实验题目：Shell 编程

姓名：王波 学号：19122557 实验日期：2021年10月28日

**Shell 编程**

**一、实验环境**

Vmware centOS 7

**二、实验目的**

（1）掌握vi 的三种工作方式，熟悉vi 编辑程序的使用；

（2）学习Shell 程序设计方法，掌握编程要领。

**三、实验内容**

（1）学习使用vi 编辑程序

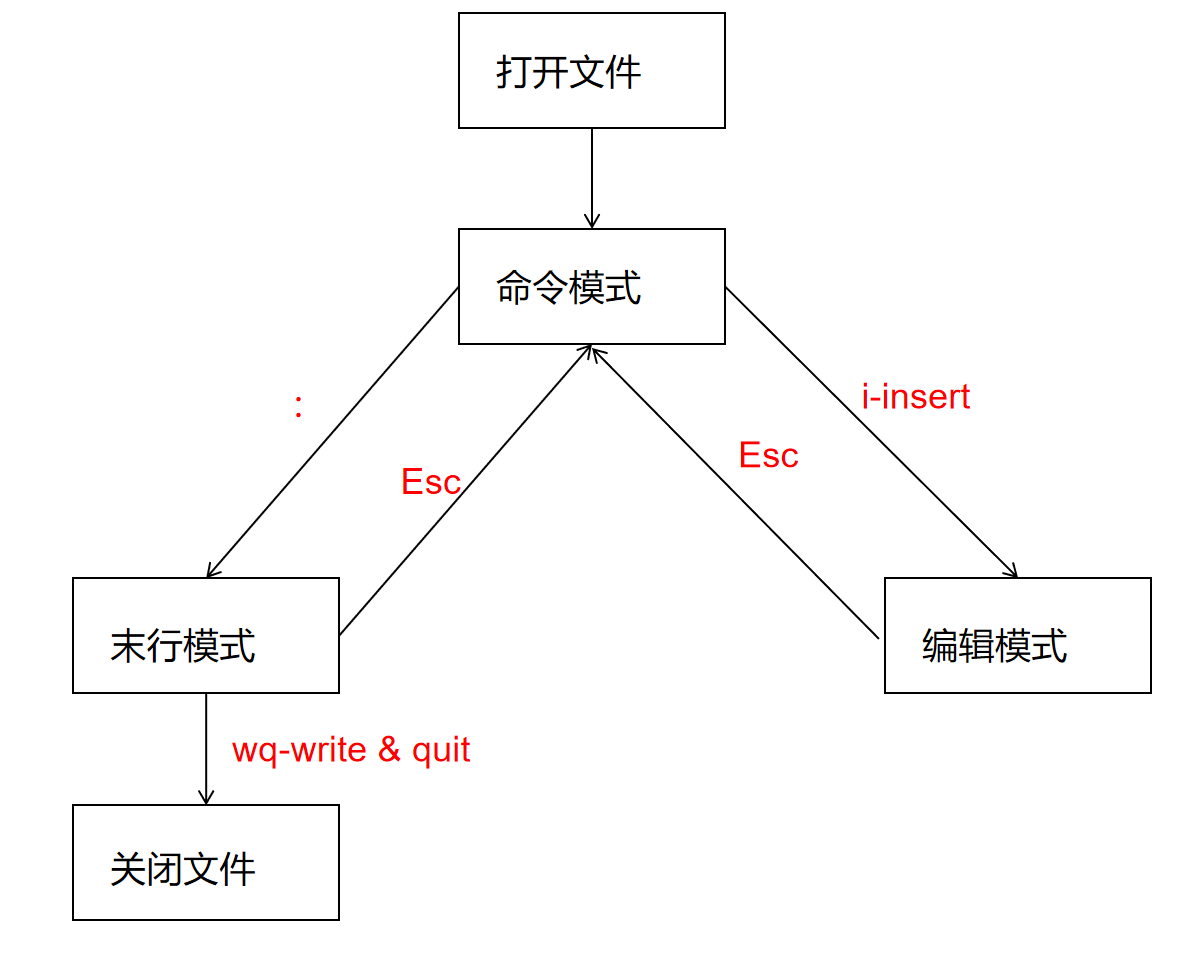
（2）编写Shell 程序；

（3）将程序文件设置为可执行文件（用chmod 命令）；

（4）在命令行方式中运行Shell 程序。

**四、实验步骤**

**1.** 按本《实验指导》第三部分的内容，熟悉vi的三种工作方式，熟悉使用各种编辑功能。 思考：试一试vi 的三种工作方式各用在何时？用什么命令进入插入方式？怎样退出插入方式？文件怎样存盘？注意存盘后的提示信息。



①命令模式：

1. 在该模式中，可以输入命令来执行许多种功能；
2. 打开文件首先进入命令模式，它是使用vim编辑器的入口

②末行模式：

1. 将文件保存或退出vi，也可以设置编辑环境，如寻找字符串、列出行号等；
2. 末行模式是vim编辑器对的出口，要退出vim，必须要在末行模式下。

③编辑模式：可以对文本进行编辑操作。

2. 创建和执行Shell 程序

用前面介绍的Vi或其他文本编辑器编写Shell程序，并将文件以文本文件方式保存在相应的目录中。 用chmod 将文件的权限设置为可执行模式，如若文件名为shdemo.h,则命令如下：

$ chmod 755 shdemo.h

(文件主可读、写、执行，同组人和其他人可读和执行)

