# Makefile 文件的作用及编写方法

## 1. Makefile 文件的作用

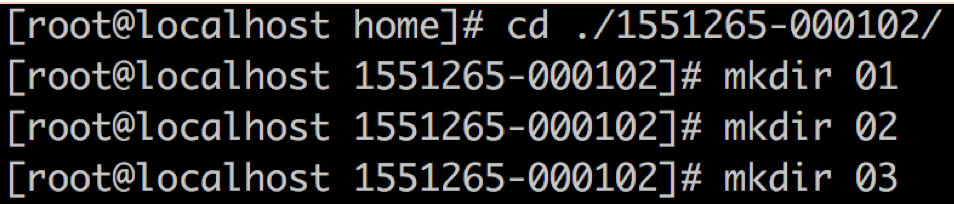
make命令执行时，需要一个 makefile 文件，以告诉make命令如何去编译和链接程序。

## 2. Makefile 文件的基本语法

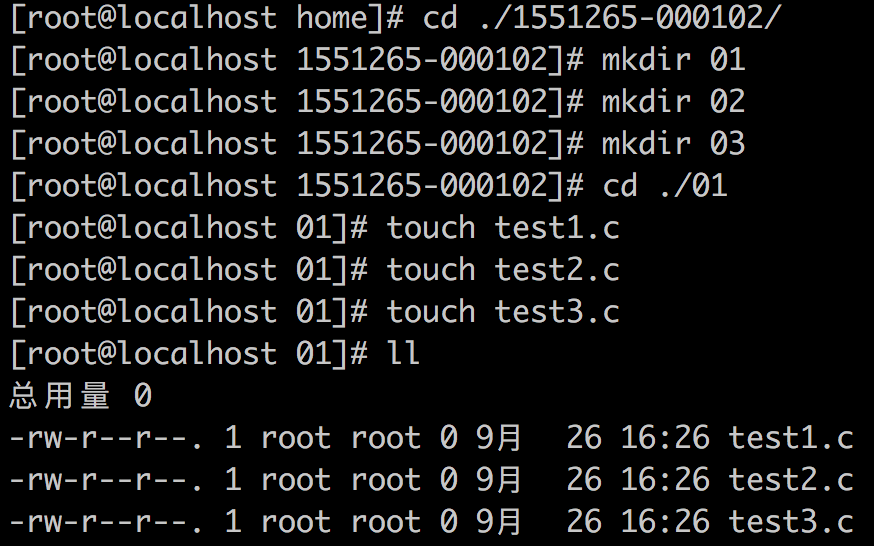
target ... : prerequisites ...  
command  
...  
...  
  
target也就是一个目标文件，可以是Object File，也可以是执行文件。还可以是一个标签（Label），对于标签这种特性，在后续的“伪目标”章节中会有叙述。  
  
prerequisites就是，要生成那个target所需要的文件或是目标。  
  
command也就是make需要执行的命令。（任意的Shell命令）  
  
这是一个文件的依赖关系，也就是说，target这一个或多个的目标文件依赖于prerequisites中的文件，其生成规则定义在command中。说白一点就是说，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。

