**《操作系统》系列课程之初级**

**《Linux操作系统使用》**

**实验指导书**

# 试验1 Linux的基本命令及vi初步

# 实验1.1 命令基础

* 实验所属系列：操作系统初级试验
* 实验对象：本科
* 相关课程及专业：计算机操作系统，软件专业
* 实验类型： 课程设计
* 实验时数（学分）： 1学时

## 实验目的

掌握Linux的基本安装，熟悉Linux命令的格式，熟悉一些常用的Linux命令，熟悉如何使用帮助。

## 实验内容与要求

### **要求**

登录进入Linux系统，而后更改自己的用户口令

执行几个常用的Linux命令

使用man命令，来查找特定命令的帮助信息

### **内容**

一、登录，更改密码

1. 安装VMware虚拟机，安装Linux操作系统。在安装过程中设置root用户的密码为111111。

从图形界面进入控制台界面的组合[Fn+Crtl+Alt]

从控制台界面进入图形界面[Alt+F7]

1. 登录后将当前的密码改为你自己希望的密码。用心记住新密码。用笔记录密码的话，容易引起安全问题。(passwd命令)
2. 退出系统后，再重新进入，以确定用户自己的密码已经被修改。

[Ctrl+d] / logout / exit

1. 系统关机：poweroff / shutdown
2. 重启系统：reboot

二、基本命令

1. 显示系统的时间(date)
2. 显示文件/etc/passwd中一共有多少行(wc -l /etc/passwd)
3. 显示2002年的全年的日历
4. 显示1752年9月的月历，请注意这个月的月历是否有些特别。
5. 显示1999年1月与99年1月的月历。1999年与99年是同一年吗？
6. 有两个命令是用来显示当前系统中的用户情况的。用这两个命令显示现在有哪些人已经登录到系统中，并查看他们是何时登录进入系统的。
7. 只显示用户自己的登录名。
8. 使用　clear　命令，清屏。

三、man 命令

使用man man命令，来查找man命令的帮助信息。仔细阅读系统所给出的帮助信息，注意man命令能提供哪几个方面的帮助？使用man cal命令，在系统所提供的帮助中，找出cal命令的正确的使用格式。尝试使用help、info等帮助命令。

# 实验1.2 文件与目录

实验所属系列：操作系统初级试验

实验对象：本科

相关课程及专业：计算机操作系统，软件专业

实验类型： 课程设计

实验时数（学分）： 1学时

## 实验目的

熟悉与目录和文件操作相关的命令

## 实验内容与要求

### **要求**

显示当前目录的目录名

改变当前目录

使用ls命令的不同命令选项，来查看文件与目录的属性

创建和删除目录

创建0长度的文件

拷贝、移动、重命名、链接及删除文件

查看文件的内容

### **简介**

在本练习中，将使用学过的命令对目录及普通文件进行操作。

### **内容**

### 一、检查你现在所处的环境

1. 登录进入系统：使用pwd命令，确认你现在正处在自己的主目录中。这个目录就是在你登录时，用户缺省进入的目录。

思考：设置默认主目录的环境变量是PATH

### 二、查看目录

1. 把你的当前目录改变为根目录。
2. 确认你的当前目录是根目录，并且使用简单列表命令和长列表命令来列出本目录中的文件。调用使用-a或-R命令选项的ls命令。每一个命令选项各有什么作用？回到你的用户主目录，并列出所有的文件（包括隐藏文件）
3. 在你的用户主目录中创建一个名为mydir的新目录。然后调用长列表命令来查看/home/xxx/mydir目录和/home/xxx目录（只查看目录文件的信息）。
4. 把目录/home/xxx/mydir变为自己的当前目录。使用touch命令在mydir目录创建名为myfile1,myfile2的两个文件。
5. 使用长列表命令来查看mydir目录中的文件的信息。myfile1与myfile2的文件大小是多少？再次使用长列表命令，同时列出索引结点(inode)的信息。这两个文件的索引结点号是什么？
6. 返回到你的用户主目录，调用ls –R命令来查看你的用户主目录中的目录树结构。
7. 使用rmdir命令删除mydir目录，这个命令工作了吗？请注意rmdir命令不能删除一个非空的目录。为了删除这个目录，你需要调用命令rm –r。

### 三、对文件进行操作

1. 查看/etc/inittab和/etc/passwd文件的内容。使用命令cat、view、less、more来分别查看每一个命令的输出方式（提示：/etc/inittab文件包括了系统登录的相关信息，/etc/passwd文件包括了被授权使用该系统的用户的列表）。比较cat、view、less、more这几个命令的异同。

截图

1. 拷贝文件/usr/bin/cat到你的用户主目录中。该目录没有cat文件，用find命令查找
2. 拷贝文件/usr/bin/cal到你的用户主目录中。
3. 列出你的用户主目录中的文件，你会发现上面两个文件已经被拷贝过来了。在你的用户主目录中创建一个名为xxxscript的目录

