

Makefile 文件的作用及编写方法

1. Makefile 文件的作用

make 命令执行时，需要一个 makefile 文件，以告诉 make 命令如何去编译和链接程序。

2. Makefile 文件的基本语法

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

target 也就是一个目标文件，可以是 **Object File**，也可以是执行文件。还可以是一个标签（**Label**），对于标签这种特性，在后续的“伪目标”章节中会有叙述。

prerequisites 就是，要生成那个 **target** 所需要的文件或是目标。

command 也就是 **make** 需要执行的命令。（任意的 **Shell** 命令）

这是一个文件的依赖关系，也就是说，**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。说白了就是说，**prerequisites** 中如果有一个以上的文件比 **target** 文件要新的话，**command** 所定义的命令就会被执行。这就是 **Makefile** 的规则。也就是 **Makefile** 中最核心的内容。

3.给出下列几种常用情况的 Makefile 文件的写法

3.1

```
[root@localhost home]# cd ./1551265-000102/  
[root@localhost 1551265-000102]# mkdir 01  
[root@localhost 1551265-000102]# mkdir 02  
[root@localhost 1551265-000102]# mkdir 03
```

3.2

```
[root@localhost home]# cd ./1551265-000102/
[root@localhost 1551265-000102]# mkdir 01
[root@localhost 1551265-000102]# mkdir 02
[root@localhost 1551265-000102]# mkdir 03
[root@localhost 1551265-000102]# cd ./01
[root@localhost 01]# touch test1.c
[root@localhost 01]# touch test2.c
[root@localhost 01]# touch test3.c
[root@localhost 01]# ll
总用量 0
-rw-r--r--. 1 root root 0 9月 26 16:26 test1.c
-rw-r--r--. 1 root root 0 9月 26 16:26 test2.c
-rw-r--r--. 1 root root 0 9月 26 16:26 test3.c
```

```
[root@localhost 01]# vi test1.c
[root@localhost 01]# vi test2.c
[root@localhost 01]# vi test3.c
```

输入代码后保存退出

编辑一个 Makefile 文件

```
test :test1.o test2.o test3.o
    cc -o test test1.o test2.o test3.o
test1.o: test1.c
    cc -c test1.c
test2.o: test2.c
    cc -c test2.c
test3.o: test3.c
    cc -c test3.c
clean:
    rm test test1.o test2.o test3.o
~
~
```

保存退出后 make 一下

可以看到编译的中间文件和执行程序已经产生

```
[root@localhost 01]# make
cc -o test test1.o test2.o test3.o
[root@localhost 01]# ll
总用量 40
-rw-r--r--. 1 root root 202 9月 26 18:16 makefile
-rwxr-xr-x. 1 root root 8640 9月 26 18:16 test
-rw-r--r--. 1 root root 73 9月 26 16:27 test1.c
-rw-r--r--. 1 root root 1496 9月 26 18:16 test1.o
-rw-r--r--. 1 root root 86 9月 26 16:28 test2.c
-rw-r--r--. 1 root root 1496 9月 26 18:16 test2.o
-rw-r--r--. 1 root root 107 9月 26 16:30 test3.c
-rw-r--r--. 1 root root 1424 9月 26 18:16 test3.o
```

随便修改其中的 test1.c 文件

重新 make 并查看产生的中间文件的创建时间

```
[root@localhost 01]# vi test1.c
[root@localhost 01]# make
cc -c test1.c
cc -o test test1.o test2.o test3.o
[root@localhost 01]# ll
总用量 40
-rw-r--r--. 1 root root 202 9月 26 18:16 makefile
-rwxr-xr-x. 1 root root 8640 9月 26 18:18 test
-rw-r--r--. 1 root root 74 9月 26 18:18 test1.c
-rw-r--r--. 1 root root 1504 9月 26 18:18 test1.o
-rw-r--r--. 1 root root 86 9月 26 16:28 test2.c
-rw-r--r--. 1 root root 1496 9月 26 18:16 test2.o
-rw-r--r--. 1 root root 107 9月 26 16:30 test3.c
-rw-r--r--. 1 root root 1424 9月 26 18:16 test3.o
```

执行 make clean

```
[root@localhost 01]# make clean
rm test test1.o test2.o test3.o
[root@localhost 01]# ll
总用量 16
-rw-r--r--. 1 root root 202 9月 26 18:16 makefile
-rw-r--r--. 1 root root 74 9月 26 18:18 test1.c
-rw-r--r--. 1 root root 86 9月 26 16:28 test2.c
-rw-r--r--. 1 root root 107 9月 26 16:30 test3.c
```

