同济大学计算机网络

实验报告



姓名： 涂远鹏-1652262

题目： Makefile文件的作用及编写方法

**1. Makefile 文件的作用**

答：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。

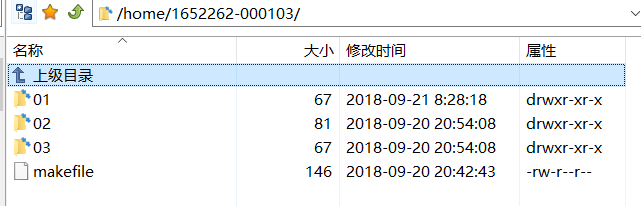
**2. Makefile 文件的基本语法**

答：

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。

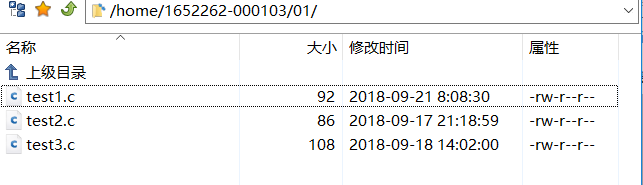
## 3.给出下列几种常用情况的 Makefile 文件的写法

**3.1首先用FlashFXP新建三个文件夹01,02,03：**

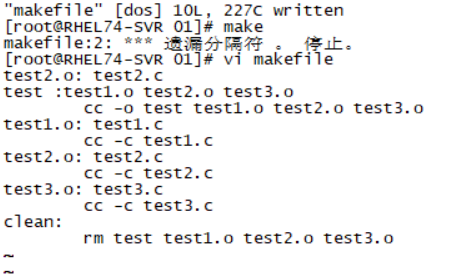


**3.2在子目录01下建立三个源程序文件，分别是test1.c,test2.c,test3.c,要求编写Makefile满足要求：**

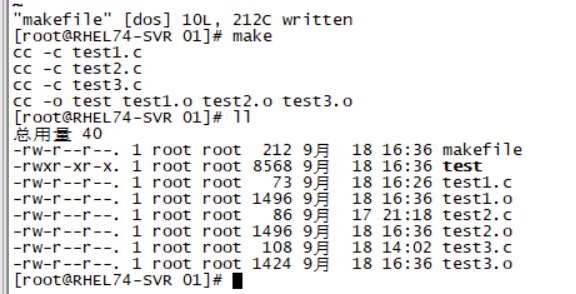
(1)用FlashFXP向01文件夹中放入test1.c test2.c test3.c三个已实现编辑好的文件：



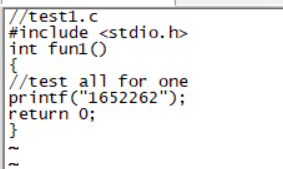
(2)编写test,test1,test2,test3的编译语句并写入生成test可执行文件：



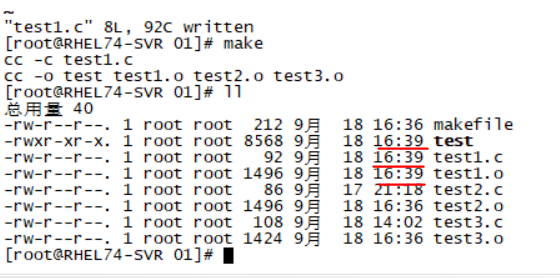
(3)保存退出后运行Makefile程序，可以看到编译的中间文件和执行程序已经产生



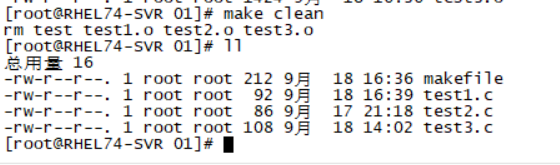
(4)任意修改test1,test2,test3其中的一个文件，此处修改的为test1.c文件



(5)重新执行make操作并查看产生的中间文件的创建时间

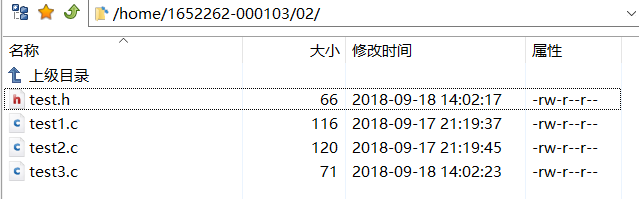


(6)执行make clean，删除生成的.o文件以及生成的test可执行文件：



**3.3在子目录02下建立四个源程序文件+头文件，分别是test1.c,test2.c,test3.c,test.h,要求编写Makefile满足要求：**

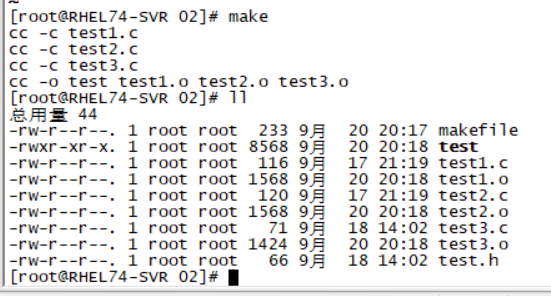
(1)首先创建test1.c,test2.c,test3.c,test.h并输入对应内容