在提示符后执行Shell 程序：

$ shdemo.h （直接键入程序文件名执行）

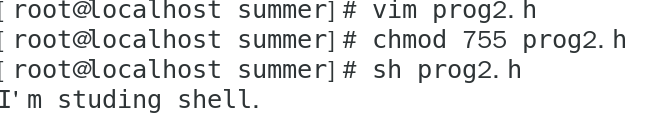
或 $ sh shdemo.h （执行Shell 程序）

或 $ .shdemo.h （没有设置权限时可用点号引导）

【代码】



【运行结果】



1. 用vi编写《实验指导》“第四部分Shell程序设计”中的例1，练习内部变量和位置参数的用法。

用chmod将文件的权限设置为可执行模式，并在提示符后键入命令行：

$./prog1. 或 $sh prog1. #有一个参数

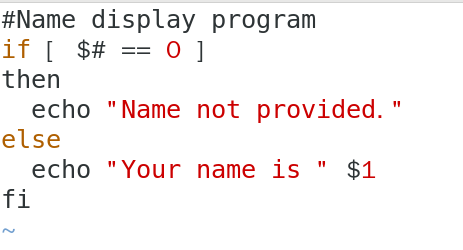
屏幕显示: Name not provided

在提示符后键入命令行：

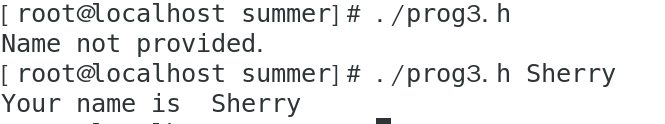
$./prog1.h Theodore

屏幕显示: Your name is Theodore #引用$1参数的效果

【代码】

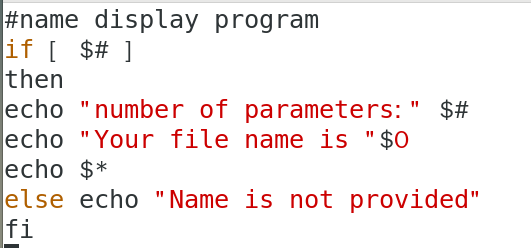


【运行结果】

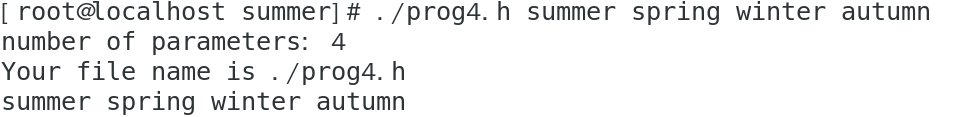


1. 进一步修改上一题中的程序，要求显示参数个数、程序名字，并逐个显示参数。

【代码】

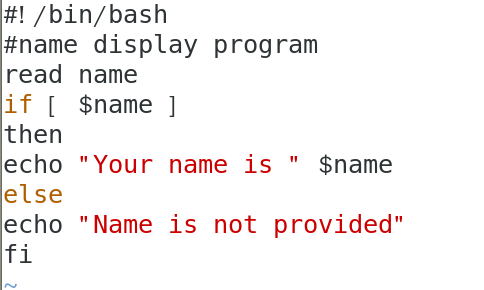


【运行结果】



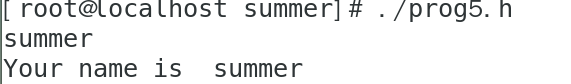
1. 修改例1程序，用read命令接受键盘输入。若没有输入显示第一种提示，否则第二种提示。