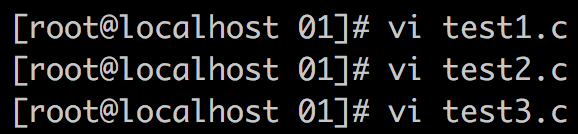
## 3.给出下列几种常用情况的 Makefile 文件的写法

### 3.1



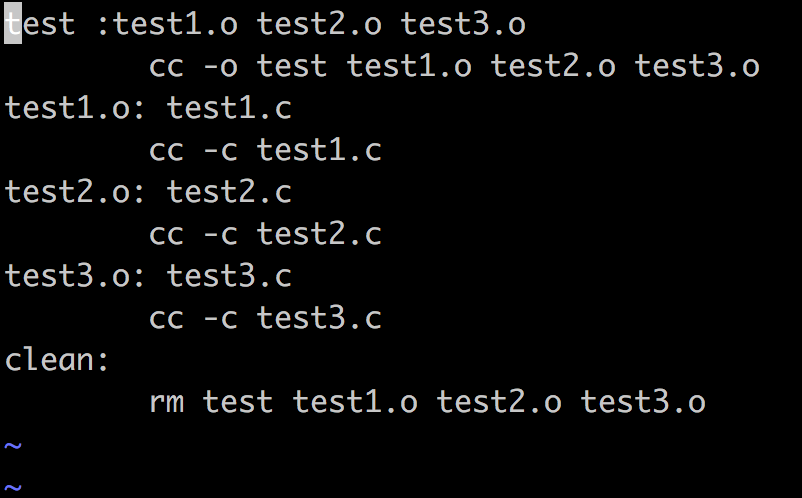
### 3.2





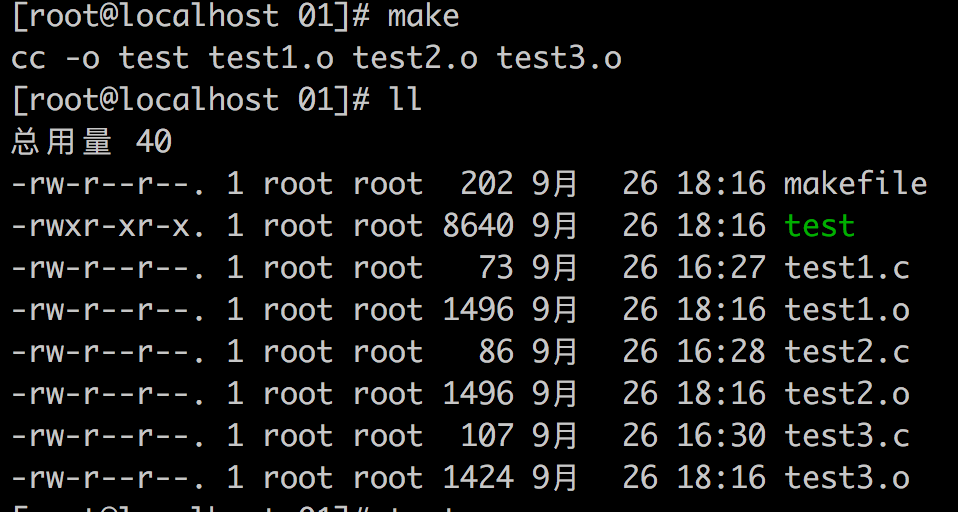