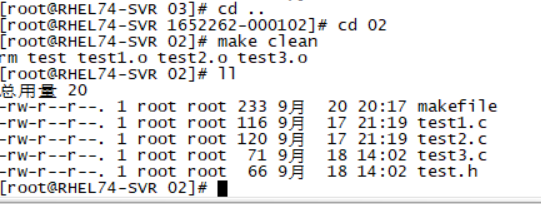
(2)首先编辑makefile文件，使用FlashFXP查看Makefile编写结果：

### 1

(2)执行make操作测试编译结果，成功生成三个.o文件以及一个总的test可执行文件



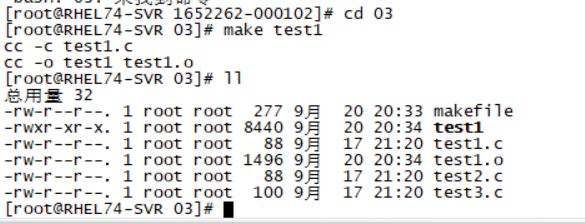
(3)执行make clean操作测试编译结果，成功删除三个.o文件以及一个总的test可执行文件

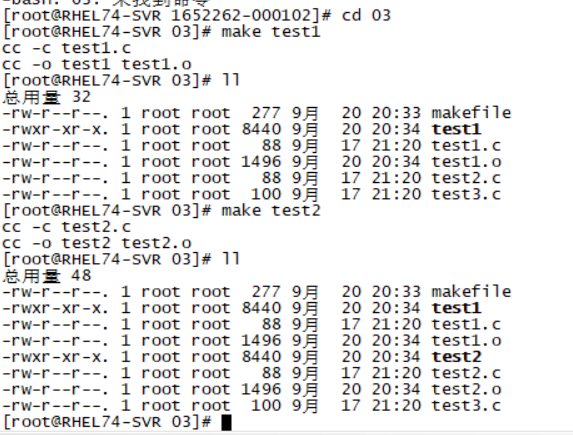


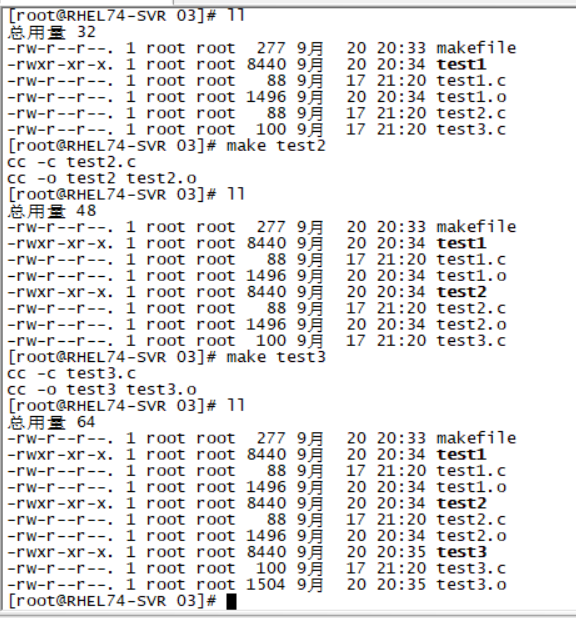
**3.4在子目录03下建立三个源程序文件，分别为test1.c/test2.c/test3.c，要求写一个满足要求的Makefile**

(1)编写的makefile文件如下：

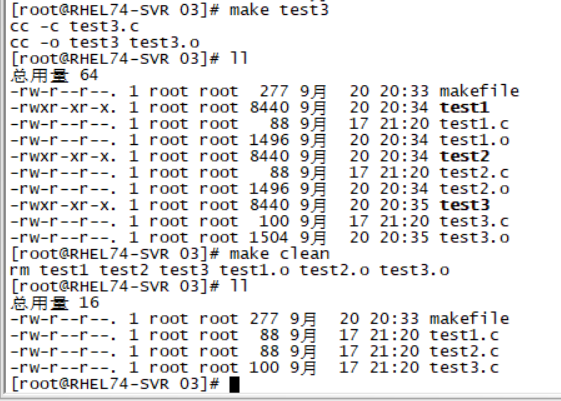
(2)使用Makefile分别单独编译test1.c，test2.c，test3.c执行make test1，make test2，make test3操作：





