

河南工业大学 Linux 基础与应用 实验报告

专业班级: 软件 1305 班 学号: 201316920311 姓名: 田劲锋 指导老师: 赵晨阳 评分: _____

实验题目: Linux 常用命令 (二)

实验目的: (1) 理解文件处理命令, 包括 `grep` 命令、`head` 命令等; (2) 理解压缩解压命令, 包括 `bzip2` 命令和 `bunzip2` 命令、`gzip` 命令、`unzip` 命令、`zcat` 命令和 `tar` 命令; (3) 理解磁盘操作命令, 包括 `mount` 命令、`umount` 命令、`df` 命令、`du` 命令; (4) 理解变换用户身份命令, 包括 `su` 命令、`sudo` 命令、`useradd` 命令、`passwd` 命令等; (5) 理解关机重启命令, 包括 `shutdown` 命令、`reboot` 命令等; (7) 理解网络操作命令; (8) 理解 `more` 命令、`less` 命令等其他命令。

实验内容:

- (1) 新建一个文件, 并写入三行内容, 分别为 “hello world”、“hello everyone” 和 “end”, 然后试执行 `grep` 命令查找一个字符串 “hello”。
- (2) 使用命令查看 `/etc` 下的目录的详细信息, 并要求只显示前 5 行。
- (3) 统计文件 `file.txt` 的信息, 要求显示该文件的行数、单词数和字符数。
- (4) 试想可以通过哪些命令查询 `pwd` 命令的相关信息, 比如查询 `pwd` 命令的绝对路径以及相关关联的文件名。
- (5) 在一个目录下新建两个文件 `1.txt` 和 `2.txt`, 试使用命令将这两个文件打包压缩成 `3.tar.gz` 和 `4.tar.bz2`, 然后删除原文件, 解压缩文件 `3.tar.gz`。
- (6) 使用命令的形式写出挂载和卸载 U 盘的步骤。
- (7) 查看当前磁盘的分区信息以及查看当前目录下所有文件及目录的信息。
- (8) 试写出几个关机的命令, 并尝试一个。
- (9) 实现用户 `ubuntu` 和用户 `root` 的转换。
- (10) 新建一个用户 `ubuntu1`, 并设置密码。
- (11) 查看机子的网络配置, 测试网络是否畅通。
- (12) 使用 `more` 和 `less` 命令查看一个文件。
- (13) 用 `ubuntu` 用户登录 Linux, 在宿主目录下创建目录 `perm`, 在该目录下创建文件 `newfile`, 授予所有用户对 `perm` 目录都有 `rw` 权限; 切换到普通用户 `ubuntu1`, 执行 “`rm /home/ubuntu/perm/newfile`” 是否可以执行。
- (14) 用 `ubuntu` 用户登录 Linux, 在宿主目录下创建文件 `newfile2`, 移动文件 `newfile2` 到 `perm` 目录下同时改名为 `file01`, 然后, 改变 `file01` 的文件权限为 “`rw-rw-r-`”。

实验步骤:

本次实验项目较多, 不再一一详细叙述, 直接给出所执行的命令和截图。

- (1) 非常强大的正则表达式, 这里并没有发挥其强大能力。

```
$ cat > hello.txt
$ grep 'hello' hello.txt
```