输入代码后保存退出

编辑一个Makefile文件



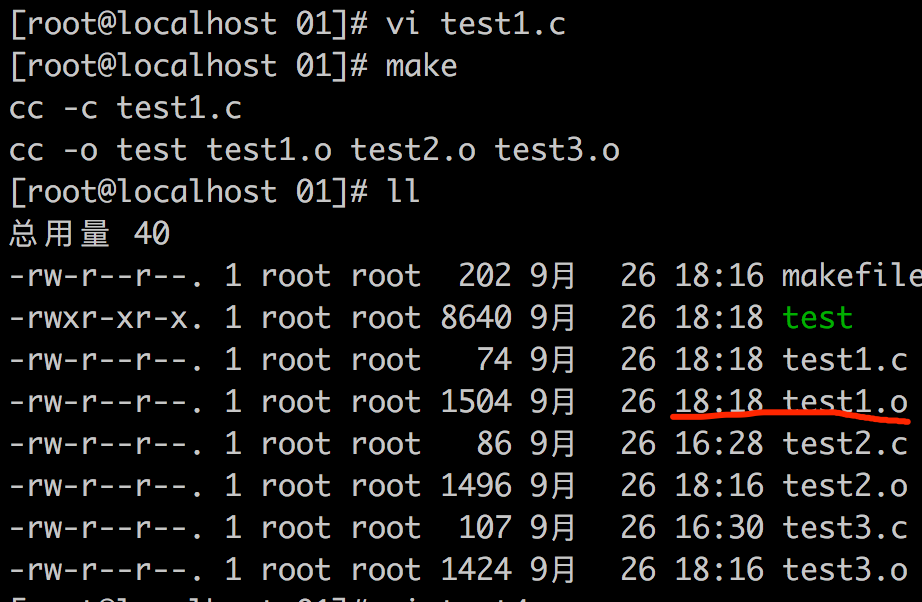
保存退出后make一下

可以看到编译的中间文件和执行程序已经产生

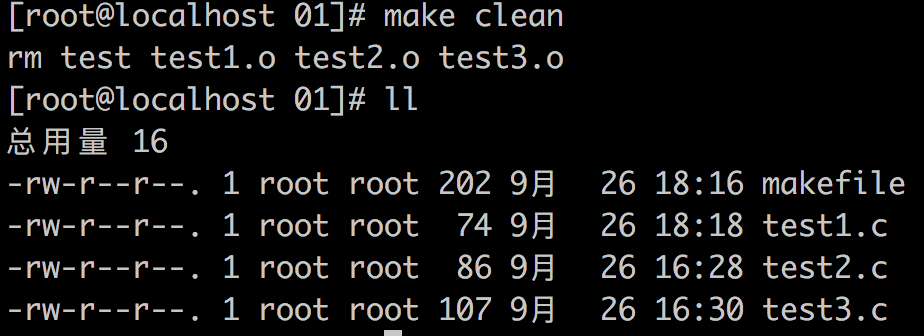


