

MAKE

giovedì 17 novembre 2022 16:44

IL MAKE PRENDE IN INPUT dei file di regole che si chiamano "MAKEFILE", questo MAKEFILE descrive la ricetta che è necessaria per passare da N file sorgente ad un eseguibile finale!

IL MAKE FILE È UN FILE DI TESTO CHE CONTIENE LA RICETTA PER SEMPLIFICARE IL PROCESSO DI COMPILAZIONE.

È BASATO SU UN INSIEME di regole, ciascuna regola ha delle dipendenze

MAKE FILE!

Implementazione della regola. \$(CC) è una variabile di ambiente che identifica il compilatore predefinito di sistema. \$(CFLAGS) richiama la variabile precedentemente dichiarata. ex1.c è il file di input al compilatore. -o è un flag che indica come chiamare l'eseguibile compilato (default: a.out)

CFLAGS=-Wall -Wextra → Variabile INTERNA al programma

→ È il nome di una regola, dopo i : posso mettere un elenco di file che sono le dipendenze di questa regola
all:ex1.c

→ \$(CC) \$(CFLAGS) ex1.c -o ex1

clean:

-rm ex1

È la regola di default!
(se non vergho passi parametri a make)

Una REGOLA è definita fondamentalmente da un nome. È un nome che descrive cosa bisogna fare in quella specifica regola.

Cflags (che è una variabile) sono dei flags di compilazione aggiuntivi che chiediamo al compilatore di utilizzare nel momento in cui porta a termine uno step di compilazione. Stiamo chiedendo di utilizzare due flags di compilazione: -Wall -Wextra
-W CHIEDE AL COMPILATORE DI GENERARE DEI WARNING.

all dipende dal file ex1.c.

Queto make si accorge se ex1.c non è stato modificato, e quindi non è necessario ricompilare prima ex1.c.

CFLAGS=-Wall -Wextra
all:ex1.c
\$(CC) \$(CFLAGS) ex1.c -o ex1
clean:
-rm ex1

gcc -Wall -Wextra ex1.c -o ex1

Wall - Wextra

Implementazione della regola. \$(CC) è una variabile di ambiente che identifica il compilatore predefinito di sistema. \$(CFLAGS) richiama la variabile precedentemente dichiarata. ex1.c è il file di input al compilatore. -o è un flag che indica come chiamare l'eseguibile compilato (default: a.out)

clean non è una regola di default: cancella e ripulisce l'ambiente di compilazione;

con il ~~clean~~ dico a make che, se si verifica un errore, ovvero rimuovere il file ex1 ancora prima di averlo creato, considera quel comando come se sia andato a buon fine;

NELLA PRATICA: tutto quel codice lo mettiamo in un file chiamato "Makefile".

e poi, quando ~~makefile~~ ed EX1.C sono nello stesso file, invochiamo il programma

MAKE.

>> make ↙ non gli passo nulla

>> ls

ex1 ex1.c

↳ EXECUTABLE

DEVE TROVARE DENTRO LA STESSA
CARTELLA UN FILE che si
CHIAMA MAKEFILE!

ORA, SCRIVERE MAKE O MAKEALL, IN QUESTO CASO, È UGUALE;

POSSO SCRIVERE: >> **MAKE CLEAN**, dove "clean" è il nome della Regla.

GCC -Wall -Wextra file.c -o file1
↓ ↓
GENERA WARNING

ATTENZIONE: SE ESEGUO due volte "MAKE CLEAN" e non c'è niente da eliminare, grazie a -, viene ignorato l'errore.
Se tolgo -, quando mostro l'errore, si avvia il MAKE!