### 四、对目录进行操作

1. 将上面两个文件（cat, cal）移动到xxxscript目录中，并且重新命名为mycat和mycal；
2. 将xxxscript目录变为你的当前目录。列出当前目录中的各个文件名，确认上述两个文件已经被拷贝过来并已更名。
3. 使用mycat命令来显示你的用户主目录中的文件.bash\_profile中的内容。返回到你的用户主目录中。
4. 在你的用户主目录中创建另一个子目录goodstuff，拷贝文件/etc/profile到这个新建的目录中，并将拷贝过来的文件更名为newprofile。
5. 使用cat命令来查看文件的内容。是否容易阅读？再试一下more命令。
6. 文件名newprofile真的是太长了，不方便输入。请把文件名改为np。在改名完成后，请查看goodstuff目录中的文件名，以确保你已经完成了这项工作。
7. 现在是检查你的劳动成果的时候了。从你的用户主目录向下，层次地显示你的子目录和文件。

Ls -R

1. 确认你正处在自己的用户主目录中。试着删除目录goodstuff。你成功了吗？为什么？

### 五、删除目录

1. 进入到goodstuff目录。将该目录中的所有文件（包括隐藏文件）列出来。删除这些文件。再对该目录中的所有文件列表。注意. 和 ..这两个项目仍然在那里。如果一个目录中只有. 和..两个条目，那么这个目录就是空的。删除这个目录。
2. 在xxxscript目录中，只使用mkdir命令一次，就创建出子目录sports和sports目录中下的三个子目录tennis、basketball、baseball。创建完成后，检查你所做的工作的成果。
3. 使用cat命令将文件/etc/inittab文件拷贝到tennis目录中，并在basketball目录中使用cat > 命令，创建两个新的文件。查看目录的内容，确保这些文件操作都已经成功完成。
4. 使用rm命令来删除sports目录及其中的所有文件与目录。

# 实验1.3 修改文件和目录权限

实验所属系列：操作系统初级试验

实验对象：本科

相关课程及专业：计算机操作系统，软件专业

实验类型： 课程设计

实验时数（学分）： 1学时

## 实验目的

给学生一个操作文件与目录权限的机会。作完这些练习之后，使学生能够对Linux中文件所有者的概念和权限的概念有基本的理解。

## 实验内容与要求

### **要求**

使用长列表命令来查看文件与目录的信息

能够解释文件与目录权限中各位的含义

能够对普通文件与目录的权限进行操作

### **简介**

在本练习中，学生将使用Linux命令来操纵文件与目录的权限。理解文件权限和文件的所有者的含义，并在必要的时候改变权限。

**提示**

在进行各种操作时，首先必须弄清楚自己所在的目录。如果你对自己目前的位置失去了感觉，那么各种命令可能都会失效。最有效的方法就是，经常使用pwd命令，来查看你所在的当前目录。

### **内容**

1. 登录进入系统。进入xxxscripts目录。使用长列表命令来查看目录中的各个文件。请注意，你在前面练习中拷贝到该目录中的几个文件的所有者和权限

记录mycat文件的权限：

记录mycal文件的权限：

1. 现在对/usr/bin目录中的cat文件和cal文件使用长列表命令，并把它们与xxxscripts目录中的文件相比较。有什么不同？（对于拷贝文件，你是新的拥有者）
2. 改变xxxscripts目录中mycal和mycat文件的更改时间。请调用相关的命令来查看改动是否已经成功了？touch命令的另外一个功能是什么？
3. 设法做到这一点：在你的用户主目录中能够以home\_mycal的文件名来调用xxxscripts/mycal文件。比较这两个文件。它们有什么不同吗？文件的链接数是多少？
4. 回到你的用户主目录，执行home\_mycal文件。执行的结果是什么？现在，改变文件home\_mycal的权限，使拥有者只具有读的权限。试着再运行home\_mycal文件，它运行了吗？为什么？删除home\_mycal。当你这样做的时候，也删除了myscripts/mycal文件吗？为什么？

./ home\_mycal

chmod u-wx home\_mycal

./ home\_mycal

rm home\_mycal

1. 进入到myscripts目录中。使用chmod命令的符号方式，将mycat文件中的其它用户的读权限删除，查看新的权限。

chmod go-r mycat

1. 使用chmod命令的八进制方式，将mycat文件的拥有者的权限设置为只有读权限，将其它所有用户的权限设置为没有任何权限。查看新的权限。
2. 对.bash\_profile文件执行mycat命令。它工作了吗?发生了什么？
3. 回到你的用户主目录，查看你的当前目录。
4. 改变myscripts目录的权限，使你自己对它仅具有读的权限。使用长列表命令来查看，你是否正确设置了权限。
5. 列出该目录中的文件名。再试着用一下长列表命令，它工作了吗？为什么？

试着运行mycat文件，它运行了吗？为什么？

试着删除mycal文件，你成功了吗？为什么？

1. 将xxxscripts目录的权限恢复到原来的形式rwxr-xr-x，然后删除mycal文件。
2. 尝试用数值方式修改一个文件的权限。
3. 尝试使用whereis、grep、which命令