随便修改其中的test1.c文件

重新make并查看产生的中间文件的创建时间



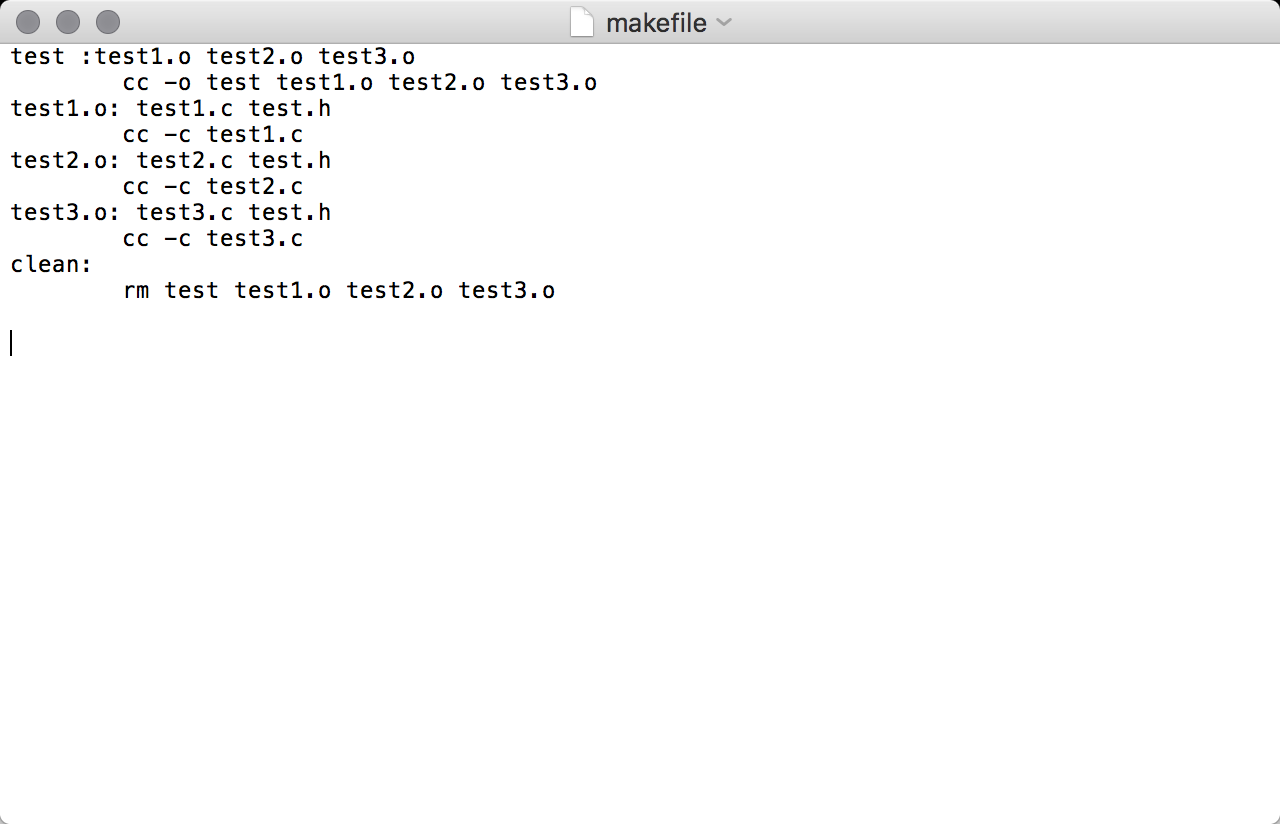
执行make clean



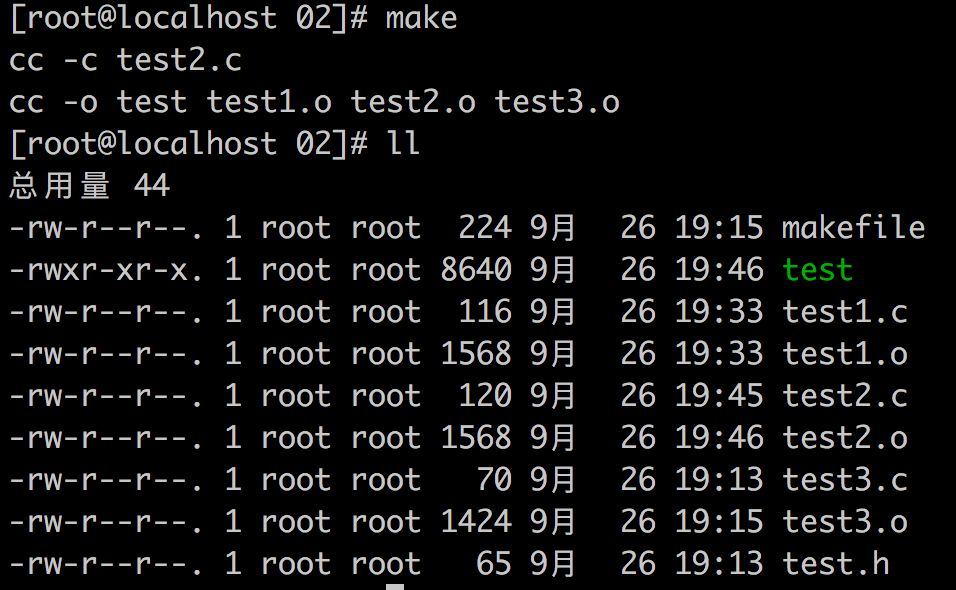