【代码】

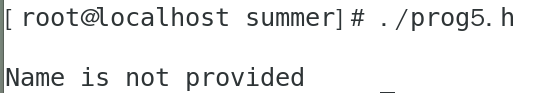


【运行结果】

有输入：

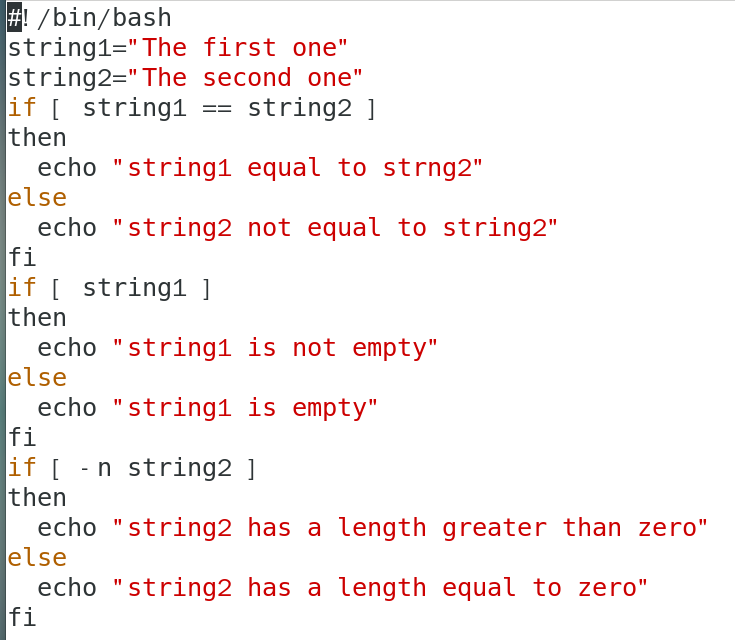


没有输入：

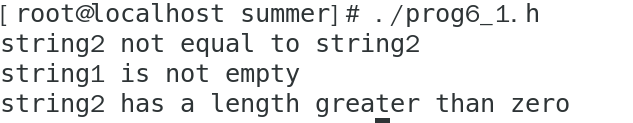


1. 用vi 编写《实验指导》“第四部分 Shell程序设计”中的例2、例3，练习字符串比较运算符、数据比较运算符和文件运算符的用法，观察运行结果。

【代码1】



【运行结果1】

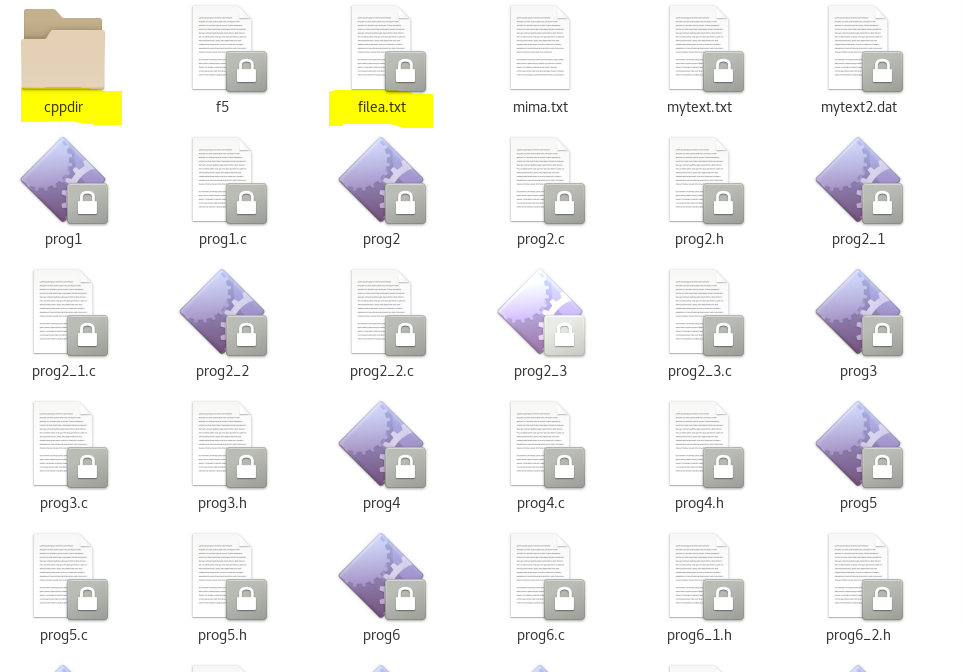


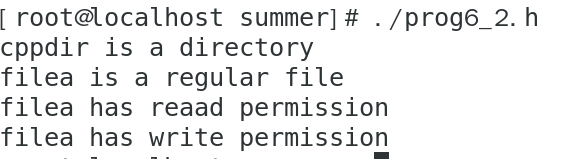
【代码2】



【运行结果2】

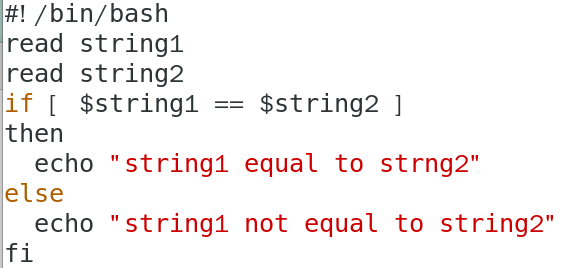
设置文件夹cppdir和文件filea.txt：





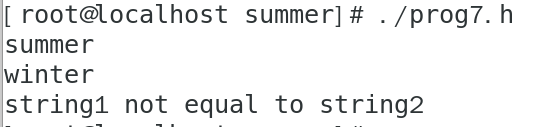
1. 修改例2程序，使在程序运行中能随机输入字符串，然后进行字符串比较。

【代码】

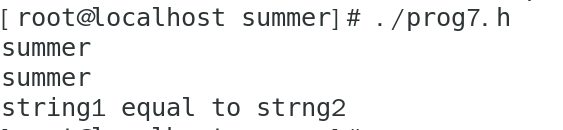


【运行结果】

1. 输入两个不相同的字符串：

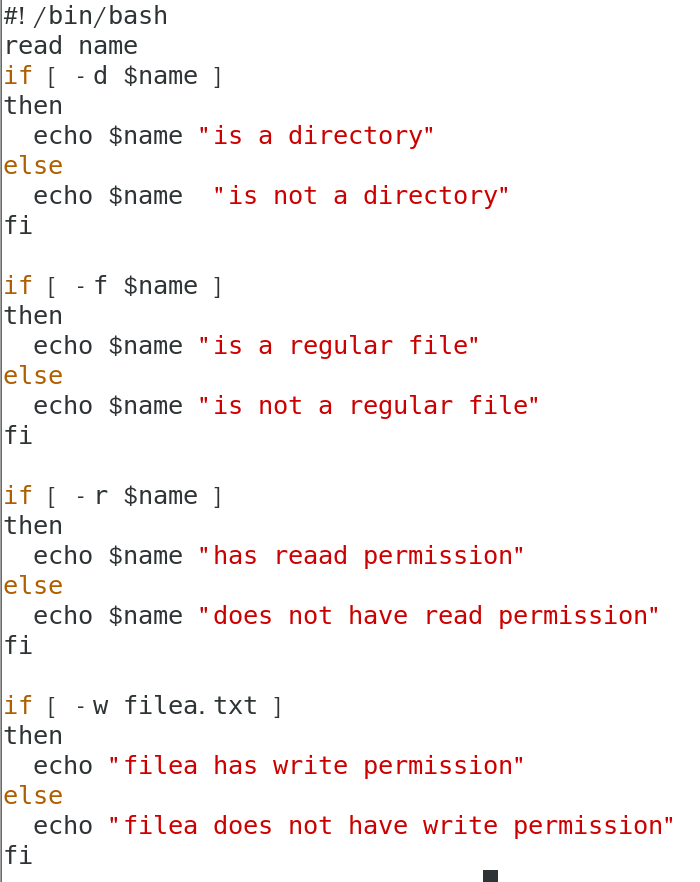


1. 输入两个相同的字符串：



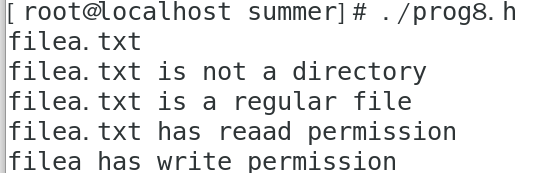
1. 修改例3程序，使在程序运行中能随机输入文件名，然后进行文件属性判断。

【代码】



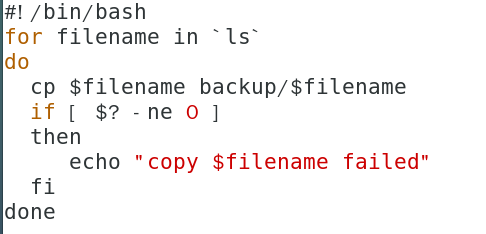
【运行结果】

输入文件filea.txt：

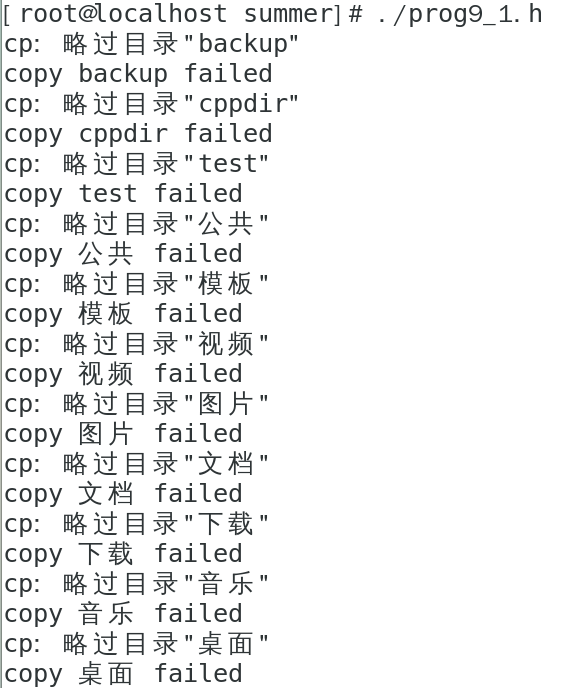


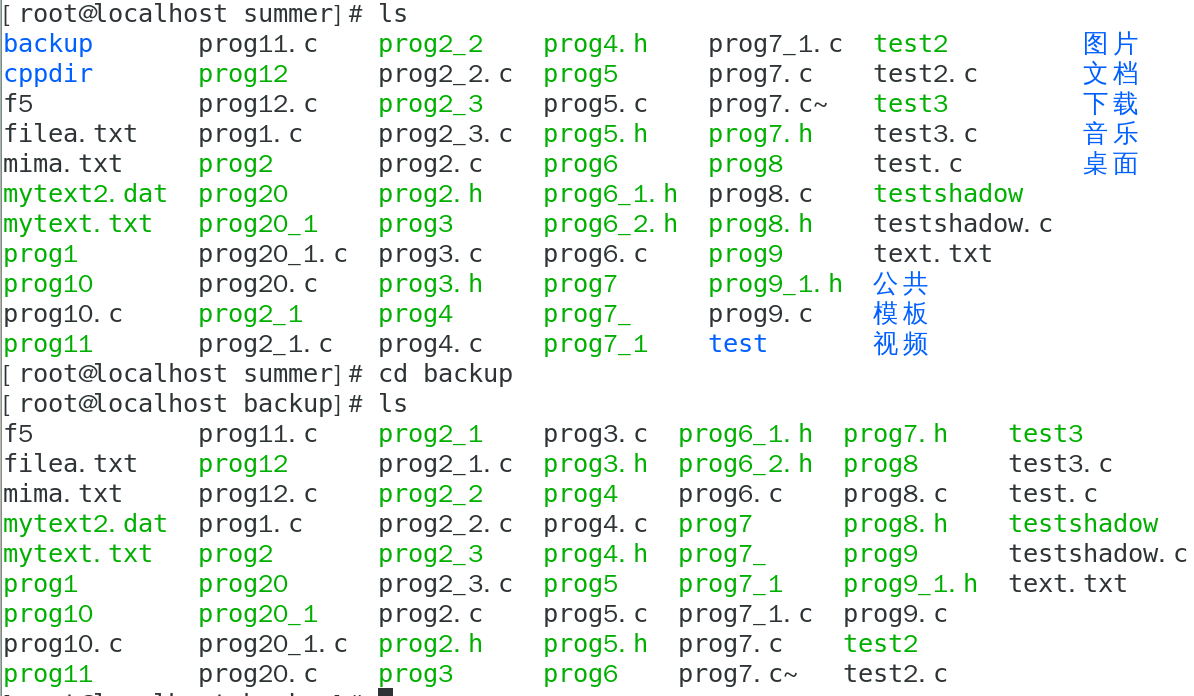
1. 用vi 编写《实验指导》“第四部分 Shell 程序设计”中的例4、例5、例6、例7，掌握控制语句的用法，观察运行结果。

