

《计算机操作系统》实验报告

实验题目：Linux 操作系统基本命令

姓名：严昕宇 学号：20121802 实验日期：2022.10.26

实验环境：

实验设备：Lenovo Thinkbook16+ 2022

操作系统：Ubuntu 22.04.1 LTS 64 位

实验目的：

1. 了解 Linux 运行环境，熟悉交互式分时系统、多用户环境的运行机制
2. 练习 Linux 系统命令接口的使用，学会 Linux 基本命令、后台命令、管道命令等命令的操作要点

实验内容：

通过终端或虚拟终端，在基于字符的交互界面中进行 Shell 的基本命令的操作

操作过程 1：

- ① 执行 pwd 查看当前目录
- ② 用 who am i 看看当前用户信息
- ③ 通过 who 看看有谁在系统中
- ④ 用 vmstat 显示系统状态

结果 1：

```
yanxinyu@Thinkbook16-2022:~$ pwd
/home/yanxinyu
yanxinyu@Thinkbook16-2022:~$ who am i
yanxinyu@Thinkbook16-2022:~$ whoami
yanxinyu
yanxinyu@Thinkbook16-2022:~$ who
yanxinyu tty2          2022-10-26 20:51 (tty2)
yanxinyu@Thinkbook16-2022:~$ vmstat
procs -----memory----- ---swap-- -----io----- -system-- -----cpu-----
 r b 交换 空闲 缓冲 缓存   si  so   bi   bo   in   cs us sy id wa st
 3  0      0 1868820 40952 933716    0    0   900   60  196 425  2  9 89  0  0
yanxinyu@Thinkbook16-2022:~$
```

值得注意的是，Ubuntu 22 系统中，输入 who am i 指令时不会进行输出，因此在查阅资料后，发现只能使用 whoami 指令。

但是两者存在区别，whoami 显示的是当前“操作用户”的用户名，而 who am i 显示的是“登录用户”的用户名。

思考：你的用户名、用户标识、组名、组标识是什么？当前你处在系统的哪个位置中？现在有哪些用户和你一块儿共享系统？

答：用户名：yanxinyu，用户标识为 1000，组名 1000，组标识 1000。目前处于/home/yanxinyu 家目录下。目前和 root 用户一起共享系统。

```
yanxinyu@Thinkbook16-2022:~$ id
用户id=1000(yanxinyu) 组id=1000(yanxinyu) 组=1000(yanxinyu),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),
122(lpadmin),134(lxd),135(sambashare)
```

操作过程 2：

- ① 执行 `cat > mytext.txt` 通过键盘输入“hello world!”
- ② 执行 `cat mytext.txt`
- ③ 执行
`ln mytext.txt mytext2.dat`
`cat mytext2.dat`
- ④ 执行 `ls -l mytext?.*`

结果 2：

```
yanxinyu@Thinkbook16-2022:~$ cat >mytext.txt
hello,world!
^C
yanxinyu@Thinkbook16-2022:~$ cat mytext.txt
hello,world!
yanxinyu@Thinkbook16-2022:~$ ln mytext.txt mytext2.dat
yanxinyu@Thinkbook16-2022:~$ cat mytext2.dat
hello,world!
yanxinyu@Thinkbook16-2022:~$ ls -l mytext?.*
-rw-rw-r-- 2 yanxinyu yanxinyu 13 10月 26 21:17 mytext2.dat
yanxinyu@Thinkbook16-2022:~$
```

可以看到，通过指令 `ln` 创建出来的链接文件 `mytext2.dat` 的内容与 `mytext.txt` 一致。

思考：文件链接是什么意思？有什么作用？

答：链接文件是 Linux 文件系统的特性。如需要在系统上维护同一文件的两份或多份副本，除了保存多份单独的物理文件副本之外，还可以采用保存一份物理文件副本和多个虚拟副本的方法。这种虚拟的副本就称为链接。链接是目录中指向文件真实位置的占位符。在 Linux 中有两种不同类型的文件链接——符号链接(软链接)与硬链接。此处使用 `ln` 命令创建得到的，即使用的是符号链接。