### 3.3

同上一问

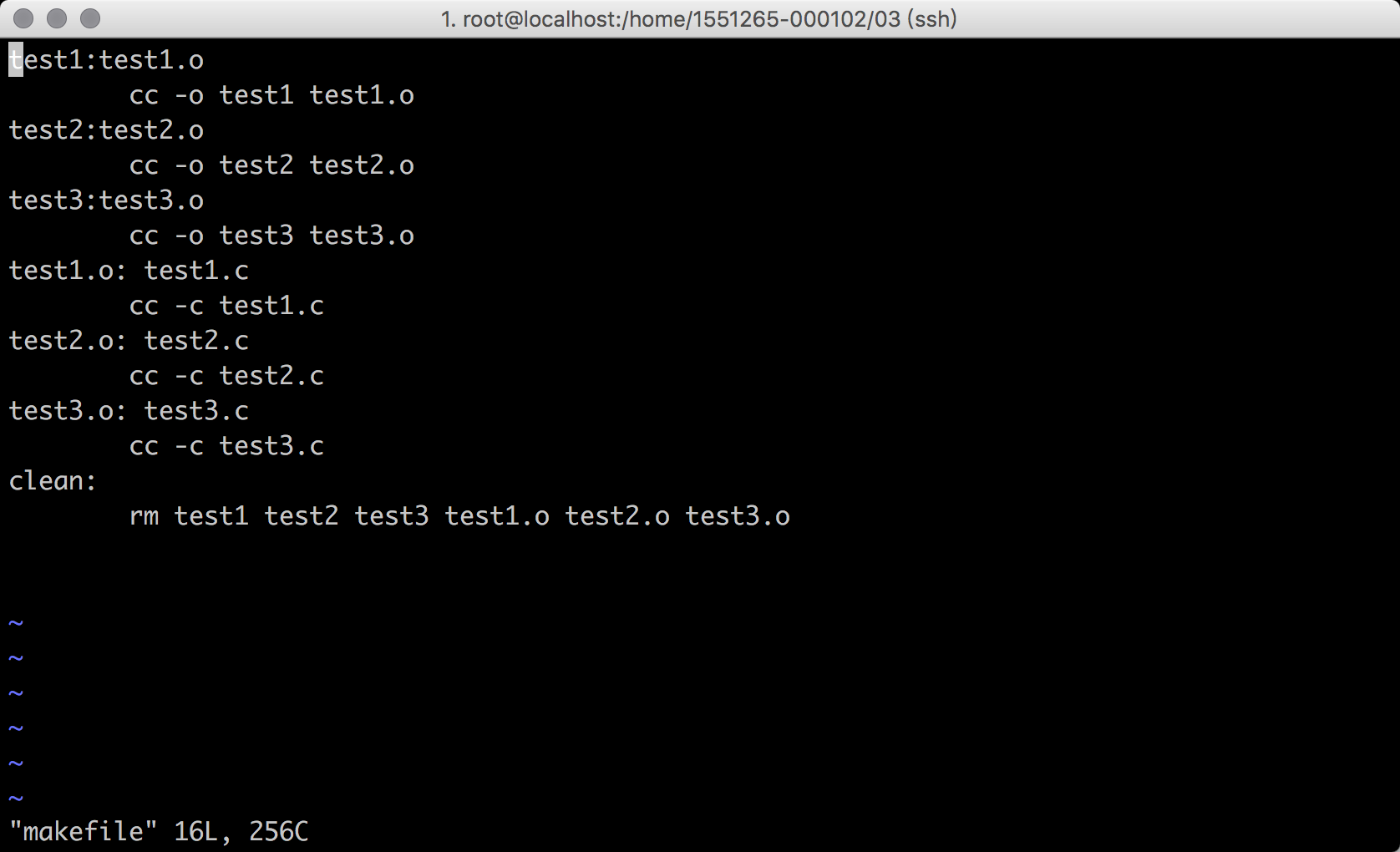
Makefile文件



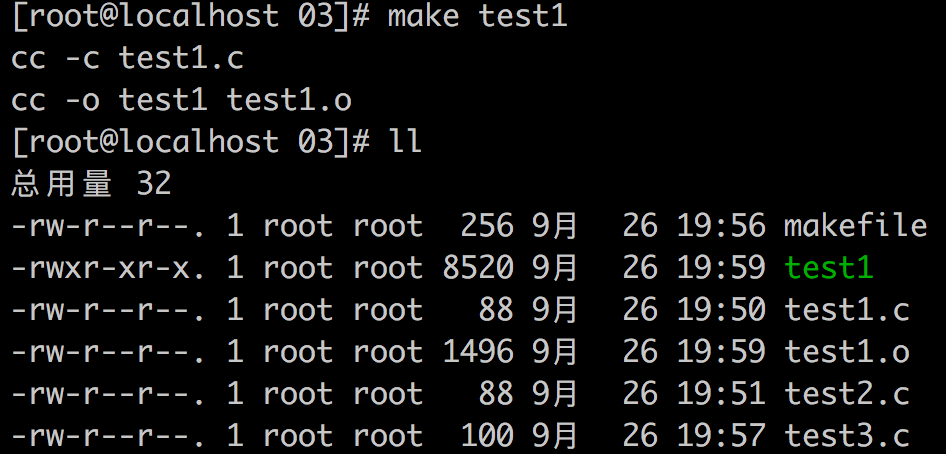
make



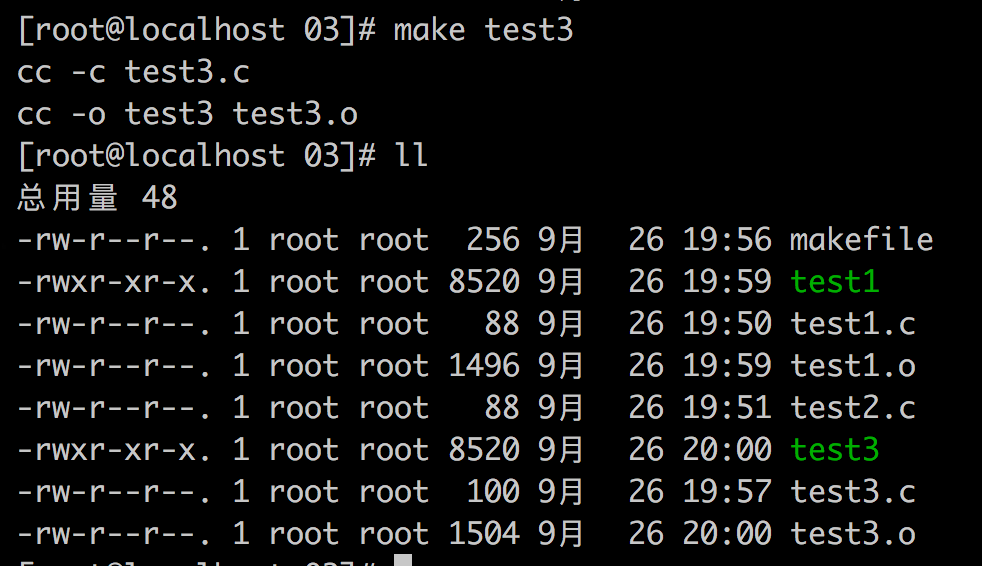