# 实验1.4 vi编辑器

实验所属系列：操作系统初级试验

实验对象：本科

相关课程及专业：计算机操作系统，软件专业

实验类型： 课程设计

实验时数（学分）： 1学时

## 实验目的

给学生一个使用vi创建与编辑文件的机会。vi是Linux中最常用的编辑器。对vi编辑器的全面而清晰的理解，对于后续内容的学习至关重要。

## 实验内容与要求

### **要求**

创建一个文件

保存退出一个文件及不保存退出一个文件

在文本中使用不同的键进行光标的移动

在一个文件中加入、删除与修改文本

设定选项以自定义编辑环境

调用命令行编辑功能

### **简介**

vi编辑器是由加州大学伯克利分校计算机科学系开发的软件演化而来的。vi的意思是“vee-eye short for visual”。接下来的练习将使你熟悉vi的一些基本的功能和特色。

### **内容**

### 一、创建一个文件

确认你现在正处在你的用户主目录中。在你的用户主目录中创建一个文件vitest。当你用vi打开一个文件时，被缺省地置为命令模式中。按i（insert）键，将转换为输入（文本）模式。你也可以按a(append)键，进行转换。应当根据是在光标处输入还是在光标后输入决定是按i键还是按a键。你进入输入模式时，没有任何信息提示你。从输入模式转换回“命令模式”时，请按Esc键。注意，当你连续两次按Esc键时，终端可能会发出beep声来提示你，它的意思是你已经处于命令模式了，不用再按了。再按下i键，你又回到了输入模式。

请严格地将下面所提供的内容一行行地输入到文件中去。在输入完成后，再将字母a,b,….z输入到文件中去，每一个字母占一行。这样的话，我们就有两屏的内容可供练习使用。

This is a training session about the usage of the vi editor. We need some more lines to learn the most common commands of the editor. We are now in the entry mode and we will switch right after this to the command mode.

返回到命令模式。保存并退出文件。注意，你一按：号，：立即出现在输入区域的最后一行。一旦缓冲区为空，并且文件已经关闭，你将会看到这个文件有多少行及多少个字符。

### 二、移动光标

使用vi打开vitest文件。注意文件的最后一行，显示了文件的名字和字符数。练习使用光标键和h,j,k,l键来移动光标。向下移一行，向上移一行，向左移动两个字符，向右移动两个字符。你也许不想只是将光标每次只移动一个字符或一行。现在练习每次将光标移动一屏或者移动几行。请在命令模式下，将光标放在第一行，然后做以下练习：

将光标向下移动一屏

将光标向上移动一屏

将屏幕向上滚动1/2窗口

将光标移到文件的最后一行

将光标移到文件的第一行

将光标移到文件的第四行

将光标移到行的末尾

将光标移到行的开始

1. 将光标移动到文件的开始处。查找词entry。你的光标将被放到字符e处。转换到输入模式，并且加入一个单词text,和一个空格。
2. 将光标移到同行的单词mode后的空格后，插入一个逗号。注意，你当前处于输入模式。
3. 进入命令模式。把光标置于以“some more lines”开始的那一行中的任何位置上，然后插入一个空行，以形成两段。
4. 插入一个空行时，你被自动地转换到输入模式。现在，请转换到命令模式，保存你所作的修改，但是不要退出编辑器。
5. 现在是命令模式。删除字符c,e,g，但仍然将空行保留。也就是说，只删除字符，不删除整行。而后回到这些行的位置处，将这些空白行删除。你可以使用两种删除的方式。
6. 现在，请你用字母z来替换字母h。
7. 你现在决定不保存这些更改，退出编辑过程。（不保留上次保存以来所做的所有更改）
8. 再次编辑vitest文件。首先，一次一行地将第一段中的每一行拷贝到文件的末尾。当这项工作完成时，将第二段整个地拷贝到文件的末尾。
9. 你发现你刚才加到文件末尾的内容是重复的内容，使用一个命令一次将它们全部删除。
10. 现在，你想把当前的日期与时间嵌入到文件的第一行。请在vi编辑内部做这项工作。

### 三、使用set命令，定制编辑环境

1. 使用set命令，可以为你当前编辑过程设定环境。返回到文件的顶端，在确保你已处于命令模式的前提下，执行下列的命令：

设定右边界单词自动折行为15个空格

当处于输入模式时，能显示出当前的模式。

显示行号

1. 检查前面所设定的选项的作用。
2. 保存并退出编辑。

### 四、命令行编辑

1. 你已经熟悉了vi的模式与命令，所以现在可以练习命令行编辑了。使用set –o vi命令来启动命令行编辑功能。
2. 现在你可以调出以前执行过的命令，编辑它们，并且重新执行它们。现在我们先执行几条命令。ls /usr cat /etc/filesystems echo hello
3. 假设你现在想编辑一条你刚才执行过的命令。先按Esc键，以进入vi命令模式。多次按k键，将倒序显示刚才已经执行过的几条命令，再按几下j键，将正序显示这几条命令。这实际是从命令缓冲中重新调回它们。所有已经执行过的命令，全部保存在你的用户主目录中的.sh\_history文件中。
4. 重新取回ls命令，使用l键将光标移到/字符处（如果你使用光标键来移动光标，将会删除光标所经过的内容，所以你只能使用k键向右移动，使用l键向左移动）。而后，使用i键来插入文本，将该命令改成一个长列表命令。最后，执行它。
5. 重新调回cat命令，此次，显示文件/etc/passwd文件的内容
6. 重新调回cat命令，将光标置于行尾（使用$）,在该命令的结尾处加一个管道符，使其输出成为wc命令的输入，显示出该文件的行数。