【代码1】

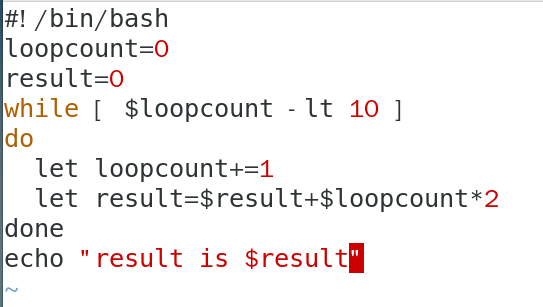


【运行结果1】

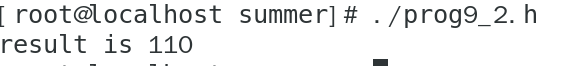




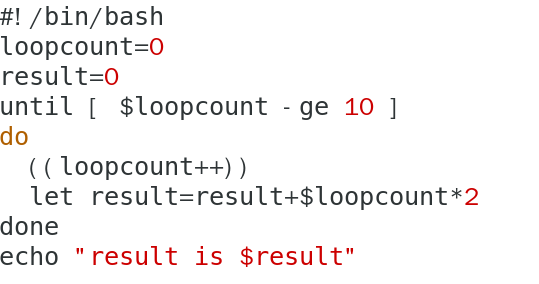
【代码2】



【运行结果2】



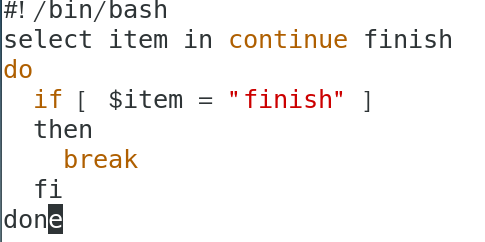
【代码3】



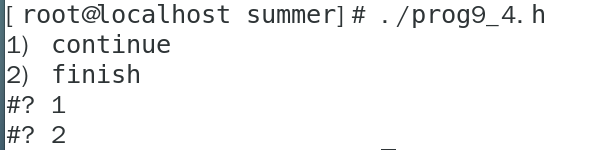
【运行结果3】



【代码4】

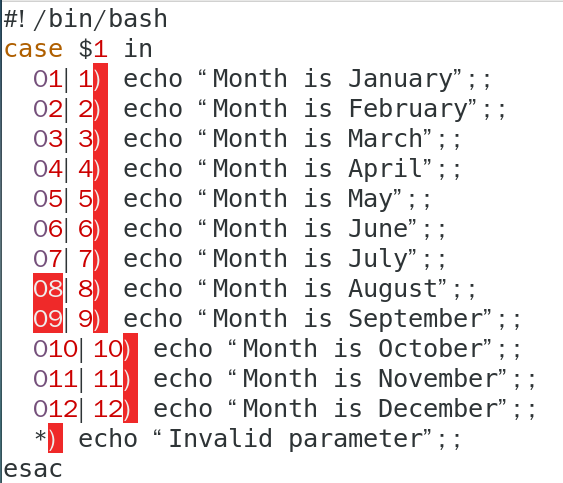


【运行结果4】

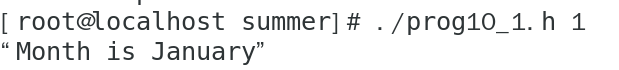


1. 用vi 编写《实验指导》“第四部分 Shell 程序设计”中的例8及例9掌握条件语句的用法，函数的用法，观察运行结果。

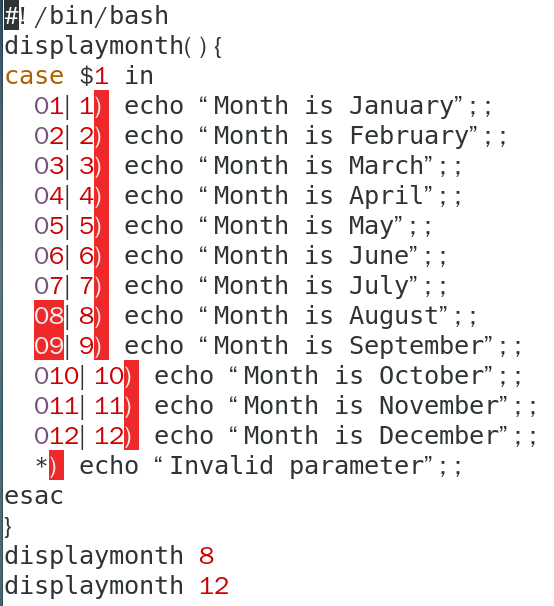
【代码1】



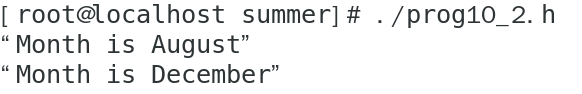
【运行结果1】



【代码2】

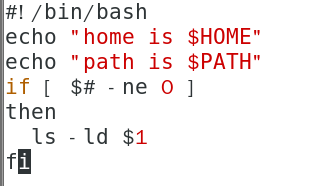


【运行结果2】

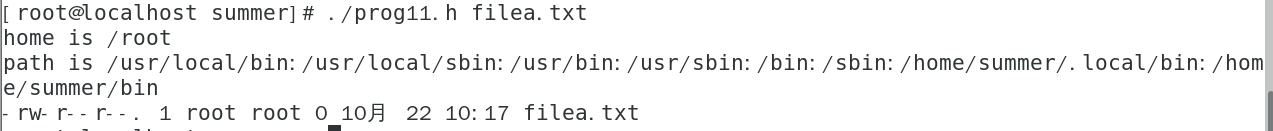