文件的链接是指文件在另一个目录中的别名，实际上是给系统中已有的某个文件指定另外一个可用于访问它的名称。文件的所有链接都指向同一个物理文件，可类比 C++ 中的引用 &。

文件链接的作用：

- 1) 对于链接出的文件，可以指定不同的访问权限，以控制对信息的共享和安全性的问题
- 2) 如果链接指向目录，用户可利用该链接直接进入被链接的目录，免去原本冗长的路径名
- 3) 即使我们删除这个链接，也不会破坏原来的目录，更加安全

操作过程 3:

① 执行 `ls -l`

② 执行

`cd /lib`

`ls -l|more`

结果 3:

```
yanxinyu@Thinkbook16-2022:~$ ls -l
总用量 44
drwxr-xr-x 2 yanxinyu yanxinyu 4096 10月 21 16:07 公共的
drwxr-xr-x 2 yanxinyu yanxinyu 4096 10月 21 16:07 模板
drwxr-xr-x 2 yanxinyu yanxinyu 4096 10月 21 16:07 视频
drwxr-xr-x 2 yanxinyu yanxinyu 4096 10月 21 16:07 图片
drwxr-xr-x 2 yanxinyu yanxinyu 4096 10月 21 16:07 文档
drwxr-xr-x 2 yanxinyu yanxinyu 4096 10月 21 16:07 下载
drwxr-xr-x 2 yanxinyu yanxinyu 4096 10月 21 16:07 音乐
drwxr-xr-x 2 yanxinyu yanxinyu 4096 10月 21 16:07 桌面
-rw-rw-r-- 2 yanxinyu yanxinyu 13 10月 26 21:17 mytext2.dat
-rw-rw-r-- 2 yanxinyu yanxinyu 13 10月 26 21:17 mytext.txt
drwx----- 4 yanxinyu yanxinyu 4096 10月 21 16:21 snap
```

可以看到当前目录的内容，还包含当前目录下文件的文件类型、文件的存取控制权限、i 节点号、文件 属主、文件属组、文件大小、建立日期等信息。

`/lib` 目录(系统函数):

```
yanxinyu@Thinkbook16-2022:/lib$ ls -l
总用量 648
drwxr-xr-x 2 root root 4096 8月 9 19:49 app
drwxr-xr-x 2 root root 4096 8月 9 19:49 apparmor
drwxr-xr-x 5 root root 4096 10月 21 15:27 apt
drwxr-xr-x 3 root root 4096 8月 9 19:49 aspell
drwxr-xr-x 2 root root 4096 4月 8 2022 binfmt.d
drwxr-xr-x 2 root root 4096 8月 9 19:49 bluetooth
drwxr-xr-x 2 root root 4096 8月 9 19:50 brltty
-rwxr-xr-x 1 root root 1075 12月 8 2021 cnf-update-db
-rwxr-xr-x 1 root root 3565 12月 8 2021 command-not-found
drwxr-xr-x 2 root root 4096 8月 9 19:48 console-setup
lrwxrwxrwx 1 root root 21 10月 20 00:17 cpp -> /etc/alternatives/cpp
drwxr-xr-x 2 root root 4096 8月 9 19:50 crda
drwxr-xr-x 10 root root 4096 8月 9 19:49 cups
drwxr-xr-x 2 root root 4096 8月 9 19:48 dbus-1.0
drwxr-xr-x 3 root root 4096 8月 9 19:50 debug
drwxr-xr-x 3 root root 4096 8月 9 19:48 dpkg
drwxr-xr-x 3 root root 4096 8月 9 19:49 enacsen-common
drwxr-xr-x 2 root root 4096 10月 21 15:27 environment.d
drwxr-xr-x 7 root root 4096 8月 9 19:49 evolution-data-server
```