# 试验2 shell初步

# 实验2.1 shell基本操作

* 实验所属系列：操作系统初级实验
* 实验对象：本科
* 相关课程及专业：计算机操作系统，软件专业
* 实验类型： 课程设计
* 实验时数（学分）： 2学时

## 实验目的

使学生熟悉shell的基本操作，并使学生有机会使用Linux中最常用的数据处理命令。

## 实验内容与要求

### **要求**

使用通配符来进行文件名的查找

重定向标准输入、标准输出、标准错误输出

使用管道操作，将一个进程的输出作为另一个进程的输入

使用命令行续行符和命令组合

在指定的目录中，对满足条件的文件名进行递归查找

在文本文件中，查找符合指定模式的文本行

将文本文件中的各行，按顺序排列

显示一个文本文件的前几行或者后几行

使用历史（history）命令

### **简介**

shell是用户与内核之间的接口，shell在（理解）解释了用户输入的命令之后，就调用内核中的程序来执行它。所以，了解shell是如何解释和执行用户输入的命令，是理解Linux用户接口的基础。你将使用教材中讲到的命令进行练习。

另外，这个练习让学生有机会使用数据处理命令。如果想知道更多的关于这些命令的格式与选项的信息，使用man命令。在这个练习中，有些很长的命令将出现多次。使用命令行编辑功能会有助于你成功地完成本练习。

### **内容**

### 一、通配符

1. 使用cd命令，回到用户的主目录中（宿主目录）
2. 调用简单的ls命令，来列出用户的主目录中所有非隐藏文件。现在，使用ls命令与通配符（\*）的组合来重新列出文件。这两个命令的输出有什么不同吗？为什么？
3. 进入/usr/bin目录，仅列出以字母a开头的所有文件名。
4. 列出所有只有两个字符的文件名。
5. 列出所有以a,b,c,d四个字符中的一个开头的文件名。
6. 列出所有不是以c-t开头的文件名。这将是一个很长的输出。你可以使用管道操作，将该命令的输出作为pg或more命令的输入(pg命令在Linux中没有)。你是否看到了：你不想列出的文件名（如大写的R，S开头的文件名）也列出来了？如果是这样，为什么？
7. 回到你的用户主目录

cd[Enter]

### 二、重定向

1. 使用cat命令与重定向，创建一个名为junk的文件。先随便输入几行内容，然后在最后一个新行的开始处，按<ctrl+d>以返回到shell的$提示符。
2. 使用cat命令在你刚刚创建的文件的末尾新加入几行文本。
3. 打印当前登录用户，并把显示情况重定向到一个文件中

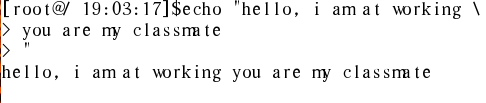
### 三、管道、过滤、tee及&

1. 使用ls命令，列出当前目录中的文件。记下文件的数量为：\_\_
2. 使用ls命令，列出当前目录中的文件，并把输出重定向到文件temp中。
3. 使用相应的命令来计算temp中的单词的数量。这个数量与你已经记下的数量相等吗？如果不相等，为什么？先查看temp中的内容，而后删除它。
4. 现在使用一个管道操作来计算当前目录中的文件数目。这次所得到的结果与你已经记下的数量相等吗？
5. 调出你刚刚执行的命令，在管道中加入tee命令，将列表的输出存到文件junk2中。你得到了显示在屏幕上的文件数目了吗？查看junk2文件的内容，其中包含了你想要的数字吗？
6. 逆序列出当前目录中的文件。把结果输出到文件junk3中。计算这个逆序列表的单词数，并将结果输出到文件junk3的末尾。注意，重定向操作，有追加内容到文件末尾的功能。在这个非常特殊的例子中，你可能得到的是你不想要的结果。因为junk3文件在本命令中使用了两次，所以重定向不能是覆盖操作。请仔细想想，如何实现它。
7. 给出一个后台运行程序的例子，例如(sleep 20; echo hi)&，再用ps命令打印当前的活跃进程，并给出当前活跃进程的父子关系分析。

### 四、命令组合以及续行符

1. 在同一命令行中，调用如下命令：显示日期；显示已经登录到系统中的所有用户名；显示当前目录的名称；显示当前目录中的所有文件名。这些命令，相互之间有联系吗？
2. 本练习的主要目的是，在命令太长，一个命名行无法容纳下的时候，使用命令行接续符，使命令能够完整地执行。第二个目的，是应用你已经获得的知识，来创建一个超长的命令。

你可以在你认为合适的地方，将一行结束。当你将命令输入完成时，马上运行它。而后查看你所创建的文件的内容，以评估命令是否被正确地执行了。这是一个使用了管道和重定向功能的超长的命令。



长列表当前目录中的所有文件，包括隐藏文件将上个命令的输出捕获（tee）到文件reverse.listing文件中，并且将这个输出送到下一命令中，以计算单词的数量。计算出单词的数量，并且该数量值同时按顺序存放到（tee）文件file1,file2,file3,file4中。最后，计算出文件file4中的行数，并将该行数的值重定向到文件file5中。