1. 编程，在屏幕上显示用户主目录名（HOME）、命令搜索路径（PATH），并显示由位置参数指定的文件的类型和操作权限。

【代码】



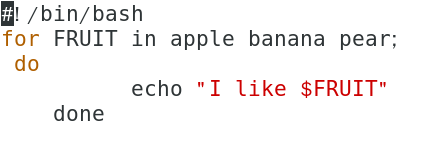
【运行结果】



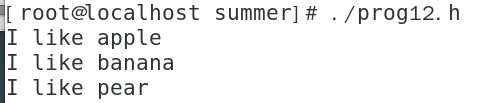
思考：到此为止你对Shell有所认识了吧？怎么样？自己再编两个程序：

1. 做个批处理程序，体会一下批处理概念。

【代码】

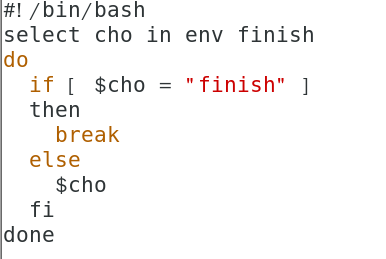


【运行结果】

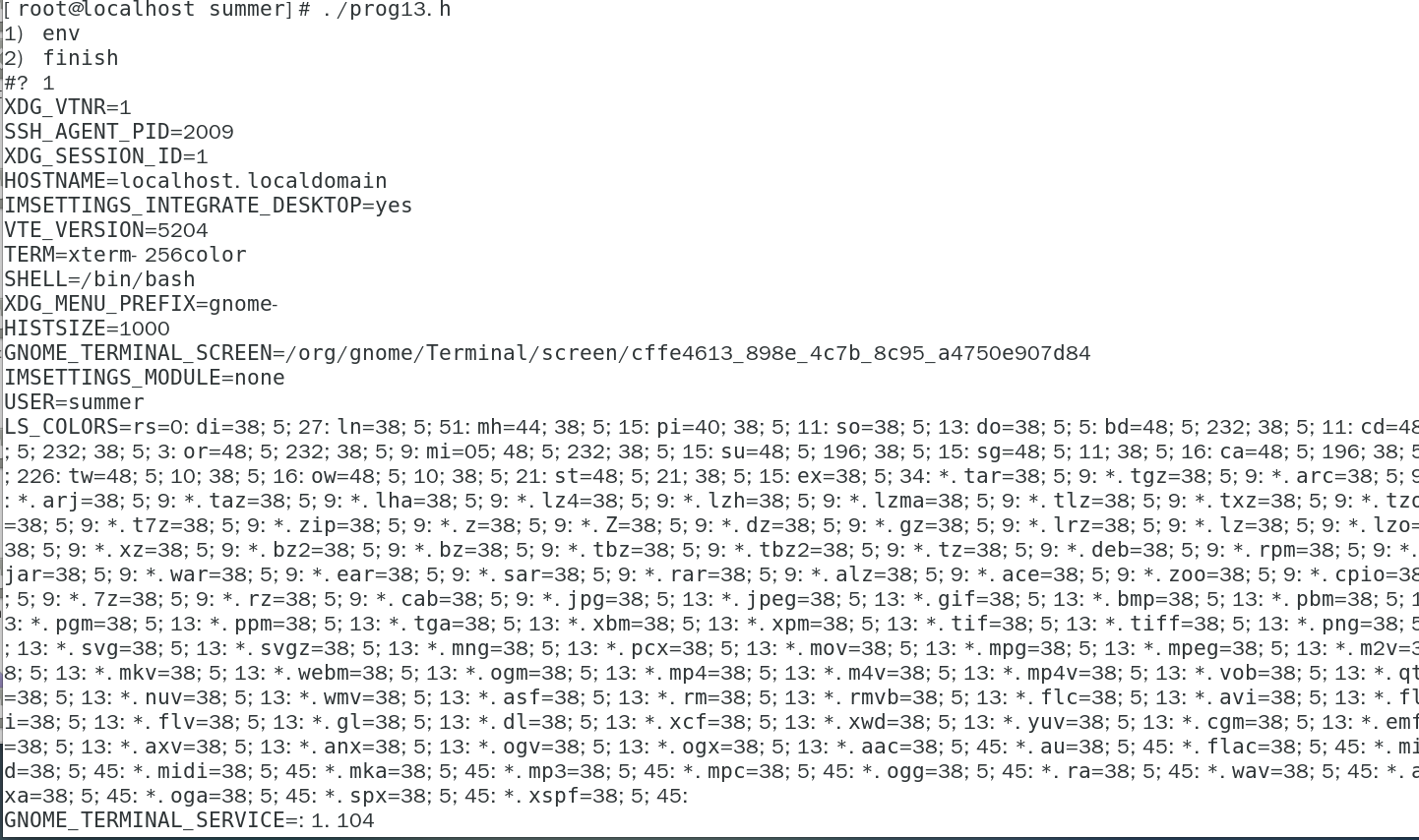


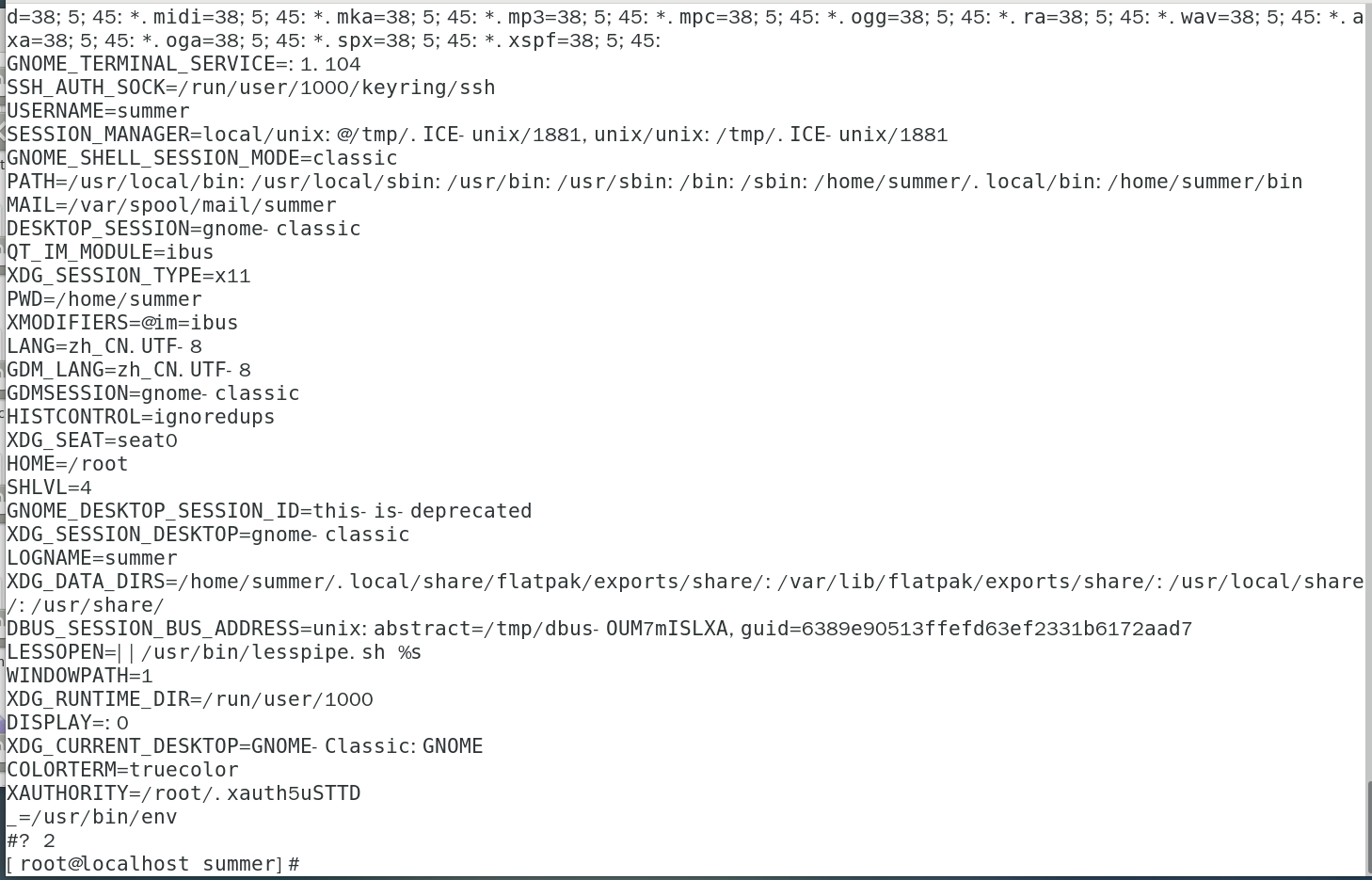
1. 做个菜单，显示系统环境参数。将此程序设置为人人可用。

【代码】



【运行结果】





五. 讨论

1. Linux的Shell有什么特点？

**（1）shell概述**

Shell是用C语言编写的程序，它是用户使用 Linux 的桥梁。Shell既是一种命令语言，又是一种程序设计语言。简单来说Shell 是指一种应用程序，这个应用程序提供了一个界面，用户通过这个界面访问操作系统内核的服务。也可以说，linux中的shell就是linux内核的一个外层保护工具，负责完成用户与内核之间的交互。