/etc 目录(系统配置用的数据文件):

```
yanxinyu@Thinkbook16-2022:/lib$ cd /etc
yanxinyu@Thinkbook16-2022:/etc$ ls -l
总用量 1128
drwxr-xr-x  3 root root    4096  8月  9 19:50 acpi
-rw-r--r--  1 root root    3028  8月  9 19:48 adduser.conf
drwxr-xr-x  3 root root    4096  8月  9 19:49 alsa
drwxr-xr-x  2 root root    4096 10月 21 16:16 alternatives
-rw-r--r--  1 root root      335  3月 23  2022 anacrontab
-rw-r--r--  1 root root      433  3月 23  2022 apg.conf
drwxr-xr-x  5 root root    4096  8月  9 19:49 apm
drwxr-xr-x  3 root root    4096  8月  9 19:50 apparmor
drwxr-xr-x  7 root root    4096 10月 21 15:29 apparmor.d
drwxr-xr-x  4 root root    4096  8月  9 19:51 appport
-rw-r--r--  1 root root      769  2月 23  2022 appstream.conf
drwxr-xr-x  8 root root    4096 10月 20 00:36 apt
drwxr-xr-x  3 root root    4096  8月  9 19:51 avahi
-rw-r--r--  1 root root    2319  1月  7  2022 bash.bashrc
-rw-r--r--  1 root root       45 11月 11  2021 bash_completion
drwxr-xr-x  2 root root    4096  8月  9 19:51 bash_completion.d
-rw-r--r--  1 root root      367 12月 16  2020 bindresvport.blacklist
drwxr-xr-x  2 root root    4096  4月  8  2022 binfmt.d
```

/bin 目录(可执行程序):

```
yanxinyu@Thinkbook16-2022:/etc$ cd /bin
yanxinyu@Thinkbook16-2022:/bin$ ls -l
总用量 168716
-rwxr-xr-x  1 root root    51632  2月  8  2022 '['
-rwxr-xr-x  1 root root   35344  6月 21 12:46 aa-enabled
-rwxr-xr-x  1 root root   35344  6月 21 12:46 aa-exec
-rwxr-xr-x  1 root root   31248  6月 21 12:46 aa-features-abi
-rwxr-xr-x  1 root root   22912  1月 12  2022 aconnect
-rwxr-xr-x  1 root root   19016  1月 25  2022 acpi_listen
-rwxr-xr-x  1 root root   14478  7月 26 19:37 add-apt-repository
-rwxr-xr-x  1 root root   14712  2月 21  2022 addpart
-rwxr-xr-x  1 root root  150376  3月 25  2022 aircan-discover
-rwxr-xr-x  1 root root   43456  1月 12  2022 alsabat
-rwxr-xr-x  1 root root   85328  1月 12  2022 alsaloop
-rwxr-xr-x  1 root root   86312  1月 12  2022 alsamixer
-rwxr-xr-x  1 root root   76160  1月 12  2022 alsatplg
-rwxr-xr-x  1 root root   31576  1月 12  2022 alsaucm
-rwxr-xr-x  1 root root   31112  1月 12  2022 amidi
-rwxr-xr-x  1 root root   63952  1月 12  2022 amixer
-rwxr-xr-x  1 root root     274  3月 23  2022 apg
```

/home 目录(每个用户主目录):

```
yanxinyu@Thinkbook16-2022:/bin$ cd /home
yanxinyu@Thinkbook16-2022:/home$ ls -l
总用量 4
drwxr-x--- 16 yanxinyu yanxinyu 4096 10月 26 21:18 yanxinyu
```

思考：Linux 文件类型有哪几种？文件的存取控制模式如何描述？

答：Linux 中一共有七种文件类型，即普通文件(-)、目录文件(d)、字符设备文件(c)、块设备文件(b)、套接字文件(s)、符号链接文件(l)和管道文件(p)。

文件属性	文件类型
-	常规文件，file
d	目录文件
b	Block Device，即块设备文件； 如硬盘支持以 block 为单位进行随机访问
c	Character Device，即字符设备文件； 如键盘支持以 character 为单位进行线性访问
l	Symbolic Link，即符号链接文件，又称软链接文件
p	Pipe，即命名管道文件
s	Socket，即套接字文件，用于实现两个进程进行通信

存取控制模式描述主要针对三类用户：

- 文件的所有者：owner，用 u 标识；
- 文件的所有组：group，用 g 标识；
- 其他人：other，用 o 标识。不包含在文件所有者和所有组内的系统其他用户。