- (2) 使用了管道。

```
$ ls -lha /etc | head -5
```

- (3) 统计代码行数的时候很有用。

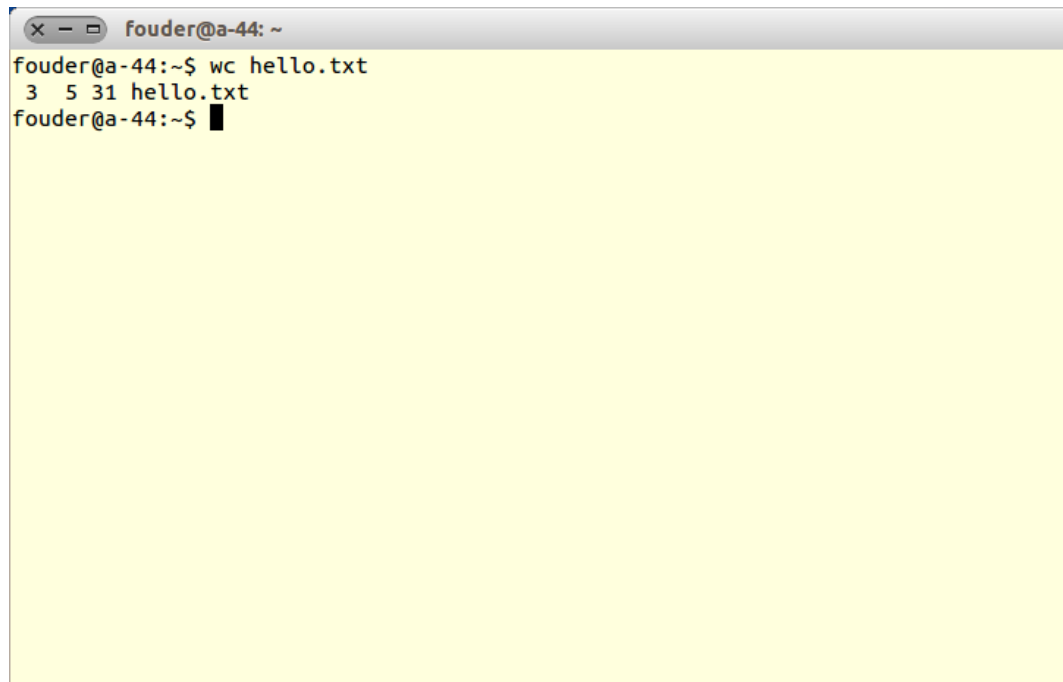
```
fouder@a-44: ~  
fouder@a-44:~$ cat > hello.txt  
hello world  
hello everyone  
end  
fouder@a-44:~$ grep 'hello' hello.txt  
hello world  
hello everyone  
fouder@a-44:~$
```

图 1: grep

```
fouder@a-44: ~  
fouder@a-44:~$ ls -lha /etc | head -5  
总用量 1.2M  
drwxr-xr-x 133 root root 12K 5月 4 13:01 .  
drwxr-xr-x 23 root root 4.0K 11月 4 18:04 ..  
drwxr-xr-x 3 root root 4.0K 4月 25 2013 acpi  
-rw-r--r-- 1 root root 3.0K 4月 25 2013 adduser.conf  
fouder@a-44:~$
```

图 2: head

```
$ wc hello.txt
```

A screenshot of a terminal window with a yellow background. The window title is 'fouder@a-44: ~'. The terminal shows the command 'wc hello.txt' being executed, followed by the output '3 5 31 hello.txt'. The prompt 'fouder@a-44:~\$' is visible at the end of the line.

```
fouder@a-44:~$ wc hello.txt
3 5 31 hello.txt
fouder@a-44:~$
```

图 3: wc

- (4)

```
$ whereis pwd
```



```
$ which pwd
```

- (5) 简单的压缩解压（打包解包）。

```
$ tar -czf 3.tar.gz 1.txt
$ tar -cjf 4.tar.bz2 2.txt
$ tar -xzvf 3.tar.gz
$ tar -xjvf 4.tar.bz2
```

- (6) 这里我们把U盘挂载到 /mnt/upan 里。

```
$ sudo fdisk -l
$ sudo mkdir -p /mnt/upan
$ sudo mount /dev/sdb1 /mnt/upan
$ sudo umount /mnt/upan
```