**（2）shell的优缺点**

优点方面：

1）shell的语法和结构比较简单，易于掌握，学习和使用比较方便；

2）shell是解释型语言，运行之前不需要编译；

3）程序开发的效率非常高，依赖于功能强大的命令可以迅速地完成开发任务。

4）把已有命令进行适当组合构成新的命令；  
5）提供了文件名扩展字符（通配符，如\* 、 ?、 [ ]），使得用单一的字符串可以匹配多个文件名，省去键入一长串文件名的麻烦；  
6）可以直接使用Shell的内置命令,而不需创建新的进程，如Shell中提供的cd、echo、exit、pwd、kill等命令。为防止因某些Shell不支持这类命令而出现麻烦，许多命令都提供了对应的二进制代码，从而也可以在新进程中运行；

缺点方面：

1）弱类型语言，对变量定义的查检不是很严格；

2）效率上存在劣势，开发产能优于运行的性能；

3）I/O性能不高；

4）一些细节上的不足：向awk传入shell定义的变量时，表达式比较复杂，且对shell的变量是只能读取而不能修改。

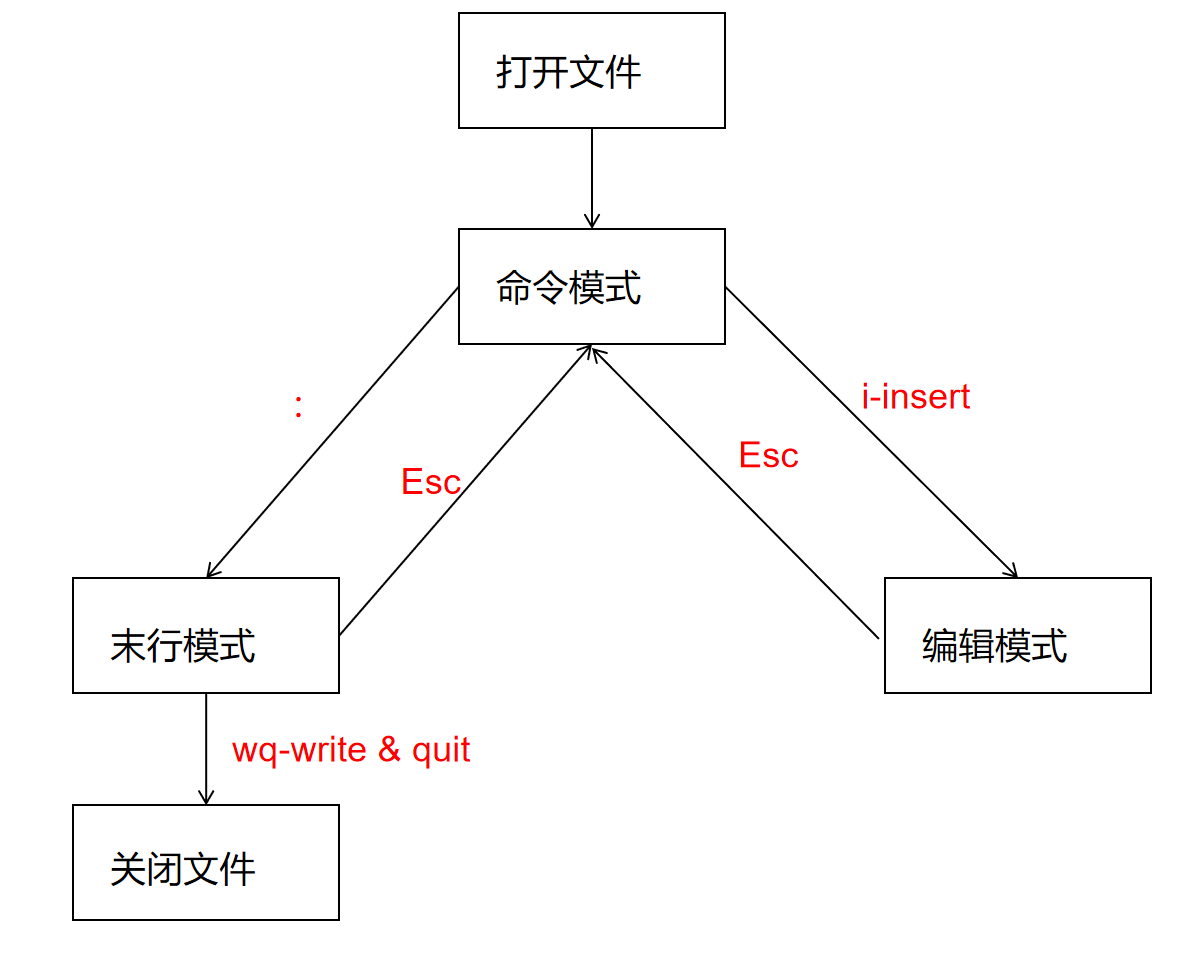
2. 怎样进行Shell编程？如何运行？有什么条件？

shell程序就是一个包含若干行shell或者linux命令的文件，像编写高级语言的程序一样，编写一个shell程序需要一个文本编辑器，如vi和vim等。

在文本编辑环境下，依据shell的语法规则，输入一些shell/linux命令行，形成一个完整的程序文件。

用chmod将文件的权限设置为可执行模式，如若文件名为shdemo.h,则命令如下：$chmod 755 shdemo.h（文件主可读、写、执行），同组人和其他人可读和执行）在提示符后执行Shell程序：$shdemo.h（直接键入程序文件名执行）或$ sh shdemo.h（执行Shell程序）或$.shdemo.h（没有设置权限时可用点号引导）。

1. vi编辑程序有几种工作方式？查找有关的详细资料，熟练掌握屏幕编辑方式、转移命令方式以及末行命令的操作。学习搜索、替换字符、字和行，行的复制、移动，以及在vi中执行Shell命令的方式。



