

浙江大学实验报告

课程名称: LINUX 应用技术基础 实验类型: 验证型

实验项目名称: Linux 文件权限、文件共享、进程、管道和重定向

学生姓名: 王冠颖 专业: 计科 学号: 3120103841

电子邮件地址: 543581485@qq.com

实验日期: 2014 年 5 月 29 日

一、实验目的:

- 学习 Linux 的文件访问权限, 用户的类型和文件访问权限的类型
- 学习如何确定一个文件的访问权限
- 学习如何设置和改变一个文件的访问权限
- 学习如何在文件或目录的创建时设置缺省访问权限
- 学习理解硬链接、符号链接
- 学习理解链接和文件访问权限之间的关系
- 学习了解 Linux 进程的属性
- 学习理解 Linux 的前台进程、后台进程及守护进程
- 学习理解 Linux 命令的顺序执行和并发执行
- 学习使用挂起进程操作和终止进程操作
- 了解系统中 Linux 进程的层次结构
- 学习使用 Linux 定时作业调度
- 学习使用 Linux 的 I/O 重定向操作和管道操作
- 学习使用 id、chmod、chgrp、chown、umask、ln、ps、kill、fg、bg、job、top、pstree、crontab、at、|、<、>等命令

二、实验内容:

1. 根据下列的要求, 写出得到这些信息的会话过程:

- 1) 你的用户名;
- 2) 你的用户 ID; id

```
wgy@localhost:~
文件(F) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)

[wgy@localhost ~]$ who am i
wgy      pts/0      2014-05-29 19:40 (:0.0)
[wgy@localhost ~]$ id
uid=500(wgy) gid=500(wgy) groups=500(wgy) context=user_u:system_r:unconfined_t
[wgy@localhost ~]$
```

用户名: wgy; ID: 500

2. 系统管理员给你的主目录设置的权限是什么? 使用什么命令来得到这个答案, 给出你的会话过程。

```
[wgy@localhost ~]$ ls -ld /home/wgy
drwx----- 18 wgy wgy 4096 05-29 19:39 /home/wgy
```

3. 在你的系统中有文件或目录分别是: / 、 /etc/passwd 、 /bin/df 、 ~ 。 用长列表格式显示这些文件或目录, 并填写下列表格。

```
[wgy@localhost ~]$ ls -lad /
drwxr-xr-x 23 root root 4096 05-29 19:37 /
[wgy@localhost ~]$ ls -lad /etc/passwd
-rw-r--r-- 1 root root 1870 03-25 19:37 /etc/passwd
[wgy@localhost ~]$ ls -lad /bin/df
-rwxr-xr-x 1 root root 41476 2006-11-28 /bin/df
[wgy@localhost ~]$ ls -lad ~
drwx----- 18 wgy wgy 4096 05-29 19:39 /home/wgy
[wgy@localhost ~]$
```

Rwx:读、写、可执行。

文件	文件类型	存取权限	链接数	所有者	组	文件大小
/	d(目录)	Rwx rx rx	23	root	root	4096
/etc/passwd	- (普通)	Rw r r	1	root	root	1870
/bin/df	- (普通)	Rwx rx rx	1	root	root	41476
~	d(目录)	Rwx - -	18	wgy	wgy	4096

4. 在你的主目录中的 temp、professional 和 personal 三个子目录, 设置使自己 (owner) 拥有读、写、执行 3 种访问权限, 设置其它用户只有读和执行权限。在 ~/temp 目录下创建名为 d1、d2 和 d3 的目录。在 d1 目录下, 用 touch 命令创建一个名为 fl 的空文件。给出 d1、d2、d3 和 fl 的访问权限。给出完成这些工作的会话。

```
[wgy@localhost ~]$ chmod u=rwx /home/wgy/temp /home/wgy/personal /home/wgy/profe
ssional

[wgy@localhost ~]$ cd /home/wgy/temp
[wgy@localhost temp]$ mkdir d1;mkdir d2;mkdir d3
```

```
[wgy@localhost temp]$ cd d1
[wgy@localhost d1]$ touch f1

drwxrwxr-x 2 wgy wgy 4096 05-29 20:06 d1
[wgy@localhost temp]$ ls -ld d2
drwxrwxr-x 2 wgy wgy 4096 05-29 20:05 d2
[wgy@localhost temp]$ ls -ld d3
drwxrwxr-x 2 wgy wgy 4096 05-29 20:05 d3
[wgy@localhost temp]$ ls -lad d1/f1
-rw-rw-r-- 1 wgy wgy 0 05-29 20:06 d1/f1
[wgy@localhost temp]$
```

