Các lệnh trong linux:

Tạo ra file .o:

**gcc <direction\_to\_.c\_file> -o < direction\_to\_.o\_file >**

Tạo ra file .exe:

**gcc <direction\_to\_.o\_file> -o < direction\_to\_.exe\_file >**

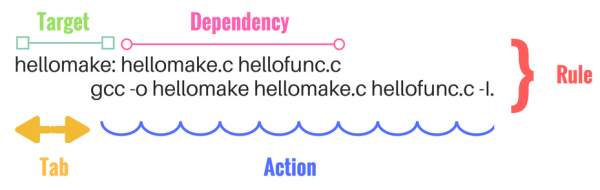
Chay 1 file:

**./<direction\_to\_.c\_file>**

In ra màn hình:

**echo <printing\_value>**

Rule trong makefile.



Target: là tên của file đc sinh ra hoặc là tên của 1 tổ hợp action.

Dependency: là các file được sử dụng như input của rule.

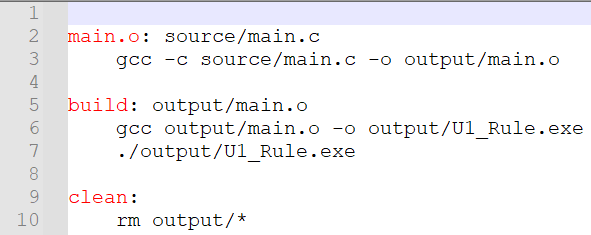
Action: là các lệnh.

Để chạy 1 rule trong makefile ta gõ lệnh:

**make <rule\_name>**

Nếu ta chỉ gõ make thì hệ thống sẽ chạy rule đầu tiên trong makefile.

VD1: Ta có makefile như bên dưới và file main.c được chứa trong thư mục source.



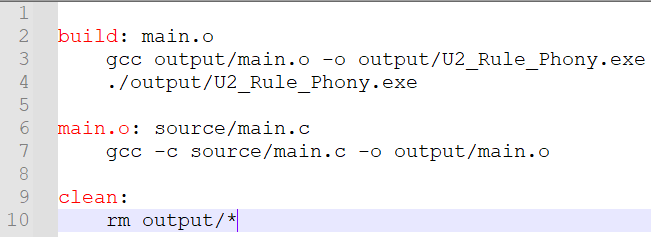
Khi gõ lệnh **make main.o** sẽ tạo ra file main.o trong thư mục output.

Khi gõ lệnh **make build** sao lệnh **make main.o** thì file U1\_Rule.exe sẽ được tạo ra trong thư mục output.

Khi gõ lệnh **make clean** thì các file trong thư mục output sẽ bị xóa sạch.

Khi gõ lệnh **make** thì chỉ có rule main.o được thực thi.

VD2: Ta có makefile như bên dưới và file main.c được chứa trong thư mục source.



Khi gõ lệnh **make** thì hệ thống sẽ thực hiện rule build, rule build có dependency là rule main.o nên rule main.o sẽ được thực hiện trước, sau đó đến rule build.

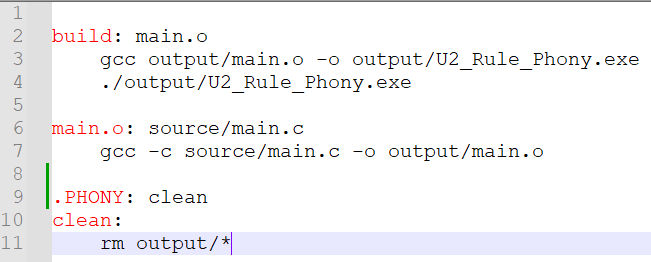
PHONY target.

Khi trong thư mục chứa makefile tồn tại 1 file có tên trùng với target của 1 rule trong makefile thì khi gõ lệnh chạy rule đó thì sẽ bị báo lỗi.

VD: Trong thư mục chứa makefile chứa 1 file tên clean (không có phần mở rộng). Khi gõ lệnh **make clean** thì sẽ bị báo lỗi.



Để khắc phục tình trạng này, người at them dòng code sau vào:



Biến và cách truy xuất giá trị của biến.

Cú pháp tạo biến:

Biến có thể thay đổi giá trị khi chạy.

**<variable\_name> = <variable\_value>**

Biến giữ nguyên giá trị ban đầu khi chạy.

**<variable\_name> := <variable\_value>**

Truy xuất giá trị của 1 biến: **$(<variable\_name>)**