### 五、find命令

1. 找到并显示出/tmp目录中所有的文件名。
2. 在你的用户主目录中查找所有以字母s开头的文件名。而后，对它们自动执行ls –l命令。
3. 重复上一步的查找。不同的是，对查找的文件名执行ls –l命令时，采用与用户进行交互的方式。
4. 在/usr目录中，查找所有用户uucp拥有的文件。而后，计算出这些文件的个数。对于该目录中的一些子目录，你是没有权力读的，执行命令时将会产生一些错误信息。请将所有的错误信息重定向到一个名为errfile的文件中。
5. 显示errfile文件的内容。查看在执行上面的命令时，是否有错误发生。
6. 向下递归地查找所有子目录

确认你正处在自己的用户主目录中

创建一个名为level1的目录

在子目录level1创建一个名为letter1的零长度文件

进入到level1目录中

创建一个名为level2的目录

在子目录level2中创建一个名为letter2的零长度文件

进入到你的用户主目录中

在你的用户主目录中，列出所有以字母l开头的文件与目录。记录所显示的文件名与目录名。

在你的用户主目录中，启动一个查找命令。查找出所有以字母l开头的文件与目录。记录找到的文件与目录。

### 六、grep命令

1. 在文件/etc/passwd中，找出所有以team开头的用户名所在的文本行。
2. 在文件/etc/passwd中，找出所有以t开头的文本行
3. 在文件/etc/passwd中，找出所有包含数字0-9的文本行
4. 重复上面的查找，但这次只显示所找到的文本行的数量
5. 联合使用ps和grep命令，显示出系统中所有由其它用户启动的进程

### 七、sort命令

1. 以字母顺序显示/etc/passwd文件的内容。然后，以逆序显示其内容。

### 八、head命令与tail命令

1. 显示/etc/passwd文件的前10行的内容
2. 显示/etc/passwd文件的前5行的内容
3. 显示/etc/passwd文件的后10行的内容
4. tail命令也可以用于去除一些命令输出的头部信息。首先用一条命令列出在系统中运行的所有进程。注意该命令输出的标题信息。然后，再使用另一条命令列出在系统中运行的所有进程（不包括头部信息）。
5. 可以通过修改/etc/passwd文件的内容，重新设置用户的默认登录目录，如果要把root用户的默认登录目录改为/home/root，应该如何操作。

### 九、history命令

1. 列出最近输入的20条指令
2. 使用命令记录号执行一条历史命令
3. 重复执行一次命令
4. 执行最后一次以ls开头的指令
5. 在history命令文件中查找mkdir命令

# 实验2.2 系统配置文件及环境变量

实验所属系列：操作系统初级试验

实验对象：本科

相关课程及专业：计算机操作系统，软件专业

实验类型： 课程设计

实验时数（学分）： 1学时

## 实验目的

熟悉Linux中常用的系统（用户）配置文件以及几个重要的环境变量

## 实验内容与要求

### **要求**

系统配置文件

用户配置文件

常用环境变量的设置

### **简介**

在本练习中，将使学生掌握修改配置文件的方法。

### **内容**

### 一、配置文件相关操作

1. 查看系统配置文件：使用vi命令分别打开/etc/bashrc，/etc/profile，~/.bashrc，~/.bashrc，~/.bash\_logout文件进行查看，请分析/etc/bashrc配置文件。
2. 修改配置文件，使任何用户在登录时打印其登录的时间；
3. 修改用户配置文件，使该用户在登录时打印如下语句：hello，i am xxx, now i am working。这里的xxx是该登录用户的用户名。

思考：系统配置文件与用户配置文件的继承关系，例如在系统配置文件中设置其PATH变量，在用户的配置文件中又重新设置该变量，那么在该用户登录后，PATH变量是什么。

### 二、shell的环境变量

1. 查看当前系统设置的环境变量，请分别说明以下变量的含义：

HOME、IFS、MAIL、PATH、PS1、PS2、SHELL、TERM、HISTSIZE、HISTFILE、HOSTNAME、USER、PWD

1. 修改当前的提示符格式为[登录用户@当前工作目录 登录时间]
2. 使当前cd[enter]操作的返回目录为/home

# 实验2.3 shell编程

实验所属系列：操作系统初级实验

实验对象：本科

相关课程及专业：计算机操作系统，软件专业

实验类型： 课程设计

实验时数（学分）： 1学时

## 实验目的

使用户熟悉Unix的基本Shell程序设计方法，包括命令行参数检测、变量设置、文本建立和输入、程序运行控制和输入输出重定向。

## 实验内容与要求

编写一个shell程序，将用户从键盘输入的文本附加到一个指定的文件中。如果该文件不存在，则新建立该文件；如果该文件已经存在，则把键盘输入的文本附加到该文件后面。输入内容及行数不限定，以空行表示输入结束。运行结束时显示该文件内容。

一、建立shell程序exam1.sh

1. 用屏幕编辑程序vi建立shell程序exam1.sh。程序头应有几行以#号开始的简单的程序功能说明行（特别说明使用哪种shell），其它主要的语句前或后都应有功能注释。
2. 通过检测变量$#的值判断命令行参数个数是否正确，如果正确则程序继续运行下去；否则提示用户该命令的运行格式：