5. 在~/temp 目录下创建名为 d1、d2 和 d3 的目录。把文件 smallFile 拷贝到 d1 目录下，长列表格式显示文件 smallFile，显示的内容包括 inode 号、访问权限、硬链接数、文件大小。给出完成这些工作的会话。

创建过程在第 4 题。

```
[wgy@localhost ~]$ cp smallFile temp/d1
[wgy@localhost ~]$ ls -li smallFile
3430731 -rw-rw-r-- 1 wgy wgy 1042 05-29 20:45 smallFile
```

6. 在~/temp 目录下，把当前目录改变成 d2。创建一个名字为 newFile.hard 硬链接到 d1 目录下的 smallFile 文件。长列表格式显示 newFile.hard 文件，与 smallFile 文件的属性进行比较。你如何确定 smallFile 和 smallFile.hard 是同一文件的两个名字，是链接数吗？给出你的会话过程。

```
[wgy@localhost ~]$ cd d2
[wgy@localhost d2]$ ln ~/temp/d1/smallFile newFile.hard

[wgy@localhost d2]$ ls -li newFile.hard
3430744 -rw-rw-r-- 2 wgy wgy 1042 05-29 20:46 newFile.hard
[wgy@localhost d2]$ ls -li ~/temp/d1/smallFile
3430744 -rw-rw-r-- 2 wgy wgy 1042 05-29 20:46 /home/wgy/temp/d1/smallF:
└─
```

两个文件的索引节点号相同，是同一个文件。

7. 创建一个名字为~/temp/d2/smallFile.soft 软链接到~/temp/d1/smallFile 文件。长列表格式显示 smallFile.soft 文件，比较这两个文件的属性。你如何确定 smallFile 和 smallFile.soft 是两个不同的文件？是这两个文件的大小吗？给出你的会话过程。

```
[wgy@localhost ~]$ ln -s ~/temp/d1/smallFile ~/temp/d2/smallFile.soft

[wgy@localhost ~]$ ls -li ~/temp/d2/smallFile.soft
3430720 lrwxrwxrwx 1 wgy wgy 27 05-30 13:55 /home/wgy/temp/d2/smallFile.soft ->
/home/wgy/temp/d1/smallFile
└─
```

不是同一个文件，inode 不同。

8. 在你使用的 Linux 系统中，有多少进程在运行？进程 init、bash、ps 的 PID 是多少？init、bash 和 ps 进程的父进程是哪一个？这些父进程的 ID 是什么？给出你得到这些信息的会话过程。

```
[wgy@localhost ~]$ ps -A1
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S   0    1    0  0  75   0 -   514 -          ?        00:00:02 init

0 S   500  4208  4205  0  75   0 -  1405 wait  pts/0    00:00:00 bash
0 R   500  4250  4208  0  77   0 -  1326 -          pts/0    00:00:00 ps
```

PID 分别为 1、4208，4205。PPID 分别为 0、4205、4208，后两个是 bash 和 gnome-pty-help 行程。

9. linux 系统中，进程可以在前台或后台运行。前台进程在运行结束前一直控制着终端。若干个命令用分号（；）分隔形成一个命令行，用圆括号把多个命令挂起来，他们就在一个进程里执行。使用“&”符作为命令分隔符，命令将并发执行。可以在命令行末尾加“&”使之成为后台命令。

请用一行命令实现以下功能：它 1 小时（实验中可以用 1 分钟代替）分钟后在屏幕上显示文字“Time for Lunch!”来提醒你去吃午餐。给出会话过程。

```
[wgy@localhost ~]$ cat >lunch
Time for lunch !

[wgy@localhost ~]$ (sleep 1m;cat lunch)&
[1] 4281
[wgy@localhost ~]$ ls lunch
lunch
[wgy@localhost ~]$ who am i
wgy pts/0 2014-05-30 13:54 (:0.0)
[wgy@localhost ~]$ Time for lunch !

[1]+ Done ( sleep 1m; cat lunch )
```