### 3.4



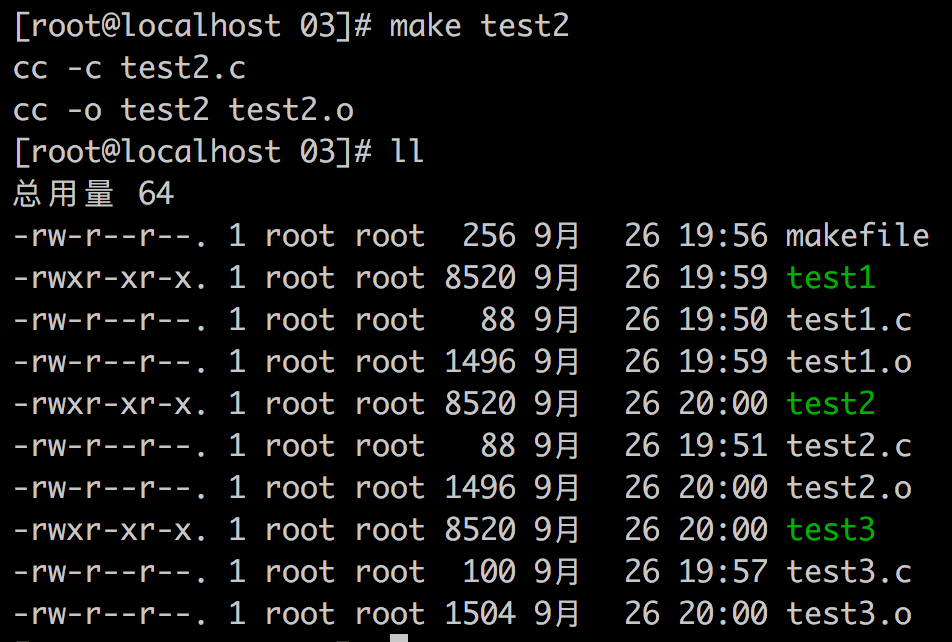
make test1



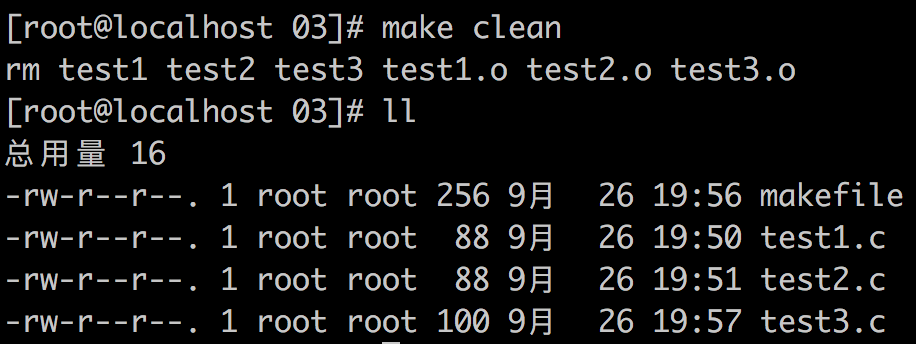
make test3



make test2



make clean