对这三种所有人分别设置三种文件权限：

- 读 r——readable；
- 写 w——writeable；
- 执行 x——executable (如脚本、二进制文件、目录等文件具有此权限)

操作过程 4：

1. 执行 `chmod 751 mytext.txt`
`ls -l mytext.txt`
2. 执行 `sudo chown root mytext.txt`

结果 4：

```
yanxinyu@Thinkbook16-2022:~$  
yanxinyu@Thinkbook16-2022:~$ chmod 751 mytext.txt  
yanxinyu@Thinkbook16-2022:~$ ls -l mytext.txt  
-rwxr-x--x 1 yanxinyu yanxinyu 13 10月 26 22:36 mytext.txt  
yanxinyu@Thinkbook16-2022:~$ sudo chown root mytext.txt  
yanxinyu@Thinkbook16-2022:~$ ls -l mytext.txt  
-rwxr-x--x 1 root yanxinyu 13 10月 26 22:36 mytext.txt
```

思考：执行了上述操作后，若想再修改文件，看能不能执行。为什么？

答：不能执行。如图，输入第一条指令 `chmod 751 mytext.txt` 后，即给 `mytext.txt` 的属主分配读、写、执行(7)的权限，给 `mytext.txt` 的所在组分配读、执行(5)的权限，给其他用户分配执行(1)的权限，即此时其他用户就只有对此文件执行的权限。然后将所有人改为 `yanthinkn` 之后，原本用户(`yanxinyu`)就成为其他用户，于是试图修改文件的操作会提示权限不够。

操作过程 5:

1. 执行

```
ps -ef
```

2. 执行

```
sleep 5 & wait $!
```

结果 5:

```
yanxinyu@Thinkbook16-2022:~$ ps -ef
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	20:50	?	00:00:03	/sbin/init auto noprompt splash
root	2	0	0	20:50	?	00:00:00	[kthreadd]
root	3	2	0	20:50	?	00:00:00	[rcu_gp]
root	4	2	0	20:50	?	00:00:00	[rcu_par_gp]
root	5	2	0	20:50	?	00:00:00	[netns]
root	7	2	0	20:50	?	00:00:00	[kworker/0:0H-events_highpri]
root	9	2	0	20:50	?	00:00:00	[kworker/0:1H-events_highpri]
root	10	2	0	20:50	?	00:00:00	[mm_percpu_wq]
root	11	2	0	20:50	?	00:00:00	[rcu_tasks_rude_]
root	12	2	0	20:50	?	00:00:00	[rcu_tasks_trace]
root	13	2	0	20:50	?	00:00:00	[ksoftirqd/0]
root	14	2	0	20:50	?	00:00:01	[rcu_sched]
root	15	2	0	20:50	?	00:00:00	[migration/0]
root	16	2	0	20:50	?	00:00:00	[idle_inject/0]
root	18	2	0	20:50	?	00:00:00	[cpuhp/0]
root	19	2	0	20:50	?	00:00:00	[cpuhp/1]
root	20	2	0	20:50	?	00:00:00	[idle_inject/1]
root	21	2	0	20:50	?	00:00:01	[migration/1]
root	22	2	0	20:50	?	00:00:00	[ksoftirqd/1]
root	24	2	0	20:50	?	00:00:00	[kworker/1:0H-events_highpri]
root	25	2	0	20:50	?	00:00:00	[kdevtmpfs]
root	26	2	0	20:50	?	00:00:00	[inet_frag_wq]
root	27	2	0	20:50	?	00:00:00	[kauditd]
root	29	2	0	20:50	?	00:00:00	[khungtaskd]
root	30	2	0	20:50	?	00:00:00	[oom_reaper]
root	31	2	0	20:50	?	00:00:00	[writeback]