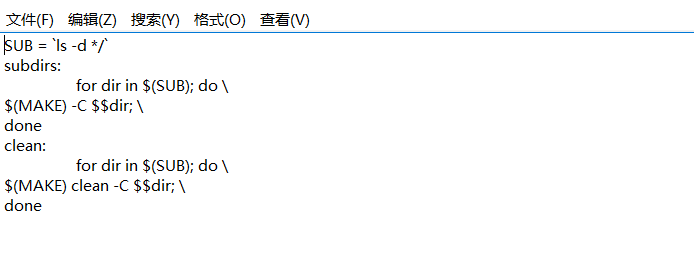


(3)最后测试makefile中的make clean是否正常，执行前后结果如下，结果正确

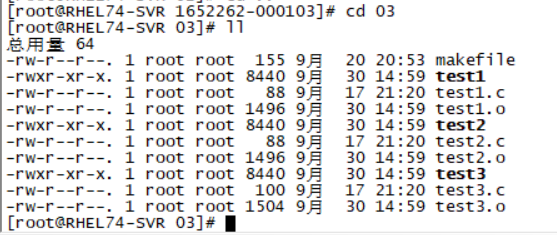
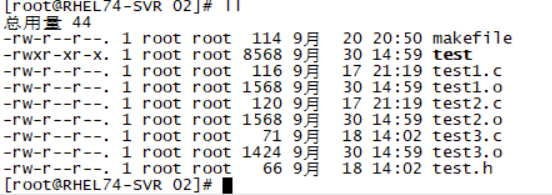
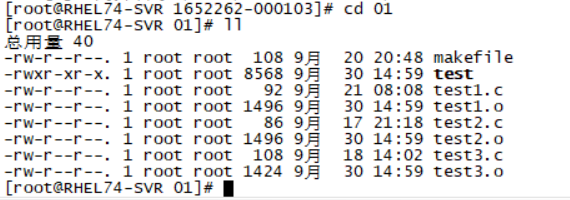
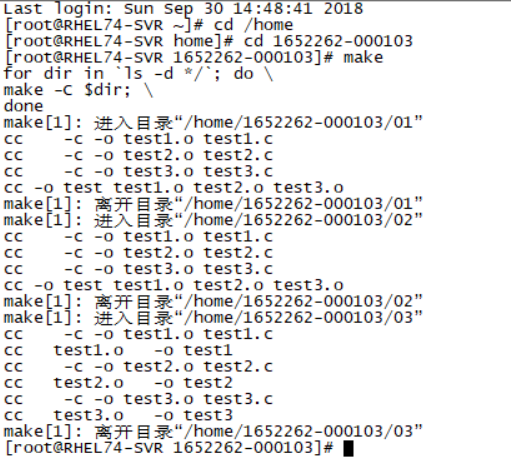


**3.5在1652262-000103目录下写一个满足下列要求的makefile文件：**

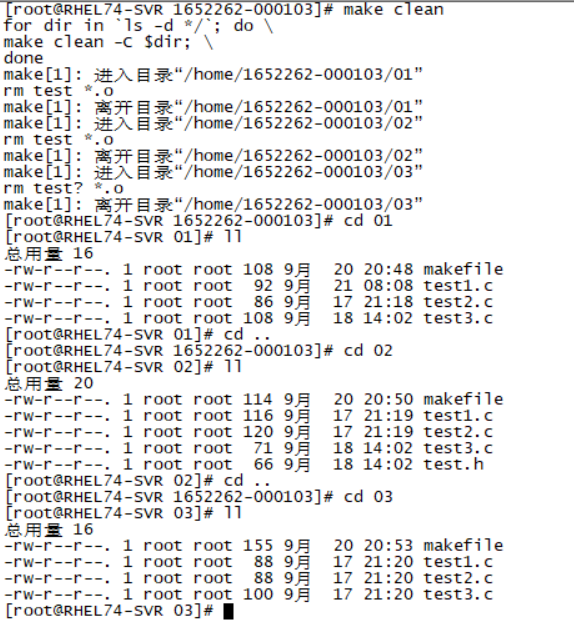
(1)编辑Makefile文件，循环至每个文件夹下单独执行，这里在进入03文件夹时 无法使其执行make1 make2 make3 只能执行一次make 但是原来的Makefile执行make操作只会生成make1，所以加入第一行all