①命令模式：

1. 在该模式中，可以输入命令来执行许多种功能；
2. 打开文件首先进入命令模式，它是使用vim编辑器的入口

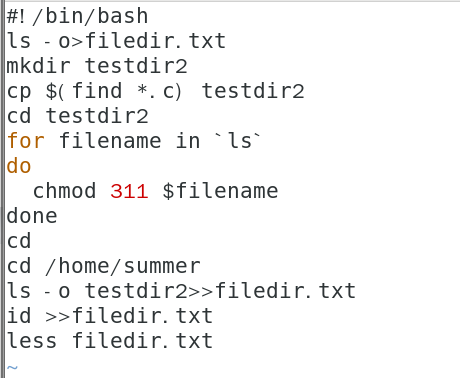
②末行模式：

1. 将文件保存或退出vi，也可以设置编辑环境，如寻找字符串、列出行号等；
2. 末行模式是vim编辑器对的出口，要退出vim，必须要在末行模式下。

③编辑模式：可以对文本进行编辑操作。

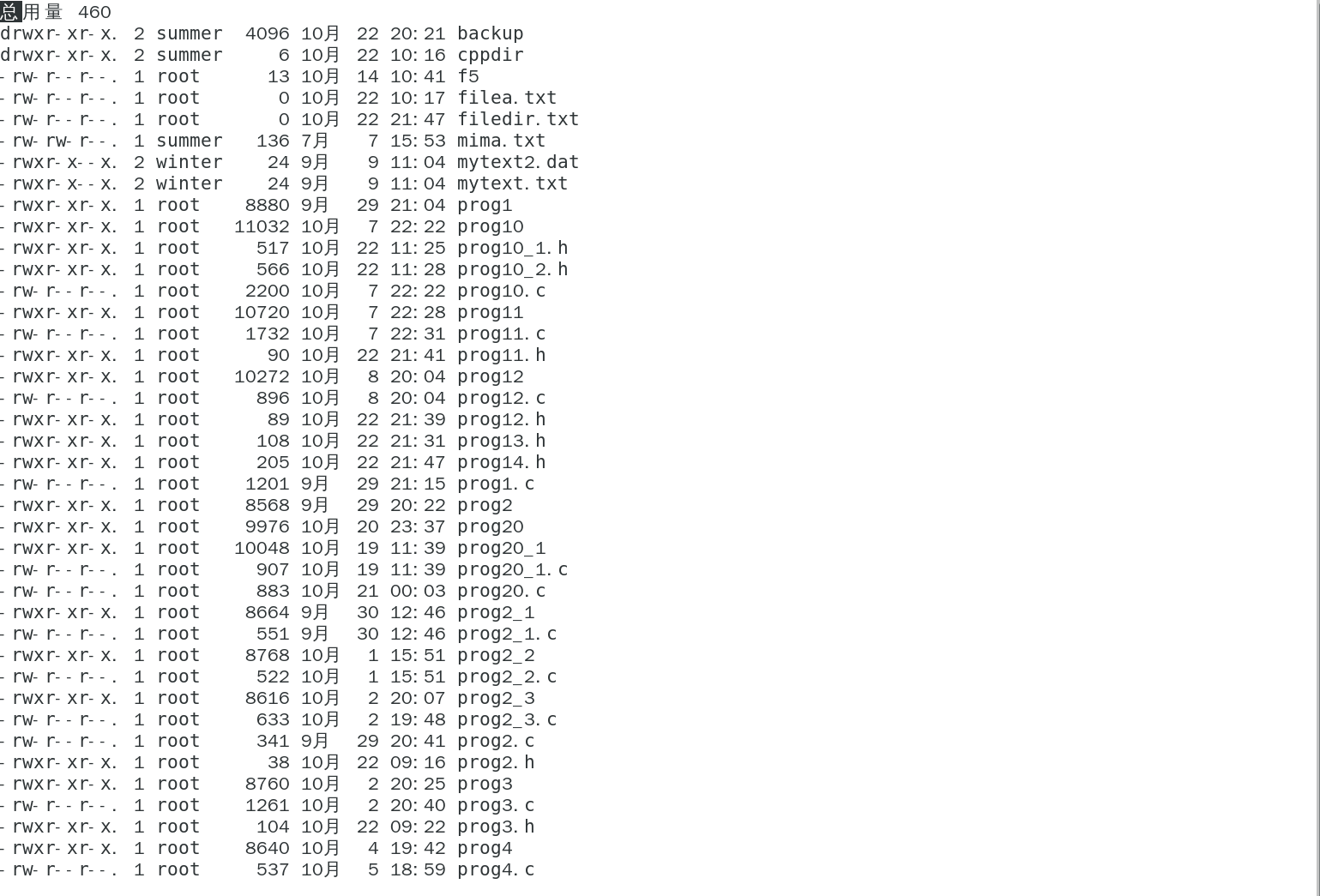
4. 编写一个具有以下功能的Shell程序。

【代码】

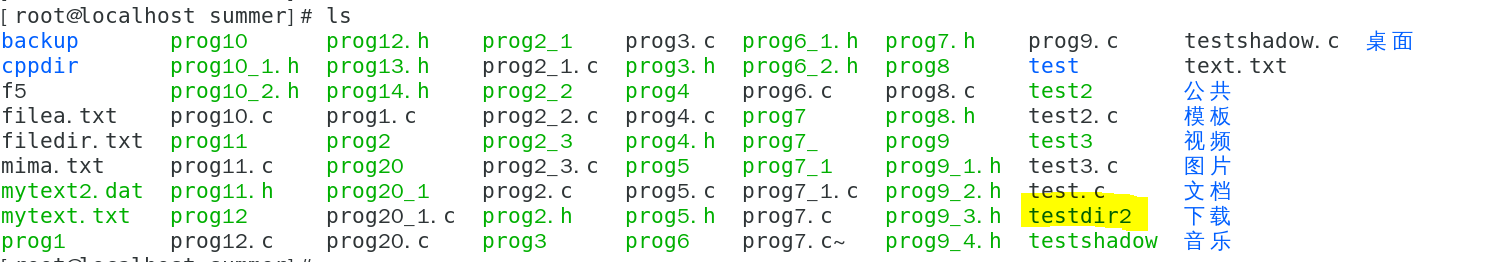


【运行结果】

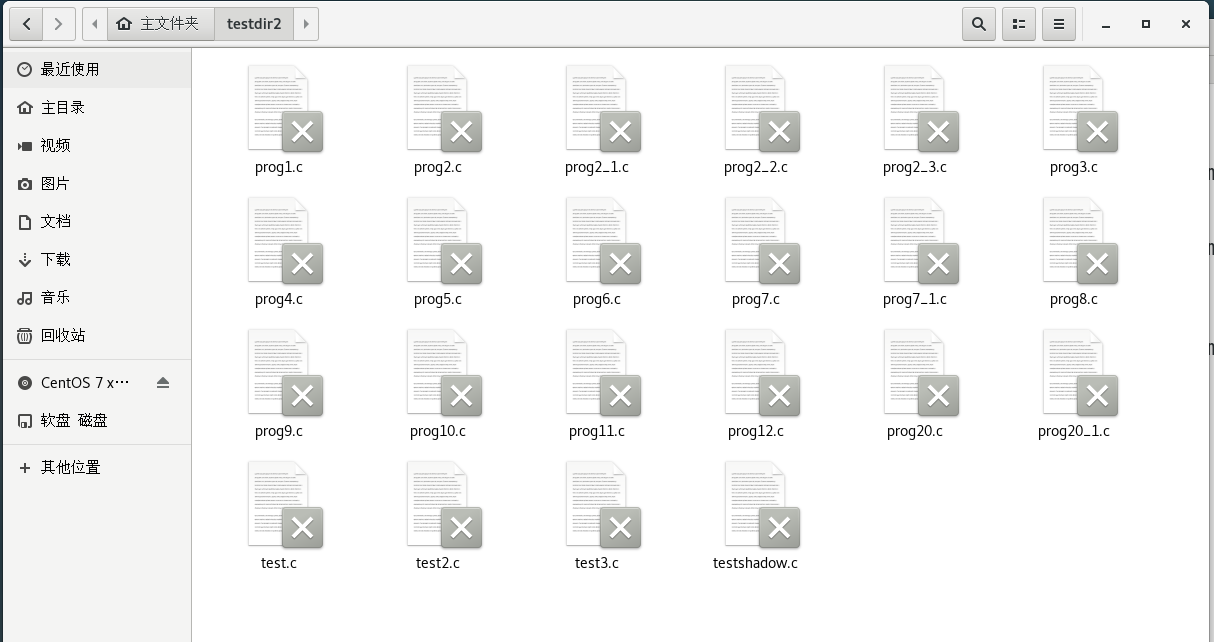
（1）把当前目录下的文件目录信息输出到文件 filedir.txt 中；