```
yanxinyu@Thinkbook16-2022:~$ sleep 5 & wait $!
```

```
[1] 5798
```

```
[1]+  已完成                sleep 5
```

`sleep 5 & wait $!`指令中先执行 `sleep 5`，启动一个等待 5s 的进程，然后 `$!`取得其进程号，并将进程号传入 `wait`，得知进程号为 5798，并等待 5s 直到其完成。

思考：系统如何管理系统中的多个进程？进程的家族关系是怎样体现的？有什么用？

答：Linux 系统中通过进程控制块 PCB 来控制和管理进程。

进程的家族关系可以用进程家族树展示，可以根据进程之间的父子关系查找到具体的进程。

讨论

1. Linux 系统命令很多，在手头资料不全时，如何查看命令格式？

答：使用 man、info、help 命令和—help 参数显示帮助。

man 命令用来访问存储在 Linux 系统上的手册页面。在想要查找的工具的名称前面输入 man 命令，就可以找到那个工具相应的手册条目。

例如输入 man ls，查找 ls 命令的手册。

```
LS(1)                                User Commands                                LS(1)

NAME
  ls - list directory contents

SYNOPSIS
  ls [OPTION]... [FILE]...

DESCRIPTION
  List information about the FILES (the current directory by default).
  Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
  fied.

  Mandatory arguments to long options are mandatory for short options
  too.

  -a, --all
      do not ignore entries starting with .

  -A, --almost-all
      do not list implied . and ..

  --author

Manual page ls(1) line 1 (press h for help or q to quit)
```

2. Linux 系统用什么方式管理多个用户操作？如何管理用户文件，隔离用户空间？用命令及结果举例说明。

答：Linux 系统是一个多用户多任务的分时操作系统，任何一个要使用系统资源的用户，都必须首先向系统管理员申请一个账号，然后以这个账号的身份进入系统。用户的账号一方面可以帮助系统管理员对使用系统的用户进行跟踪，并控制他们对系统资源的访问；另一方面也可以帮助用户组织文件，并为用户提供安全性保护。我们对用户和用户组进行添加、修改、删除最终目的，是修改系统用户/etc/passwd 和用户组的/etc/groups。

Linux 系统通过“存取控制模式”对不同用户分配文件的不同的操作权。Linux 文件系统将用户分成三类，即文件主、同组人、其他人。每种人可以行使的操作有三种，即读(r)、写(w)、执行(x)。这样就做到了管理用户文件。

```
yanxinyu@Thinkbook16-2022:~$ ls -l mytext.txt
-rwxr-x--x 1 root yanxinyu 13 10月 26 22:36 mytext.txt
```

Linux 提供一种内核级别环境隔离的方法来隔离用户空间。

3. 用什么方式查看你的进程的管理参数？这些参数怎样体现父子关系？当结束一个父进程后其子进程如何处理？用命令及结果举例说明。

答：当程序运行在系统上时，我们称之为进程(process)。想监测这些进程，可使用 ps 命令查看进程。ps 命令能输出运行在系统上的所有程序的许多信息。

树形可表示程序间的相互关系类似父子关系。使用 ps --forest 命令会显示进程的层级信息，这种格式让跟踪子进程和父进程变得十分容易。

```
yanxinyu@Thinkbook16-2022:~$ ps --forest
```

PID	TTY	TIME	CMD
3270	pts/0	00:00:00	bash
5652	pts/0	00:00:00	_ ps

ps 命令虽然在收集运行在系统上的进程信息时非常有用，但也有不足之处：它只能显示某个特定时间点的信息。如果想观察那些频繁换进换出的内存的进程趋势，用 ps 命令就不方便了。而 top 命令刚好适用这种情况。top 命令跟 ps 命令相似，能够显示进程信息，但它是实时显示的，如下图所示：