3.3

同上一问

Makefile 文件

```
makefile
test :test1.o test2.o test3.o
    cc -o test test1.o test2.o test3.o
test1.o: test1.c test.h
    cc -c test1.c
test2.o: test2.c test.h
    cc -c test2.c
test3.o: test3.c test.h
    cc -c test3.c
clean:
    rm test test1.o test2.o test3.o
```

make

```
[root@localhost 02]# make
cc -c test2.c
cc -o test test1.o test2.o test3.o
[root@localhost 02]# ll
总用量 44
-rw-r--r--. 1 root root 224 9月 26 19:15 makefile
-rwxr-xr-x. 1 root root 8640 9月 26 19:46 test
-rw-r--r--. 1 root root 116 9月 26 19:33 test1.c
-rw-r--r--. 1 root root 1568 9月 26 19:33 test1.o
-rw-r--r--. 1 root root 120 9月 26 19:45 test2.c
-rw-r--r--. 1 root root 1568 9月 26 19:46 test2.o
-rw-r--r--. 1 root root 70 9月 26 19:13 test3.c
-rw-r--r--. 1 root root 1424 9月 26 19:15 test3.o
-rw-r--r--. 1 root root 65 9月 26 19:13 test.h
```

3.4

```
1. root@localhost:/home/1551265-000102/03 (ssh)
test1:test1.o
    cc -o test1 test1.o
test2:test2.o
    cc -o test2 test2.o
test3:test3.o
    cc -o test3 test3.o
test1.o: test1.c
    cc -c test1.c
test2.o: test2.c
    cc -c test2.c
test3.o: test3.c
    cc -c test3.c
clean:
    rm test1 test2 test3 test1.o test2.o test3.o

~
~
~
~
~
~
"makefile" 16L, 256C
```

make test1

```
[root@localhost 03]# make test1
cc -c test1.c
cc -o test1 test1.o
[root@localhost 03]# ll
总用量 32
-rw-r--r--. 1 root root 256 9月 26 19:56 makefile
-rwxr-xr-x. 1 root root 8520 9月 26 19:59 test1
-rw-r--r--. 1 root root 88 9月 26 19:50 test1.c
-rw-r--r--. 1 root root 1496 9月 26 19:59 test1.o
-rw-r--r--. 1 root root 88 9月 26 19:51 test2.c
-rw-r--r--. 1 root root 100 9月 26 19:57 test3.c
```

make test3

```

[root@localhost 03]# make test3
cc -c test3.c
cc -o test3 test3.o
[root@localhost 03]# ll
总用量 48
-rw-r--r--. 1 root root 256 9月 26 19:56 makefile
-rwxr-xr-x. 1 root root 8520 9月 26 19:59 test1
-rw-r--r--. 1 root root 88 9月 26 19:50 test1.c
-rw-r--r--. 1 root root 1496 9月 26 19:59 test1.o
-rw-r--r--. 1 root root 88 9月 26 19:51 test2.c
-rwxr-xr-x. 1 root root 8520 9月 26 20:00 test3
-rw-r--r--. 1 root root 100 9月 26 19:57 test3.c
-rw-r--r--. 1 root root 1504 9月 26 20:00 test3.o

```

make test2

```

[root@localhost 03]# make test2
cc -c test2.c
cc -o test2 test2.o
[root@localhost 03]# ll
总用量 64
-rw-r--r--. 1 root root 256 9月 26 19:56 makefile
-rwxr-xr-x. 1 root root 8520 9月 26 19:59 test1
-rw-r--r--. 1 root root 88 9月 26 19:50 test1.c
-rw-r--r--. 1 root root 1496 9月 26 19:59 test1.o
-rwxr-xr-x. 1 root root 8520 9月 26 20:00 test2
-rw-r--r--. 1 root root 88 9月 26 19:51 test2.c
-rw-r--r--. 1 root root 1496 9月 26 20:00 test2.o
-rwxr-xr-x. 1 root root 8520 9月 26 20:00 test3
-rw-r--r--. 1 root root 100 9月 26 19:57 test3.c
-rw-r--r--. 1 root root 1504 9月 26 20:00 test3.o

```


make clean

```
[root@localhost 03]# make clean
rm test1 test2 test3 test1.o test2.o test3.o
[root@localhost 03]# ll
总用量 16
-rw-r--r--. 1 root root 256 9月 26 19:56 makefile
-rw-r--r--. 1 root root 88 9月 26 19:50 test1.c
-rw-r--r--. 1 root root 88 9月 26 19:51 test2.c
-rw-r--r--. 1 root root 100 9月 26 19:57 test3.c
```

3.5

编辑 Makefile 文件

循环至每个文件夹下单独执行

这里在进入 03 文件夹时 无法使其执行 make1 make2 make3 只能执行一次

make 但是原来的 Makefile 执行 make 操作只会生成 make1

所以加入第一行 all 不截图了

```
root@RHEL-zby:/home/1551265-000102 (ssh)
SUB = `ls -d */`
subdirs:
    for dir in $(SUB); do $(MAKE) -C $$dir; done
clean:
    for dir in $(SUB); do $(MAKE) clean -C $$dir; done
```