Usage: exam1.sh filename

并退出运行。

1. 设置变量INPUT，并赋予任意一个非空的字符串。
2. 用echo语句在屏幕上提示用户输入任意一行文本（空行表示整个输入结束）。
3. 用while建立一个无限循环，循环退出条件为变量INPUT为空，循环体为用read语句读入一行到变量INPUT中，用echo命令将该变量的内容用附加方式的重定向（>>）输出到用户在命令行上指定的文件名中。
4. 用cat命令将该文件的内容显示在屏幕上。

请把源代码贴在作业中

二、用chmod命令设定exam1.sh的执行权限。

1. 用ls命令查看刚建立好的exam1.sh的读写保护模式。
2. 用chmod命令修改exam1.sh的读写保护模式，使用户自己拥有执行权限。

三、运行程序。

1. 在当前目录下直接用程序名exam1.sh运行该程序.如果出现类似如下提示：

exam1.sh: not found

是什么原因？如何调整？

用./执行exam1（创建一个子进程bash，在该子进程中执行脚本，需要改权限）

用bash执行exam1；（创建一个子进程bash，在该子进程中执行脚本，不需要改权限）

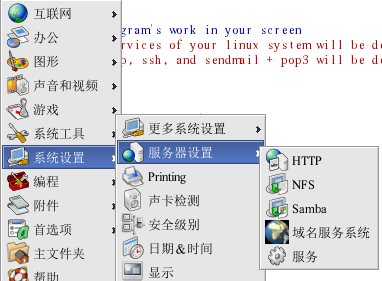
用(.)执行exam1；（直接在本进程中执行脚本，不需要改权限）

试分析这几种执行方法的区别。

1. 试试不带参数或参数个数大于等于2个来运行exam1程序，看程序能否正常判断，并显示提示信息后退出。
2. 用一个新文件名newfile作为参数正常运行exam1，任意输入若干行，并以一个空行结束输入。程序能否显示前面的所有输入行后结束运行。
3. 用ls命令查看当前目录下是否有新建立的newfile文件，并用cat命令查看该文件的内容。
4. 再次用newfile作为参数运行exam1，运行结束后查看newfile文件中本次键盘输入的内容是附加还是覆盖上次运行时输入的内容。

四、编写一个程序，能够监测主机上是否提供以下服务：www, ftp, ssh, sendmail + pop3。

1. 通过service命令或者在[服务]中设置开启vsftpd、sshd、httpd，查看当前主机提供服务的情况



# 试验3 图书馆管理系统

* 实验所属系列：操作系统初级实验
* 实验对象：本科
* 相关课程及专业：计算机操作系统，软件专业
* 实验类型： 课程设计
* 实验时数（学分）：4学时

## 实验目的

掌握shell编程的一般方法，能运用常用的shell命令编写一个简单的shell程序，并能在Linux系统所提供的/bin/sh或/bin/bash下正确的运行。

## 实验内容与要求

一、功能要求:

编写一个菜单驱动的Linux图书馆管理程序(LLIB)，该程序能够对书库中的图书执行增加一条记录、删除一条记录、显示一条记录、更新一条记录状态和打印报表等功能。图书库为一个文本文件library, 其中每一行为一个记录，包括六个字段，描述一种书的六种信息（假设每一种书只能有一本），分别是：

* 书名
* 作者
* 种类，假设有三种有效类型：

系统书：简写为system

参考书：简写为reference

教科书：简写为textbook

* 状态：表示书被借出还是在书库中。图书状态由程序决定，当增加一本书时或书被还回时，该状态自动设为in；当书被借出时，状态改为out；
* 借阅者姓名：当图书在书库中（即状态字段为in）时，该字段为空；当书被借出（即状态字段为out）时，该字段为借阅者姓名；
* 日期：当图书在书库中（即状态字段为in）时，该字段为空；当书被借出（即状态字段为out）时，该字段为借出日期。

由于在书名和作者字段的字符串中可能包含空格，所以在每条记录中各字段之间的间隔符为冒号（：）。

程序结构要求：

|  |
| --- |
| **LLIB** |

|  |
| --- |
| **DELETE** |

|  |
| --- |
| **DISPLAY** |

|  |
| --- |
| **UPDATE** |

|  |
| --- |
| **ADD** |

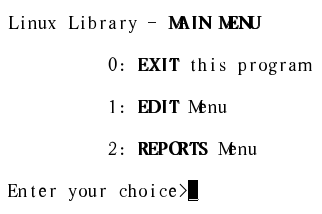
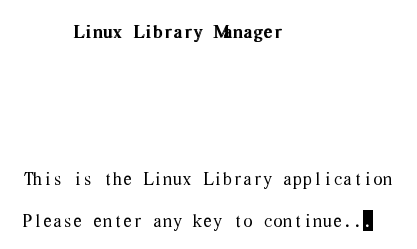
|  |
| --- |
| **EDIT** |

|  |
| --- |
| **REPORTS** |

|  |
| --- |
| **ERROR** |

|  |
| --- |
| **REPORT\_NO** |

LLIB: 主菜单程序，运行时先清屏，并在屏幕中间用加粗（bold）模式显示主菜单：



光标停留在下划线上，等待用户选择。用户选择0时程序退出；选择1、2时进入相应的功能程序，选择错误时在选择行的下一行提示：

Wrong Input. Try again.