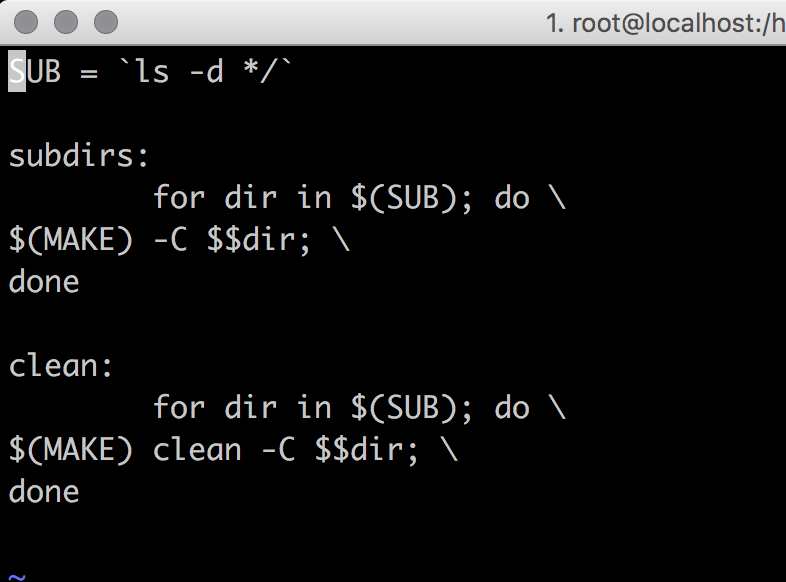
### 3.5

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



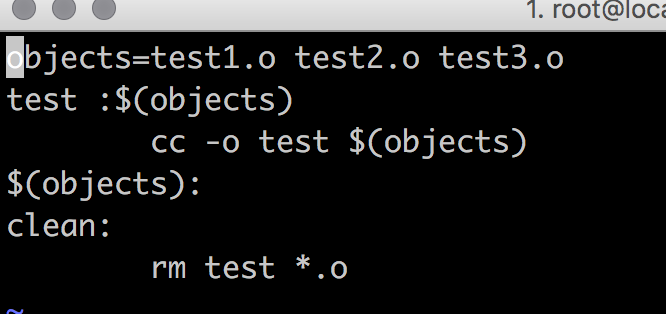
大功告成了?

看到这行字...