大功告成了？

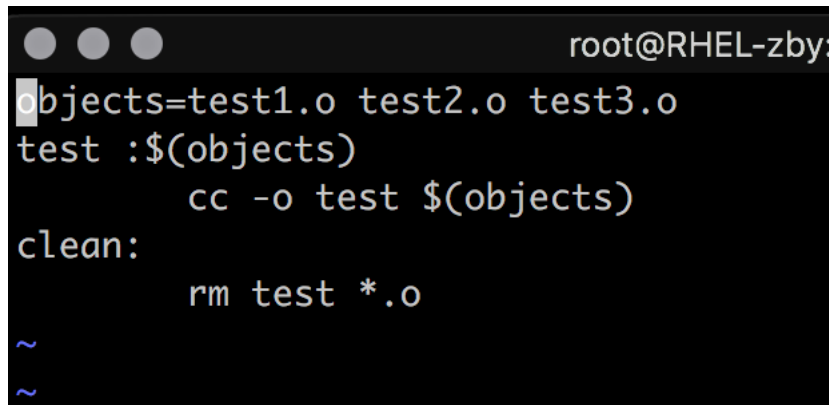
看到这行字...

编译器用 GCC/C++/g++均可

有多种 makefile 的写法，在正确的前提下，每位同学根据自己写法的不同得分可能略有差异

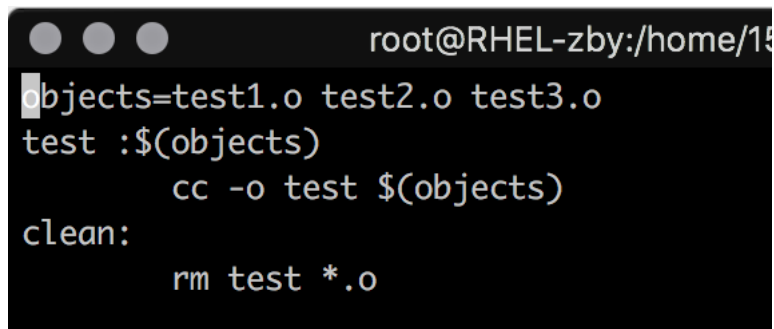
好吧 看下确实有很多值得优化的地方

01/makefile



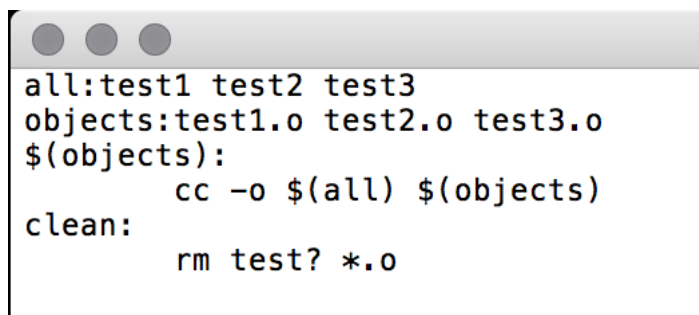
```
root@RHEL-zby:~  
objects=test1.o test2.o test3.o  
test :$(objects)  
    cc -o test $(objects)  
clean:  
    rm test *.o  
~  
~
```

02/makefile



```
root@RHEL-zby:/home/15  
objects=test1.o test2.o test3.o  
test :$(objects)  
    cc -o test $(objects)  
clean:  
    rm test *.o
```

03/makefile



```
all:test1 test2 test3  
objects:test1.o test2.o test3.o  
$(objects):  
    cc -o $(all) $(objects)  
clean:  
    rm test? *.o
```

差不多就这样吧

```
[root@RHEL-zby 1551265-000102]# make
for dir in `ls -d */`; do make -C $dir; done
make[1]: 进入目录“/home/1551265-000102/01”
cc      -c -o test1.o test1.c
cc      -c -o test2.o test2.c
cc      -c -o test3.o test3.c
cc -o test test1.o test2.o test3.o
make[1]: 离开目录“/home/1551265-000102/01”
make[1]: 进入目录“/home/1551265-000102/02”
cc      -c -o test1.o test1.c
cc      -c -o test2.o test2.c
cc      -c -o test3.o test3.c
cc -o test test1.o test2.o test3.o
make[1]: 离开目录“/home/1551265-000102/02”
make[1]: 进入目录“/home/1551265-000102/03”
cc      -c -o test1.o test1.c
cc  test1.o      -o test1
cc      -c -o test2.o test2.c
cc  test2.o      -o test2
cc      -c -o test3.o test3.c
cc  test3.o      -o test3
make[1]: 离开目录“/home/1551265-000102/03”
[root@RHEL-zby 1551265-000102]#
```

make clean

```
[root@RHEL-zby 1551265-000102]# make clean
for dir in `ls -d */`; do make clean -C $dir; done
make[1]: 进入目录“/home/1551265-000102/01”
rm test *.o
make[1]: 离开目录“/home/1551265-000102/01”
make[1]: 进入目录“/home/1551265-000102/02”
rm test *.o
make[1]: 离开目录“/home/1551265-000102/02”
make[1]: 进入目录“/home/1551265-000102/03”
rm test? *.o
make[1]: 离开目录“/home/1551265-000102/03”
[root@RHEL-zby 1551265-000102]#
```