bene farlo) con **.PHONY : clean**.

## 3 Variabili

Per semplificare la scrittura (ma anche la lettura) del makefile si possono utilizzare le variabili che funzionano da contenitori per file o regole da eseguire.

Ciò è molto utile quando si hanno liste molto lunghe di file da dover replicare, in quanto con il copia e incolla si potrebbero commettere errori.

## 4 Deduzioni

Il comando make è "intelligente", in quanto effettua delle deduzioni su convenzioni generali.

Per esempio, è in grado di capire se c'è necessità di creare un determinato *file.o* partendo dalla sua versione *file.c*

## 5 Pulizia

Abitualmente, con il comando make, si vuole anche essere in grado di effettuare una compilazione pulita, o comunque di ripulire dai file temporanei sempre in modo algoritmico.

Il comando **make clean** permette proprio di effettuare questa pulizia. La regola *clean* può essere utilizzata anche come preconditione di altre regole, comportando, per esempio, una ricompilazione completa se aggiunta come preconditione alla regola "master".