(2)执行Make操作测试，显示对应编译结果，并查看01/02/03三个文件夹下生成的可执行文件数目分别为1个，1个，3个:



(3)执行Make clean操作测试，显示将生成的.o文件以及可执行文件均删除，仅保留C语言程序:

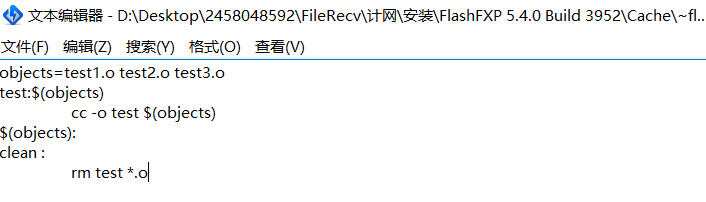


**3.6优化部分**

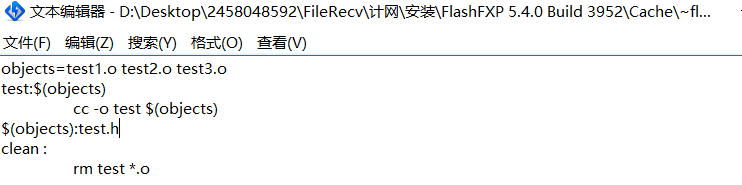
(1)利用object将三个.o的输出文件放在一起，到时候执行的时候只需要按照object一个个进行make就行而不用写三个类似的语句

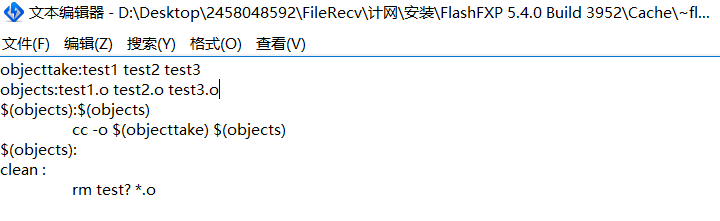
注：`为反引号，被反引号引起来的被shell解释为命令行

01/makefile显示如下

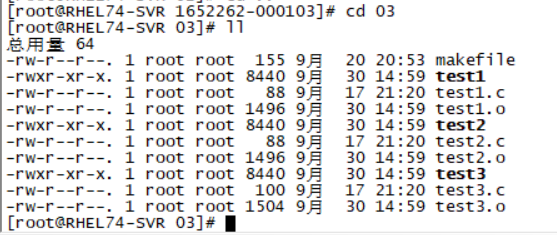
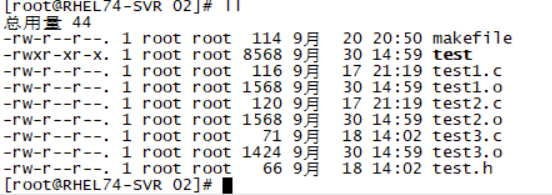
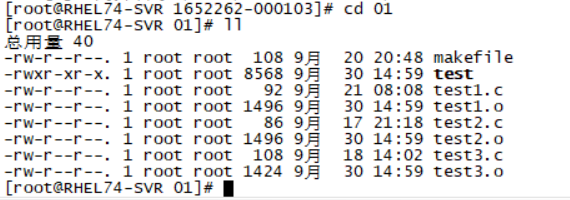
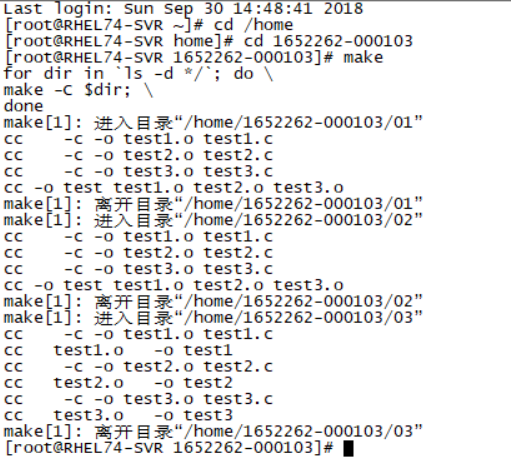


02/makefile显示如下



03/makefile显示如下

(2)执行Make操作测试，显示对应编译结果，并查看01/02/03三个文件夹下生成的可执行文件数目分别为1个，1个，3个:



(3)执行Make clean操作测试，显示将生成的.o文件以及可执行文件均删除，仅保留C语言程序:

