




山东大学Linux应用课程实验记录

原创

[shadowingszy](#)  于 2018-11-04 17:14:52 发布  5851  收藏 64

文章标签: [Linux](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/u011748319/article/details/83717674>

版权



[Linux 专栏收录该内容](#)

2 篇文章 1 订阅

订阅专栏

找到这篇博文的人, 一定被Linux实验弄得很爆炸吧哈哈哈。

这里是我Linux实验的记录, 供大家学习和参考。如有错误, 还请指正。

实验一

一. 基本命令

显示系统当前时间.

```
date
```

显示2003年的日历.

```
cal 2003
```

分别显示1999年1月份和99年1月份的日历. 1999年的和99年的是不是一个?

不是

```
cal 1 1999
```

```
cal 99
```

列出系统中所有登陆用户.

```
who
```

显示你的登陆名.

```
whoami
```

显示你的用户信息.

```
whoami
```

清屏.

```
clear
```

显示字符串'Out to lunch'.

```
echo 'Out to lunch'
```

使用wall命令进行系统广播:Good Evening everyone!

```
wall 'Good Evening everyone!'
```

使用一个命令屏蔽所有消息.这个命令对所有用户都起作用吗?

不是

```
mesg n
```

二. Man 手册

使用man手册察看ls命令如何使用.

```
man ls
```

试着解释一下ls命令的man手册有几部分组成?每一部分是用来说明什么的?

指令名, 指令格式, 指令参数, 作者, BUG反馈, 版权说明

在man手册中试着移动光标:

上下键: 上下移动一行

空格键: 到下一页

退出man手册.

q键退出

实验二

一. 对目录的操作

检查你当前所在的目录，该目录是？

用户主目录“~”。

改变当前目录到根目录(/)。

```
cd /
```

验证是否在根目录，分别用简单列表和长列表的方式列出目录中的文件。

```
ls
```

```
ls -l
```

列出当前目录中的所有文件，列出当前目录及其子目录中的所有文件。

```
ls -R
```

返回到你的主目录，列出该目录的内容包括隐藏文件。

```
cd ~
```

```
ls -a
```

在你的主目录里创建一个名为mydir的目录。然后，发出命令分别显示你的主目录和~/mydir目录的长列表。

```
cd ~
```

```
mkdir mydir
```

分别用绝对路径和相对路径两种方式转到mydir目录。创建两个名为myfile1和myfile2的文件。

```
cd mydir
```

```
cd ~/mydir
```

```
touch myfile1
```

```
touch myfile2
```

用长列表形式显示mydir目录的内容。长列表列出的各栏分别显示了什么信息？文件myfile1和myfile2的大小是多少？

读写权限 文件数 拥有者 所属的group 文件大小 建档日期 文件名

均为0

```
ls -l
```

返回到你的主目录然后使用 ls -R 命令来显示你的目录树。解释显示出来的信息。

```
cd ~
```

```
ls -R
```

试着用rmdir删除mydir目录。能行吗？为什么？

不行，因为mydir非空

进入mydir目录删除该目录中的2个文件。然后返回到你的主目录并且删除掉mydir目录。

```
cd mydir
```

```
rm myfile1
```

```
rm myfile2
```

```
cd ..
```

```
rm mydir
```

用一条rm命令可否实现12题中的要求？如何实现？解释rm命令的作用。

```
rm -rf mydir
```

二. 对文件的操作

分别用cat,more和less命令查看 /etc/passwd 文件的内容。/etc/passwd 文件包含了能使用该系统的有所有授权的用户的一个列表.这三个命令显示的方式有什么不同?

```
cat /etc/passwd
```

```
more /etc/passwd
```

```
less /etc/passwd
```

cat 显示文件全部的内容。

more/less 分页显示文件的内容。

复制/etc/passwd 文件到你的主目录,并将其重命名为usersfile.用一条命令实现。如果用两条命令实现呢?

一条命令: `cp /etc/passwd ~/usersfile`

两条命令: `cp /etc/passwd ~` `mv passwd usersfile2`

用wc命令统计文件userfile的字符数、行数和单词数.解释显示出的各栏对应的内容。如果只统计行数呢?

```
wc ~/usersfile
```

分别创建userfile的一个硬链接文件userfile-hard和一个字符链接文件userfile-soft。

```
ln userfile userfile-hard
```

```
ln -s userfile userfile-soft
```

用whereis搜索命令passwd所在的路径。用locate命令搜索文件名中带有aa字段的文件。 whereis和locate命令的区别是什么?

whereis是精确搜索, locate可以模糊搜索

```
whereis passwd
```

```
locate aa
```

三. 对文件和目录的递归操作

创建一个sub1目录,在sub1中再创建一个 sub2 目录.用一条命令来实现.

```
mkdir -p ~/sub1/sub2
```

转到 sub2 目录,创建一个文件 myfile.

```
cd ~/sub1/sub2
```

```
touch myfile
```

返回到你的主目录.复制整个 sub1目录树到tree1. 分别递归显示目录sub1和tree1里的所有文件.

```
mkdir ~/tree1
```

```
cp -Rf ~/sub1 ~/tree1
```

```
cd ~/tree1
```

```
ls -R
```

```
cd ~/sub1
```

```
ls -R
```

现在你有了2个目录树, sub1和tree1. 把目录树tree1 移到sub1子目录中.

```
mv ~/tree1 ~/sub1
```

列出你的主目录的内容. 递归显示sub1目录中的所有文件和目录.

```
cd ~
```

```
ls
```

```
cd sub1
```

```
ls -R
```

四. 创建用户帐号

为了完整地演示权限, 我们需要创建一些附加的用户, tux1和tux2, 它们都是penguins 组的成员.

切换到tty3(按Ctrl+Alt+F3键),用 root帐号登录。

执行下列一系列命令:

```
groupadd penguins
useradd -m -g penguins -c "Tux the Penguin (1)" tux1
useradd -m -g penguins -c "Tux the Penguin (2)" tux2
passwd tux1
New password: penguin1
Retype new password: penguin1
passwd tux2
New password: penguin2
Retype new password: penguin2
```

3. 切换到tty1(按Ctrl+Alt+F1键), 用tux1帐号登录, 密码是penguin1, 然后再切换到 tty2(按Ctrl+Alt+F2键),用tux2帐号登录, 密码是penguin2.

这里没什么可说的, 照做就好

五. 文件和目录的权限

切换到tty1(按Ctrl+Alt+F1键), 你是以 tux1帐号登录的, 查看你的主目录的权限.

读、写、执行权限

切换到tty2, 你是以 tux2帐号登录的. 试着改变到tux1的主目录, 或者显示tux1主目录的内容. 能行吗?为什么?

不行, 没有权限

切换到tty1. 改变tux1主目录的权限以使其他用户能够有读(read)和执行(execute)的权限.然后再以tux2帐号试着进入tux1的主目录. 现在能行吗?

可以

作为用户tux2, 试着创建和删除tux1的主目录里的文件. 能否成功?

不行

再次切换到tty1. 创建一个bin目录,复制文件/bin/ls 到此目录,并且重命名为my_ls.

设置my_ls的权限为rw-r-----, 然后分别以tux1和tux2登录试着运行它. 能否运行? 为什么?

不行, 没有执行权限

重新设置my_ls的权限为rwxr-xr-x, 然后再次分别以tux1和tux2登录试着运行它. 现在能否运行?

可以, 有执行的权限

试着运行 my_ls, 分别以tux1, tux2, 和你自己的帐号, 权限分别为 rw-----, rw-rw----, rwx-----, rwx-x--- 和 rwx-x-x. 对于tux1, 运行my_ls所需要的最小权限是什么? 对于tux2呢? 对于你自己的帐号呢?

对于tux1, 需要-x-----

对于tux2, 需要----x---

对于自己的帐号, 需要-----x

实验三

一. 使用vi文本编辑器

以某个帐号登入系统，确认是否在主目录里。在主目录里用vi文本编辑器创建一个名为vittest的文件。可随意输入，尽量多输入以便能多分几屏显示。

```
cd ~  
vi vittest
```

返回到命令行模式，保存文件但不退出vi。注意：按下“:”键, 该符号会在输入区域的末行显示。一旦缓冲区是空的，文件被关闭，可以看到一段信息，显示出文件的行数和字符数。

任意修改文件vittest的内容，然后放弃所有修改，强制退出vi。

输入 q!

二. 光标移动键的使用

再次打开vittest文件，注意在屏幕的底部指示出该文件的名称和字符数。

使用，，和键，练习在文件中移动光标。最简单的移动光标的方法是使用上,下,左,右键。

三. 删除和粘贴

将光标移动至第5行alphanumeric处。用yy命令复制这一行内容，并粘贴到下一行。

将光标移至第6行，删除这一行的内容。

四. 字符串搜索与取代

设置一个文件中已经存在的字符串，比如在本文件中设置该字符串为"binary"。试着正向和反向搜索字符串binary。

转入第14行。将该行中的字符串"binary"用字符串"str1"来取代。如果要用"str1"替换文件中出现的所有字符串"binary"呢？

用命令u撤销11中进行的操作。

五. 其它

移至文件的末尾，另起一行输入"the current path of file myfile is: ", 然后写入myfile所在的路径。

在命令模式下用查看当前文本的状态。都显示什么信息？

保存文件退出。

在命令方式下可有几种方法退出vi编辑器, : wq , : ZZ , : q!

实验四

一. shell环境

(一) 命令和文件名补全功能

用date命令查看系统当前时间，在输入da后，按tab键，让shell自动补齐命令的后半部分。

用mkdir命令创建新的目录newdir。首先输入第一个字母m，然后按tab键，由于以m开头的命令太多，shell会提示是否显示全部的可能命令，输入n。

```
cd ~  
mkdir newdir
```

再多输入一个字母k，按tab键，让shell列出以mk开头的所有命令的列表。

在列表中查找mkdir命令，看看还需要多输入几个字母才能确定mkdir这个命令，然后输入需要的字母，再按tab键，让shell补齐剩下的命令。

最后输入要创建的目录名newdir，按回车键执行命令。

用ls命令查看newdir目录下的内容，看newdir是否可以用文件补全功能来输入？

可以进入

(二) 命令别名功能

输入alias命令，显示目前已经设置好的命令的别名。

```
alias
```

设置别名ls为ls -l,以长格形式显示文件列表。

```
alias ls='ls -l'
```

显示别名ls代表的命令，确认设置生效。

```
alias ls
```

使用别名ls显示当前目录中的文件列表。

```
ls
```

在使定义的别名不失效的情况下，使用系统的ls命令显示当前目录中的命令列表：

```
\ls
```

删除别名ls。

```
unalias ls  
alias cde='cd /etc'
```

显示别名ls，确认删除别名已经生效。

```
alias ls
```

最后再用命令ls显示当前目录中的文件列表。

```
ls
```

写出定义别名cde为：cd /etc的命令。

```
alias cde='cd /etc'
```

二. 通配符的使用

以tux1帐号登入系统，用ls /bin/*sh 命令显示系统中的shell种类。

```
ls /bin/*sh
```

进入/etc目录并且列出该目录下的所有文件。

```
cd /etc
```

使用带通配符的ls命令列出文件名：

(1) 文件名以conf结尾的文件

```
ls *conf
```

(2) 文件名以一个字母d或D开头的文件

```
ls [d,D]*
```

(3) 文件名的第五位包含一个字母o的文件

```
ls ?????o*
```

(4) 文件名包含文字tab且不区分大小写字母的文件

```
ls *[t,T][a,A][b,B]*
```

(5) 文件名以一位数字结尾的文件

```
ls *[0-9][0-9]
```

(6) 文件名不以一位数字结尾的文件

```
ls *[0-9]
```

执行命令ls -d ?[!y]*[e-f]将会发生什么情况？

匹配不是y开头的且以e或f结尾的文件。

返回到你的主目录。

```
cd ~
```

三. 重定向

使用cat命令重定向创建一个名为junk的文件，该文件包含了一些文本行。键入一些行以后，中止cat命令返回到shell提示符。查看你刚才创建的文件的内容。使用重定向在junk文件中再添加一些行。然后查看该文件的内容，检查你所输入的内容是否完整。

```
cat junk
```

输入一些内容之后：wq

将其标准输出重定向到文件cat.out,标准错误重定向到cat.err。

```
cat 1>>cat.out
```

```
cat 2>>cat.err
```

四. 管道

使用管道统计你的当前目录里的文件数，不要手动去计数。

```
ls -l|wc -l
```

执行ls -l> tempfile ; wc -l tempfile ; rm tempfile 命令与第7题作用一样吗？为什么一样或不一样？

第七题在哪.....

使用ls命令并把输出结果保存在名为tempfile2的文件中，然后统计并显示文件数。

```
ls -l>tempfile2
```

```
ls -l|wc -l
```

五. 命令组

1. 在同一个命令行上，显示当前的系统日期和登入的所有用户，包括一些说明注释，结果存入到一个文件中。

```
date>>tempfile;who>>tempfile
```

六. 进程环境

显示在你的当前进程环境中定义的所有变量。显示所有环境变量。

```
env
```

用命令 `$localvar=10` 给本地变量`localvar`赋值。

```
localvar=10  
echo $localvar
```

用 `$bash` 命令启动一个子shell。再显示变量`localvar`的值。这次和28中显示的为什么不同？

```
/bin/bash  
echo $localvar  
无法显示localvar
```

七. 使用grep和正则表达式

复制`/etc/passwd`文件到你的主目录，并换名为`lpasswd`。

```
cp /etc/passwd ~/lpasswd
```

搜寻在`lpasswd`文件中所有首字母是`s`的行,并用`wc,grep`和管道组合成一条命令统计`lpasswd`文件中首字母是`s`的行数。

```
grep '^s' ~/lpasswd|wc -l
```

重复上面的操作，但这次只显示符合条件的行的计数。

```
grep '^s' ~/lpasswd
```

尝试用`grep` 和正则表达式在`lpasswd`中查找其它你想要查找的信息。

略

八. `Shell程序及其三种运行方式

使用`vi`编辑器,编辑一个名为`setdir`的脚本文件:

```
dir1=/tmp  
dir2=/usr  
echo $dir1 $dir2
```

分别用学过的三种不同的运行方式运行该脚本,每次运行一种方式后,再次执行`echo $dir1 $dir2`

三种方式:

```
sh setdir.sh  
./setdir.sh  
. setdir.sh
```

实验五

一. 引号的使用

双引号和单引号的使用。依次输入下列命令：

```
$string =" hello world!"
```

```
$echo "$string"
```

```
$echo '$string'
```

比较(2)和(3)的结果，为什么不同？

(2)中输出hello world

(3)中输出string

反引号的使用。进入你的主目录，输入：`$echo "the current work directory is: `pwd`"`会显示什么结果？解释反引号的作用。

```
echo "the current work directory is: `pwd`"
```

输出为the current work directory is:/home/username

二. shell编程

1. 位置变量参数和特定变量参数.验证讲义中Param脚本的运行结果。

(1)在你的主目录/home/student下建立目录bin,并进入到bin目录。

用vi编辑器编辑本脚本，内容如下：

```
echo " script name: "  
echo $0  
echo "show arguments:"  
echo $*  
echo "my process id:"  
echo $$  
echo "Did my script go with any errors?"  
echo $?
```

(2)执行本脚本, 显示什么内容？

进入终端，执行：

```
cd 实验五的根目录
```

```
sh 实验5 .sh hello world
```

显示内容为：

```
script name:  
/home/zy/桌面/Linux/实验5.sh  
show arguments:  
  
my process id:  
19416  
Did my script go with any errors?  
0
```

(3)结合执行结果，解释脚本中0,*,,\$?的意义。

```
$$ Shell本身的PID (ProcessID)  
$? 最后运行的命令的结束代码 (返回值)  
$* 所有参数列表。  
$0 Shell本身的文件名
```

4. 表达式。依次进行以下操作：

(1) 给变量i赋值为10，变量j赋值为5。

(2) 分别用算术扩展\${()}和expr两种方式求表达式i*j的值。

```
i=10
j=5
r=$((i*5-j))
echo $r
x=`expr $i \* 5`
y=`expr $x + $j`
echo $y
```

5.test命令。

(1)字符串操作符。依次输入以下命令,并解释执行结果。

```
$str1=abcd
$str2="abcd "
$test "$str1" = "$str2"
$echo $?
```

本脚本输出结果为 1, 意思为str1与str2不相等。

(2)整数操作符。依次输入以下命令,并解释执行结果。

```
$str1=1234
$str2=01234
$[ "$str1" -eq "$str2" ]
$echo $?
```

本脚本输出结果为 0, 意思为str1与str2相等。

(3)文件操作符。依次输入以下命令,并解释执行结果。

```
$touch aa
$ [ -w aa ]
$ test -w aa
$ echo $?
```

输出 0, aa具有写权限。

```
$[ ! -d aa ]
$ echo $?
```

输出 1, aa不是一个目录。

(4)逻辑操作符。利用(3)中的文件aa执行如下命令,并解释结果。

```
$[ -r aa -a -x aa ]
$echo $?
```

输出 1, aa可读,但是不可执行。

6. shell控制语句. 编写一个脚本, 要求如下:

- (1)把用户名作为输入参数(位置变量参数),每隔60秒检查一次用户是否登陆,直至该用户登陆。
- (2)如果执行脚本时输入的参数数目不为1, 则输出"incorrect number of arguments"提示信息, 并退出执行脚本。
- (3)该用户登陆后, 显示"the user 用户名 has logged on"。其中用户名对应于用户执行时输入的参数信息。

提示:

- (1)用 `["$###" -ne 1]` 测试输入参数个数是否为1
- (2)用 `who|grep "^$user "> /dev/null` 判断用户\$user是否登陆
- (3)用 `sleep 60` 表示每隔60秒进行某项操作。

脚本内容:

```
if [ $# -ne 1 ]; then
    echo "Incorrect number of arguments"
    exit
else
    while true
    do
        if who | grep "^$1 " > /dev/null; then
            echo "the user $1 has logged on"
        fi
        sleep 60
    done
fi
```

实验六

一.用户和组的管理

列出你的系统中定义的所有帐号。

```
compgen -u
```

查看是否所有帐号都有密码？如何确定是否有密码？

```
sudo cat /etc/shadow
```

列出的所有数据中，第二列数据为密码的哈希值，如果密码不是一个哈希字符串，则代表没有密码。

以root帐号登录，增加一个帐号tux3。

```
sudo -s
```

```
adduser tux3
```

查看/etc/shadow文件。帐号tux3的密码是什么？

登入另一个终端

```
sudo cat /etc/shadow
```

密码为“！”，代表没有密码。

给出帐号tux3的密码“penguin3”。再次查看/etc/shadow 文件，这次帐号tux3的密码是什么？

```
sudo cat /etc/shadow
```

密码（哈希字符串）为：

```
$6$8sbhzr9N$H2.6l20UxLBIgA.SdxVRH7xguTByAvGUT4b8x4TfwR3yURP6HojTsjnxVBUSykdXv1TI5e41nXjcVO.U4Z7OV
```

查看/etc/passwd- 文件的内容。该文件存储的是什么内容？

```
sudo cat /etc/passwd-
```

储存的是本机上用户的信息，包括用户名，密码，根目录等信息。

帐号tux3所在的组是什么？

```
groups tux3
```

所在的组为tux3

增加一个组“penguins”。

```
groupadd penguins
```

把帐号tux3 加入到组penguins中。

```
gpasswd -a tux3 penguins
```

在哪个文件中可以看到这个变化？解释一下如何变化的。

不退出root帐号，用一个命令转入到tux3用户帐号，要求改变身份时，同时变更工作目录。如果不要求变更工作目录那应该怎么做？

改变工作目录：

```
su - tux3
```

不改变工作目录：

```
su tux3
```

用一个命令从tux3用户帐号退回到root帐号。

```
sudo -s
```

二.进程管理和任务自动化

1. 使用vi编辑器，在你的用户主目录的bin目录下创建一个脚本文件myclock，内容如下：

```
while true
do
date
sleep 10
done
```

在前台运行该脚本文件。

```
cd ~/bin
sh myclock.sh
```

挂起刚才运行的任务。

```
ctrl + z
```

列出系统中运行的所有任务，在后台重新运行上面的任务。

```
jobs
```

查看系统中正在运行的所有进程的详细信息（包括前台和后台的），这里面有刚才启动的任务myclock吗？

```
ps -A
```

没有

只查看由root用户启动的进程信息

```
ps -u root
```

把myclock任务转为前台运行，屏幕显示时间后退出该任务的运行。

```
bg sh myclock.sh
ctrl + c
```

再次在后台运行myclock脚本文件。

```
sh myclock.sh &
```

使用kill命令停止shell脚本myclock的运行。

```
kill [myclock的进程号]
```

三.文件的压缩和归档

拷贝/usr/share目录到用户主目录下,改目录名为backup,然后建立一个包含backup子目录中全部文件的名为backup.tar的档案文件

```
cp -r /usr/share ~/backup
tar -cf backup.tar backup
```

在当前目录下，建立一个包含backup子目录中全部文件的经过gzip压缩的名为backup.tar.gz的档案文件

```
tar -czf backup.tar.gz backup
```

试着释放backup.tar.gz文件.

```
tar -xf backup.tar.gz
```

四.网络配置

1. 验证一下网络配置相关命令的使用。

```
hostname  
ifconfig  
ifup {interface}  
ifdown {interface}  
ping www.baidu.com  
netstat  
nslookup
```

2. 验证netconfig程序的使用。

控制台输入：`netconfig`

进入网络设置（貌似ubuntu不行）