```
top - 21:03:09 up 6 min, 1 user, load average: 2.04, 1.08, 0.53
任务: 309 total, 1 running, 308 sleeping, 0 stopped, 0 zombie
%Cpu(s): 9.1 us, 11.4 sy, 0.0 ni, 78.7 id, 0.2 wa, 0.0 hi, 0.7 si, 0.0 st
MiB Mem : 3889.8 total, 428.0 free, 1743.7 used, 1718.1 buff/cache
MiB Swap: 2140.0 total, 2140.0 free, 0.0 used, 1783.0 avail Mem
```

进程号	USER	PR	NI	VIRT	RES	SHR	%CPU	%MEM	TIME+	COMMAND	
3616	yanxinyu	20	0	3493416	436640	220516	S	12.2	11.0	0:18.86	firefox
4932	_apt	20	0	34820	11704	6960	S	8.1	0.3	0:00.64	store
1925	yanxinyu	20	0	4519324	292696	163140	S	5.7	7.3	0:16.98	gnome-shell
4943	yanxinyu	20	0	2398880	62728	50692	S	3.0	1.6	0:00.09	Web Content
3178	yanxinyu	20	0	739424	77912	61372	S	2.4	2.0	0:02.86	gnome-terminal-
4653	_apt	20	0	33212	9732	8740	S	2.0	0.2	0:00.15	http
4569	yanxinyu	20	0	2422308	78140	64064	S	1.4	2.0	0:00.41	Web Content
4111	yanxinyu	20	0	2520292	116624	84688	S	1.0	2.9	0:01.22	Privileged Cont
4652	_apt	20	0	33208	9544	8552	S	1.0	0.2	0:00.12	http
1304	root	20	0	511804	84204	66856	S	0.7	2.1	0:00.82	packagekitd
3780	yanxinyu	20	0	1586908	186164	71004	S	0.7	4.7	0:10.42	snap-store
9	root	0	-20	0	0	0	I	0.3	0.0	0:00.43	kworker/0:1H-kblockd
136	root	0	-20	0	0	0	I	0.3	0.0	0:00.31	kworker/1:1H-kblockd
291	root	20	0	0	0	0	I	0.3	0.0	0:00.87	kworker/u256:29-writeback
385	root	20	0	0	0	0	I	0.3	0.0	0:00.18	kworker/1:3-events
866	message+	20	0	11212	6880	4096	S	0.3	0.2	0:01.34	dbus-daemon
1152	rtkit	21	1	154000	1560	1336	S	0.3	0.0	0:00.03	rtkit-daemon