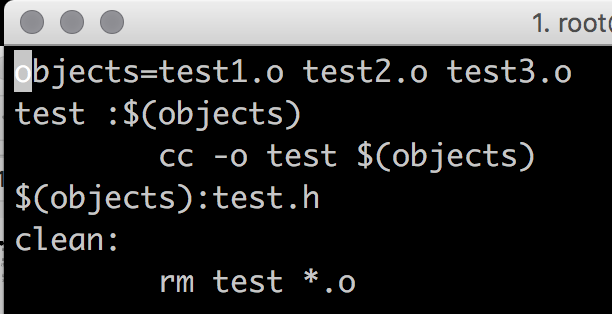


好吧 看下确实有很多值得优化的地方

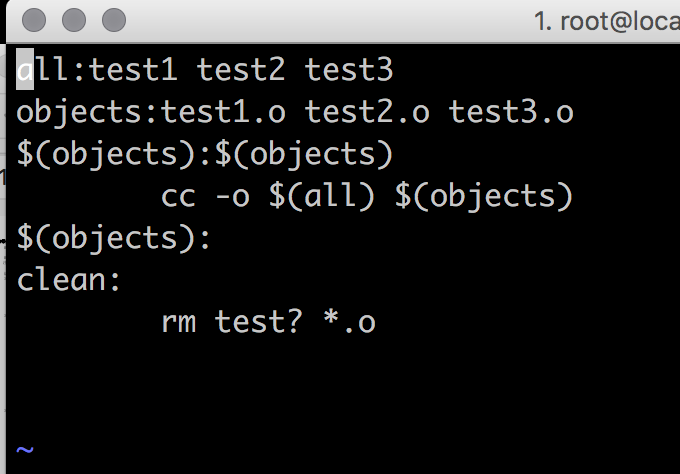
01/makefile



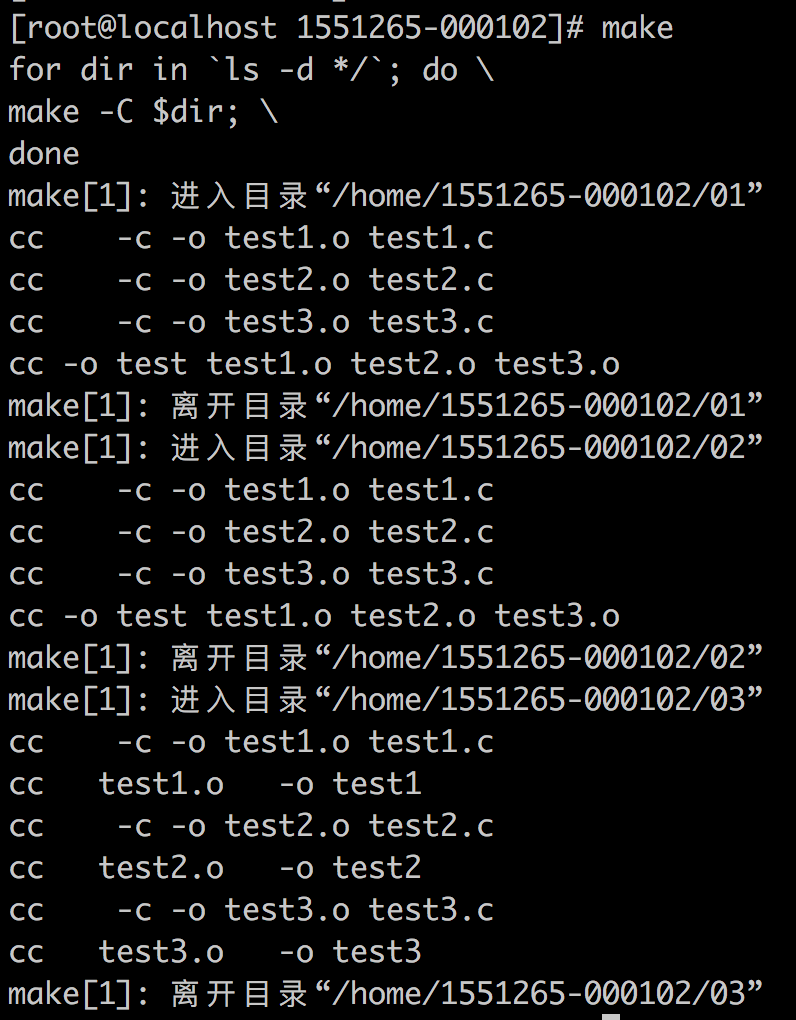
02/makefile



03/makefile



差不多就这样吧



make clean