将 sleep 进程调入后台后，可以同时在前台继续输入命令。最后一行是一分钟以后的结果。

10. 写一命令行，使得 date 、 uname -a 、 who 和 ps 并发执行。给出会话过程。

```
[wgy@localhost ~]$ (date&uname -a&who&ps);
2014年 05月 30日 星期五 14:33:51 CST
Linux localhost.localdomain 2.6.18-53.el5xen #1 SMP Wed Oct 10 17:06:12 EDT 2007 i686
i686 i386 GNU/Linux
wgy pts/0 2014-05-30 13:54 (:0.0)
PID TTY TIME CMD
4208 pts/0 00:00:00 bash
4328 pts/0 00:00:00 bash
4332 pts/0 00:00:00 ps
```

11. 写一命令行，先后执行 date 、 uname -a 、 who 和 ps 命令，后面 3 个命令的执行条件是：当只有前面一个命令执行成功后，才能执行后面一个命令。给

出会话过程。

```
[wgy@localhost ~]$ (date && uname -a && who &&ps);
2014年 05月 30日 星期五 14:34:08 CST
Linux localhost.localdomain 2.6.18-53.el5xen #1 SMP Wed Oct 10 17:06:12 EDT 2007 i686
i686 i386 GNU/Linux
wgy      pts/0      2014-05-30 13:54 (:0.0)
  PID TTY          TIME CMD
  4208 pts/0      00:00:00 bash
  4335 pts/0      00:00:00 bash
  4339 pts/0      00:00:00 ps
```

12. 在 shell 下执行下面的命令。3 个 pwd 命令的运行结果是什么？

```
$ pwd
```

```
$ bash
```

```
$ cd /usr
```

```
$ pwd
```

```
$<Ctrl-D> #终止 shell
```

```
$ pwd
```

“\$”为系统提示符

```
[wgy@localhost ~]$ pwd
/home/wgy
[wgy@localhost ~]$ bash
[wgy@localhost ~]$ cd /usr
[wgy@localhost usr]$ pwd
/usr
[wgy@localhost usr]$ exit
[wgy@localhost ~]$ pwd
/home/wgy
```

13. 搜索并显示你主目录下 foobar 文件的绝对路径，错误信息重定向到/dev/null 中。给出你的会话。（提示：若没有 foobar 文件，可以创建一个）

```
[wgy@localhost ~]$ find ~ -name foobar 2>/dev/null
/home/wgy/temp/foobar
```

14. 搜索你主目录下 foobar 文件，保存它的绝对路径到 foobar.path 文件中，错误信息写到/dev/null 中，再显示 foobar.path 文件的内容。给出会话过程。

```
[wgy@localhost ~]$ find ~ -name foobar >foobar.path 2>/dev/null
[wgy@localhost ~]$ cat foobar.path
/home/wgy/temp/foobar
```

15. cat 程序（命令），输入从标准输入设备中读入，输出送到标准输出设备中。现在运行这个程序，要求输入从 student.records 中读入，输出结果重定向到 output.data 文件中，错误重定向到 error.log 文件中。用一条命令来实现上述过程。

```
[wgy@localhost ~]$ cat <student.records 1>output.data 2>error.log
```

16. 写出一个命令将 shell 的标准输入 (stdin) 更改到当前目录下名为 fdata 文件，标准输出 (stdout) 更改到当前目录下名为 fout 的文件。如果 fdata 文件包含下面的内容，那么在命令执行后会发生什么？

```
echo -n "The time now is:"
date
echo -n "The users presently logged on are:"
who
```

提示：使用 exec 命令

```
[wgy@localhost ~]$ bash <./fdata> ./fout
[wgy@localhost ~]$ cat fout
The time now is:2014年 05月 30日 星期五 15:13:10 CST
The users presently logged on are:wgy pts/0 2014-05-30 15:13 (:0.0)
[wgy@localhost ~]$ exec cat fout
```

输出如图所示，最后一句执行完后终端关闭。

17. 计算命令 ls -l 的输出中的字符数、单词数和行数，并把它显示在显示器上。给出命令和输出结果。

```
[wgy@localhost ~]$ ls -l | wc -lwc
    30    234   1393
```

18. 在/bin 目录下有多少个普通文件、目录文件和符号链接文件？如何得到这个答案？

```
[wgy@localhost ~]$ find /bin -type f | wc -l
88
[wgy@localhost ~]$ find /bin -type d | wc -l
1
[wgy@localhost ~]$ find /bin -type l | wc -l
19
```

三、实验环境：

vmware workstation 9.0

Linux Redhat 5 32 位

四、讨论心得：

这次感觉比上次好了许多，指令多看几遍，不用过于依赖书本了。

有些地方，像文件重定向、管道等等，感觉还是很不熟练，原理也不太了解，还需要进一步学习。