2156	yanxinyu	20	0	297636	38972	30304	S	0.3	1.0	0:00.78	vmtoolsd
2443	yanxinyu	20	0	2877388	91964	74572	S	0.3	2.3	0:01.42	gjs

Linux 系统保证每个进程都有一个父进程，若父进程比子进程先终止，则该父进程的所有子进程的父进程都变为 init 进程。其执行顺序大致如下：在一个进程终止时，内核逐个检查所有活动进程，以判断它是否是正要终止的进程的子进程，如果是，则该进程的父进程 ID 就更改为 1(init 进程的 ID))。

4. Linux 系统“文件”的含义是什么？它的文件有几种类型？如何标识的？

答：在 Linux 系统中，一切都是文件。所有的硬件设备也都被系统看作是文件。Linux 中共有七种文件类型，即普通文件(-)、目录文件(d)、字符设备文件(c)、块设备文件(b)、套接字文件(s)、符号链接文件(l)和管道文件(p)。

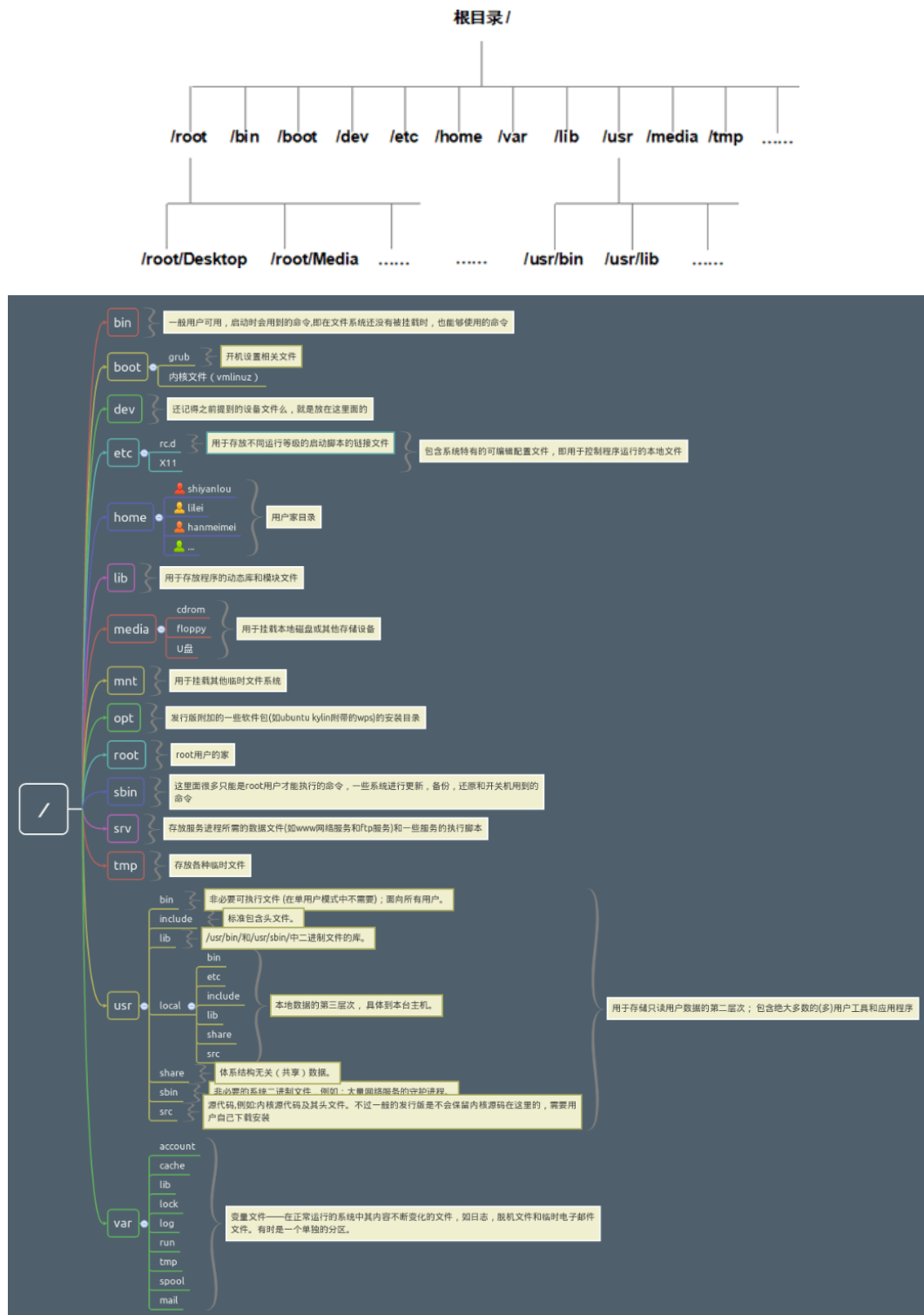
5. Linux 系统的可执行命令主要放在什么地方？找出你的计算机中所有存放系统的可执行命令的目录位置。

答： /etc/rc.d/init.d/ 可执行脚本
 /usr/bin 二进制可执行文件

6. Linux 系统的设备是如何管理的？在什么地方可以找到描述设备的信息？

答：Linux 系统通过设备文件来管理设备。每个设备对应着一个设备文件。设备文件主要包括可供系统识别的设备号、设备权限和设备类型等信息。Linux 把所有的设备文件都置于/dev 目录下。使用 lshw 命令可以找到设备的信息。

7. 画出 Linux 根文件系统的框架结构。描述各目录的主要作用。你的用户主目录在哪里？
答：



- **/home**：普通用户主目录，当新建账户时，都会分配在此。
- **/bin**：提供用户使用的基本命令，存放二进制命令，不允许关联到独立分区，OS启动会用到里面的程序。
- **/lib**：存放系统在启动时依赖的基本共享库文件以及内核模块文件。系统使用的函数库的目录也存放了大量的脚本库文件，程序在执行过程中，需要调用时会用到。

- **/proc:** 用于输出内核与进程信息相关的虚拟文件系统，目录中的数据都在内存中，如系统核心，外部设备，网络状态，由于数据都存放于内存中，所以不占用磁盘空间。
- **/tmp:** 系统的临时文件，一般系统重启不会被保存，任何人都可以访问,重要数据一定不要放在此目录下。
- **/usr:** 应用程序存放目录。

8. Linux 系统的 Shell 是什么？请查找这方面的资料，说明不同版本的 Shell 的特点。

答: Shell 既是一种脚本编程语言，也是一个连接内核和用户的软件。在 Linux 发展的早期，唯一能用的工具就是 Shell，Linux 用户都是在 Shell 中输入文本命令，并查看文本输出；如果有必要的话，Shell 也能显示一些基本的图形。

常见的 Shell 有 sh、bash、csh、tcsh、ash、dash、zsh 等。

- **Bourne shell(sh)**

UNIX 最初使用，且在每种 UNIX 上都可以使用。在 shell 编程方面相当优秀，但在处理与用户的交互方面做得不如其他几种 shell。sh 的全称是 Bourne shell，由 AT&T 公司的 Steve Bourne 开发，为了纪念他，就用他的名字命名了。