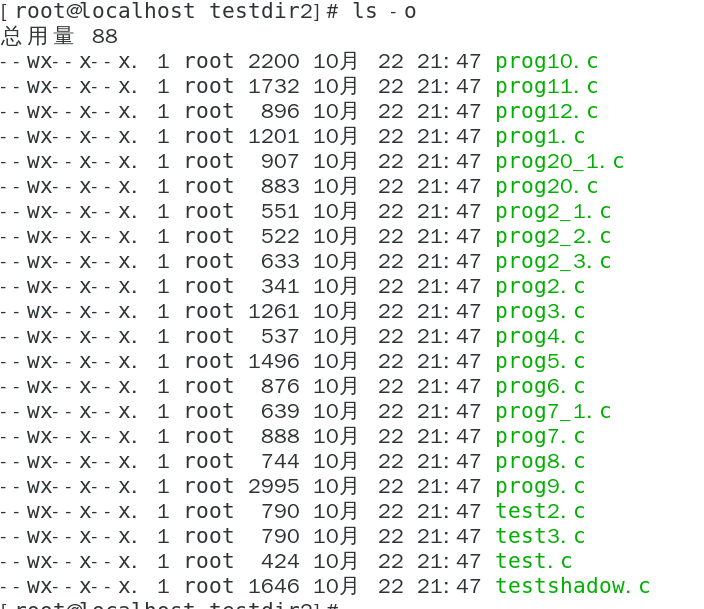
（2）在当前目录下建立一个子目录，目录名为 testdir2 ；



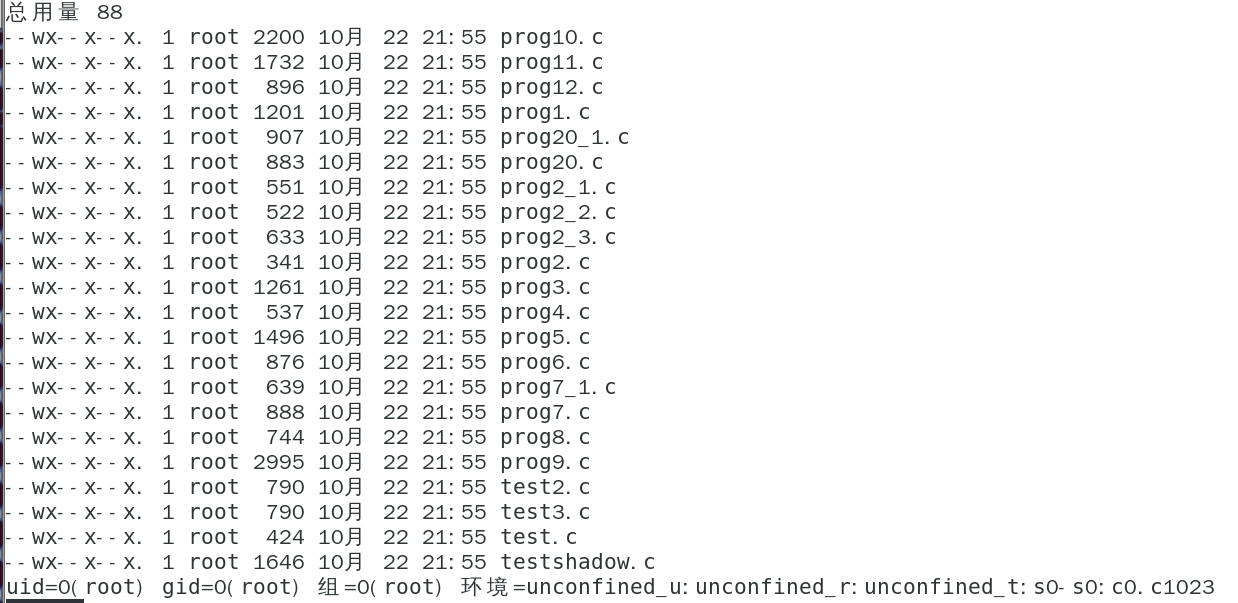
（3）把当前目录下的所有扩展名为 c 的文件以原文件名复制到子目录testdir2中；



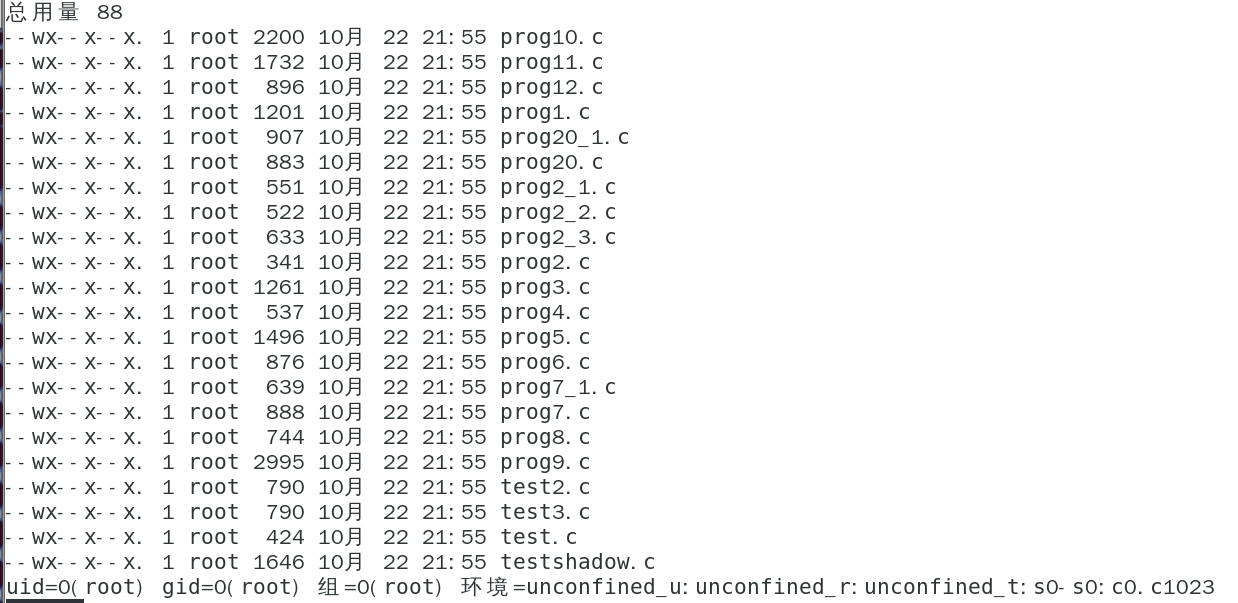
（4）把子目录中的所有文件的存取权限改为不可读。（提示：用 for 循环控制语句实现，循环的控制列表用 ’ls’ 产生。）



（5）在把子目录 testdir2 中所有文件的目录信息追加到文件 filedir.txt 中；



（6）把你的用户信息追加到文件 filedir.txt 中；



（7）分屏显示文件 filedir.txt