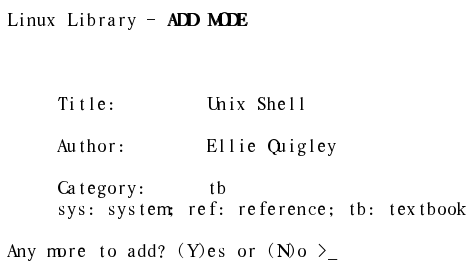
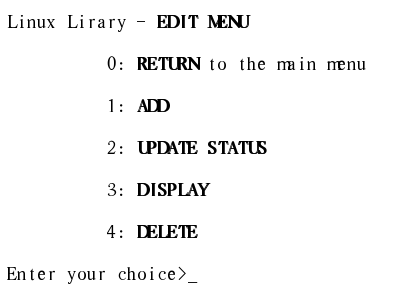
Press any key to continue...>\_

光标停留在提示行的行尾，等待用户按键，用户按任意键后, 清除错误提示行并重新显示选择行。

用户选择1时，进入EDIT菜单。

* ADD：增加一条记录(一本书)。

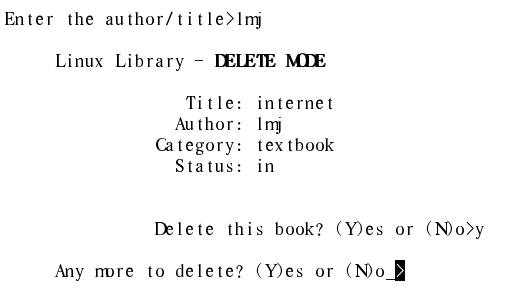
程序运行时，先清屏，并依次提示用户输入书名(title)、作者(author)和类型(category)，再将用户输入的内容和该记录的状态（status）、借阅者（bname）和日期（date）等字段的初始值一起组成一行，用附加方式的重定向（>>）存入到图书库文件（library）中（字段之间用冒号隔开）。



一条记录输入完成后，询问用户是否继续输入新的记录，如果是，则程序循环下去；否则退出ADD程序，返回到EDIT菜单程序中。

* DELETE：删除一条记录

程序运行时，先清屏，并提示用户输入要删除的书名。程序在图书库文件library中查找与书名相匹配的记录，如果发现了该记录，则用合适的格式显示该记录，并显示确认提示，例如：

****

用户确认后，删除该记录。提示，删除记录可用下列命令：

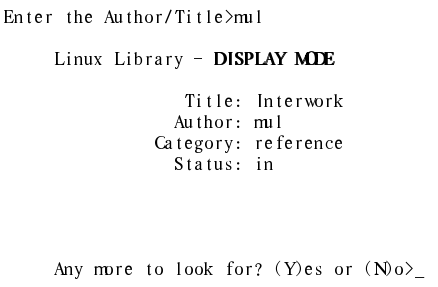
grep -v “$title:$author:$category:$status:$bname:$date” ULIB\_FILE > temp

mv temp ULIB\_FILE

用户回答N或删除完成后，提示用户是否继续进行删除，如是，则循环下去；否则，返回EDIT菜单程序。

* DISPLAY：把ULIB\_FILE库文件中指定记录的内容显示出来。

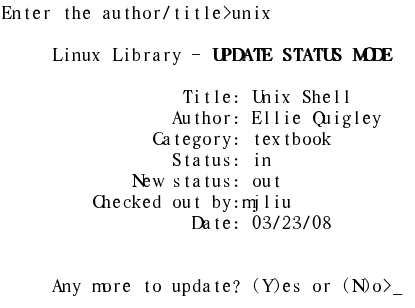
程序运行时，先清屏，并提示用户输入书名(title)，如果找到指定记录，则按适当格式显示该书的信息；否则显示该书找不到，并提示是否继续查找，如用户回答肯定，则重新开始下一轮查找；回答否定则返回EDIT菜单程序。

****

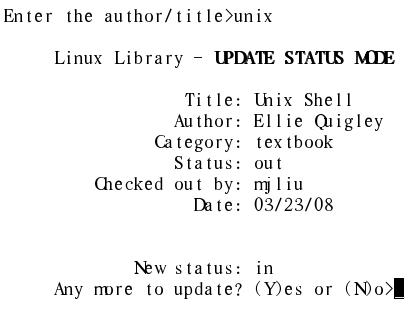
* UPDATE：当借出一本书或还回一本书时，执行该程序更改库文件ULIB\_FILE中指定记录的状态。

程序运行时，先清屏，并提示用户输入书名(title)，如果找到指定记录，则按适当格式显示该书的信息；

如果原来该记录的状态为in，则显示该书的信息后，显示新的状态为out，再提示用户输入借阅者名字，显示系统时间。从库文件中删除该记录，再把状态字段已改为out，以及已填入了借阅者和日期字段后的新记录附加到库文件中去，例如：

