make 命令執行时，需要一个 Makefile 文件以告诉 make 命令需要怎么样的去编译和链接程序。基本规则：

1. 如果这个工程没有编译过，则所有 C 文件都要编译并被链接。
2. 如果这个工程的某几个 C 文件被修改，则只编译被修改的 C 文件，并链接目标程序。
3. 如果这个工程的头文件被修改，则需要编译引用了这个头文件的 C 文件，并链接目标程序。

Makefile 规则：

```
target ... : prerequisites ...
             command
             ...
             ...
```

target 是一个目标文件，可以是 Object file 也可以是可执行文件，还可以是一个标签 (Label)。prerequisites 是要生成 target 所需要的文件或目标。command 是 make 需要执行的命令（任意的 Shell 命令）也就是，target 这一个或多个的目标文件依赖于 prerequisites 中的文件，其生成规则定义在 command 中。再即是，prerequisites 中如果有一个以上的文件比 target 文件新，则 command 命令就会执行。定义好依赖关系后，下一行定义了如何生成目标文件的操作系统命令，一定要以一个 Tab 键作为开头。make 并不关心命令是怎么工作的，它只是执行所定义的命令。clean 不是一个文件，而是一个动作的名字，像 C 语言的 label 一样，其冒号后面没有文件，则 make 就不会自动去找文件的依赖性，也就不会自动执行其后所定义的命令。要执行其后的命令，就要在 make 命令后显式的指出这个 label 的名字。这样就可以在一个 Makefile 文件中定义不用的编译或是和编译无关的命令，比如程序的打包、备份等等。

1 工作原理

默认的方式下 (只输入 make)，那么：

1. make 在当前目录下查找 Makefile 或 makefile 文件。
2. 如果找到，就会找到文件中的第一个目标文件，并把这个文件作为最终的目标文件。

3. 如果目标文件不存在，或是目标文件所依赖的文件的修改时间比目标文件新，则执行后面定义的命令来生成目标文件。

4. 如果目标文件所依赖的.o 文件存在，make 会在当前文件中查找.o 文件所依赖的文件，如果找到则再根据那一个规则生成.o 文件。

5. 最后一定是源代码文件，于是会 make 生成.o 文件，再用.o 文件执行 make，生成目标文件。

这就是 make 的依赖性，一层一层的去找文件的依赖关系，直到最后编译出第一个目标文件。注意 make 只解决依赖关系，而不管所定义的命令的错误或编译是否成功。

2 使用变量

如声明一个变量，OBJECTS, 则在引用时可以

```
OBJECTS = main.o kbd.o command.o display.o \
          insert.o
edit: $(OBJECTS)
   cc -c edit $(OBJECTS)
.....
clean:
   rm edit $(OBJECTS)
```

3 让 make 自动推导

只要 make 看到一个.o 文件，就会把相应的.c 文件加到.o 文件的依赖中。例如会自动把 main.c 加到 main.o 的依赖文件中。于是类似上面的 Makefile 又可以省略.o 和.c 文件同名的。

4 另类的 Makefile

5 伪目标

.PHONY: clean 表示 clean 是一个伪目标，不成文的规定是 clean 从来都放在 Makefile 文件的最后。

6 细节

Makefile 里主要包含了 5 个东西：显式规则，隐晦规则，变量定义，文件指示和注释。

1. 显式规则：显式规则说明了如何生成一个或多个目标文件文件。由 Makefile 的书写者显式的指出要生成的文件，文件的依赖文件和生成的命令。

2. 隐晦规则：由于 make 有自动推导功能，所以隐晦规则可以允许比较粗糙简略的书写 Makefile，这是 make 所支持的。

3. 变量的定义：在 Makefile 中要定义一系列的变量，变量一般都是字符串，类似 C 语言的宏，当 Makefile 被执行时，其中的变量都会被扩展到相应的引用位置上。

4. 文件指示：包括三部分，一个是在一个 Makefile 中引用另一个 Makefile，就像 C 语言的 #include；另一个是根据某些情况指定 Makefile 中的有效部分，就像 C 语言的预编译 #if；另一个是定义一个多行的命令。

5. 注释：Makefile 只有行注释，使用 # 作为注释标记。

6.1 文件名

默认查找顺序：GNUmakefile,makefile, Makefile 用的最多的是 Makefile。如果要指定 makefile 文件，使用 -f 选项。

6.2 引用

使用 include 关键词把别的 Makefile 包含进来。

```
include <filename>
```

filename 可以是当前操作系统 Shell 的文件模式（可以包含路径和通配符）。

6.3 环境变量 MAKEFILES

如果当前环境变量中定义了 MAKEFILES, 那么 make 会把这个变量中的值做一个类似于 include 的动作。这个变量的值是其他的 Makefile，用空格分隔。但是从这个环境变量中引入的 Makefile 的目标不回起作用。如果环境变量中的文件发现错误，make 也不会理会。

6.4 工作方式

1. 读入所有的 Makefile
2. 读入被 include 的其他 Makefile
3. 初始化文件中的变量
4. 推导隐晦规则，并分析所有规则
5. 为所有的目标文件创建依赖关系链
6. 根据依赖关系，决定哪些目标需要重新生成
7. 执行生成命令