- **C shell(csh)**

csh，一个语法上接近于 C 语言的 shell。sh 之后另一个广为流传的 shell 是由柏克莱大学的 Bill Joy 设计的，这个 shell 的语法有点类似 C 语言，所以才得名为 Cshell，简称为 csh。

- **Korn shell(ksh)**

完全向上兼容 Bourne shell，并包含了 C shell 的很多特性。

- **Bourne Again shell(bash)**

因为 Linux 操作系统缺省的 shell。即 bash 是 Bourne shell 的扩展，与 Bourne shell 完全向后兼容。在 Bourne shell 的基础上增加、增强了很多特性。可以提供如命令补全、命令编辑和命令历史表等功能。包含了很多 C shell 和 Korn shell 中的优点，有灵活和强大的编程接口，同时又有很友好的用户界面。

- **Debian Almquist shell(dash)**

- 原来 bash 是 GNU/Linux 操作系统中的/bin/sh 的符号连接，但由于 bash 过于复杂，有人把 bash 从 NetBSD 移植到 Linux 并更名为 dash，且/bin/sh 符号连接到 dash。Dash Shell 比 Bash Shell 小的多（ubuntu16.04 上，bash 大概 1M，dash 只有 150K），符合 POSIX 标准。Ubuntu6.10 开始默认是 Dash。

- **zsh**

zsh 配置复杂，所以很多人都不会使用。直到有一天出了 Oh My Zsh 项目，才让更多人发现并开始使用 zsh。有人评价 zsh 是终极 shell，我也这样认为！

9. 下面每一项说明的是哪类文件。

- (1)-rwxrw-r-- (2) /bin (3) ttyx3 (4) brw-rw-rw- (5)/etc/passwd
(6) crw-rw-rw- (7) /usr/lib (8) Linux

答: (1) 一般文件 (2) 二进制执行文件目录 (3) 终端设备 (4) 块文件 (5) 用户数据库，包含用户名清单等信息 (6) 字符文件 (7) 库文件 (8) 系统文件

实验体会

大多数计算机用户只是熟悉图形用户界面(GUI)，并且认为命令行界面(CLI)是一种很古老的，很难使用的东西。其实不然，一个好的命令行界面可是让我们更加充分，高效的利用计算机。正如人们所说，“图形用户界面让简单的任务更容易完成，而命令行界面使完成复杂的任务成为可能。”

通过本次实验，平时习惯于使用图形界面操作系统的我熟悉了 Linux 中的终端命令，学习了一些基本的文件操作命令，进程查看命令等。相比于图形界面的直观，命令行固然显得不那么生动，但是命令的使用不光应是我们每个计算机专业学生的基本素养，它更有助于从底层来帮助理解操作系统内部的运作方法以及文件组织、进程管理策略。

其实通过实验的尝试，我明白学好 Linux 不是一件一蹴而就的事，一定要能坚持使用它，特别是在学习初期，要增加自己的 Linux 技能，只有通过实践来实现。遇到问题时要自己去寻找答案，在寻找答案的过程中，会学到更多知识。

而且，应该说目前我们对于 Linux 的学习只是入门而已，学习 Linux 是一个长期的过程，很耗时间。作为计算机科学与技术专业的学生，对我们来说，学习 Linux 将主要应用于对服务器的操作和管理，这非一日之功，我们还需要不断地深入学习它。