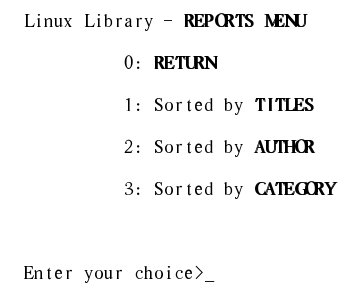
****

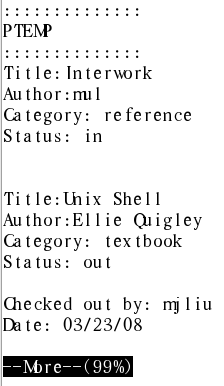
如果原来该记录的状态为out，则显示该书的信息后，显示新的状态为in，从库文件中删除该记录，再把状态字段已改为in，以及已清除借阅者和日期字段后的新记录附加到库文件中去，例如：

****

一条记录的状态更新完成后，提示用户是否更新其它记录的状态，如果是，则继续循环；否则退出该程序返回到EDIT菜单程序。

* REPORTS：该程序运行时先清屏，再把库文件ULIB\_FILE中的各条记录，以书名排序，按适当的格式，逐屏（用more命令）显示在屏幕上。显示完成或中断显示后，返回EDIT菜单程序，例如：

****

****

二、 程序编写要求：

1. 程序开始几行应为以#开始的程序说明， 程序中的重要语句也应有说明；SHELL的类型应选用 /bin/bash；

2．程序语句应选用本课程中介绍过的命令和语句

# 试验4 Linux程序设计基础

* 实验所属系列：操作系统初级实验
* 实验对象：本科
* 相关课程及专业：计算机操作系统，软件专业
* 实验类型： 课程设计
* 实验时数（学分）：4学时

注：本实验有三个方向可以选择完成：

* + 依照本实验内容完成。
  + 制作RPM包。
  + 使用QT/GTK完成一个图形界面的hello，world程序。

## 实验目的

掌握Linux下C/C++编译器gcc，能够使用gcc(g++)开发一个简单的应用程序；掌握Linux调试工具gdb，并能进行简单的程序调试；学会使用GNU make，并能够通过automake和autoconf工具自动生成makefile文件。

## 实验内容与要求

1. 使用gcc开发一个简单的应用程序

该程序功能如下：由用户输入一个整数，程序计算该整数的阶乘，并显示计算结果。请使用gcc编译生成一个名为factroial的可执行程序。

给出编译命令解图及原代码

1. 使用gdb调试一个有错误的c程序

源代码如下：

#include <stdio.h>

static char buff[256];

static char \*string;

int main()

{

printf("please input a string:\n");

gets(string);

printf("your string is: %s\n", string);

}

调试步骤：

* + - 1. 对源代码进行编译，发现出错；
      2. 运行 gdb bugging 命令，装入 bugging 可执行文件；
      3. 执行装入的 bugging 命令；
      4. 使用 where 命令查看程序出错的地方；
      5. 利用 list 命令查看调用 gets 函数附近的代码；
      6. 唯一能够导致 gets 函数出错的因素就是变量 string。用 print 命令查看 string 的值；
      7. 在 gdb 中，我们可以直接修改变量的值，只要将 string 取一个合法的指针值就可以了，为此，我们在第11行处设置断点；
      8. 程序重新运行到第 11 行处停止，这时，我们可以用 set variable 命令修改string的取值；
      9. 然后继续运行，将看到正确的程序运行结果。

注意截图说明

最后给出正确的源代码

1. 根据图示依赖关系编写一个makefile文件

依赖关系图如下：



1. 通过automake和autoconf完成一个makefile文件的自动生成
   * 1. 检查是否已经安装以下软件：GNU automake、GNU autoconf、GNU m4、perl、GNU libtool；
        + 1. 、创建hello.c源代码

#include<stdio.h>

int main(int argc,char \*\*argv)

{

printf("hello,world!\n");

return 0;

}

3、使用autoscan命令，产生configure.scan文件和autoscan.log 文件，进行如下修改：

AC\_PREREQ(2.57)（版本号不用改）

AC\_INIT(hello.c)（格式可能不同）

AM\_INIT\_AUTOMAKE(hello,1.0)（也可以不带参数）

AC\_PROG\_CC

AC\_OUTPUT(Makefile)

1. 、把configure.scan修改为configure.ac作为aclocal命令的输入文件，运行aclocal命令，生成autom4te.cache子目录和 aclocal.m4； 执行autoconf命令，生成configure可执行文件；
2. 、执行autoheader命令，生成config.h.in文件；
3. 、创建Makefile.am文件，内如如下：

AUTOMAKE\_OPTIONS = foreign

bin\_PROGRAMS = hello

hello\_SOURCES = hello.c

1. 、执行automake --add-missing命令，生成重要的Makefile.in文件； （会出警告，直接朝下进行）
2. 、执行./configure
3. 、make
4. ./hello

注意理解第7、8、9、10步。

vim hello.c

vim Makefile.am

autoscan

vim configure.scan

mv configure.scan configure.ac

aclocal

autoheader

autoconf

automake --add-missing

./configure

make