- (7)

```
$ df
```



```
$ du
```

- (8) 几个关机和重启命令不再演示，只是列出其帮助信息。最后一张是真正关机。

```
$ shutdown
$ halt
```

```
fouder@a-44: ~  
fouder@a-44:~$ whereis pwd  
pwd: /bin/pwd /usr/include/pwd.h /usr/share/man/man1/pwd.1.gz  
fouder@a-44:~$ which pwd  
/bin/pwd  
fouder@a-44:~$ █
```

图 4: which

```
fouder@a-44: ~  
fouder@a-44:~$ cat > 1.txt  
11111  
fouder@a-44:~$ cat > 2.txt  
2222222  
fouder@a-44:~$ tar -czf 3.tar.gz 1.txt  
fouder@a-44:~$ tar -cjf 4.tar.bz2 2.txt  
fouder@a-44:~$ ls  
1.txt 3.tar.gz haut 模板 图片 下载 桌面  
2.txt 4.tar.bz2 公共的 视频 文档 音乐  
fouder@a-44:~$ rm *.txt  
fouder@a-44:~$ ls  
3.tar.gz 4.tar.bz2 haut 公共的 模板 视频 图片 文档 下载 音乐 桌面  
fouder@a-44:~$ tar -xzf 3.tar.gz  
1.txt  
fouder@a-44:~$ tar -xjvf 4.tar.bz2  
2.txt  
fouder@a-44:~$ ls  
1.txt 3.tar.gz haut 模板 图片 下载 桌面  
2.txt 4.tar.bz2 公共的 视频 文档 音乐  
fouder@a-44:~$ cat 1.txt 2.txt  
11111  
2222222  
fouder@a-44:~$ █
```

图 5: tar

```
fouder@a-44:~$ sudo fdisk -l
```

Disk /dev/sda: 317.8 GB, 317824786944 bytes
255 heads, 63 sectors/track, 38639 cylinders, total 620751537 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x12340001

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	493131240	554563799	30716280	83	Linux
/dev/sda2		554692320	564845399	5076540	82	Linux swap / Solaris
/dev/sda3		564973920	605843279	20434680	7	HPFS/NTFS/exFAT

Disk /dev/sdb: 7756 MB, 7756087296 bytes
245 heads, 6 sectors/track, 10305 cylinders, total 15148608 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1	*	63	15148607	7574272+	7	HPFS/NTFS/exFAT

```
fouder@a-44:~$
```

```
$ reboot
$ init
```

- (9) 切换之前要为根用户设置密码，来激活根用户。

```
$ sudo passwd root
$ su
# exit
$
```

- (10)

```
$ sudo adduser ubuntu1
$ su ubuntu1
```

- (11)

```
$ ifconfig
$ ping weibo.com
$ ping 172.168.69.181 -c 4
```

- (12) 这里我把前两次实验的 \LaTeX 源文件显示了出来。

```
$ more
$ less
```

- (13) 注意用户间的切换。所有权限为777。

```
ubuntu:~$ mkdir perm
ubuntu:~$ touch perm/newfile
ubuntu:~$ chmod 777 perm
ubuntu:~$ su ubuntu1
ubuntu1:/home/ubuntu$ rm perm/newfile
ubuntu1:/home/ubuntu$ exit
ubuntu:~$
```

- (14) “rwxrw-r-” 权限表示为数字是764。

```
$ touch newfile2
$ mv newfile2 perm/file01
$ chmod 764 perm/file01
```

实验体会：

依旧是基本命令的用法，这些命令强大且常用。`grep`强大但是并不容易学会，`tar`比较常用于备份，`mount`用来挂载分区和硬盘，`su`也是多用户操作常用的，`shutdown`等关机命令其实并不常用，尤其是在服务器上，`ifconfig`还可以用来配置网络，`less`要比`more`人性化那么一点。

```
fouder@a-44:~$ shutdown --help
用法: shutdown [OPTION]... 时间 [信息]
使系统关闭。

Options:
  -r                reboot after shutdown
  -h                halt or power off after shutdown
  -H                halt after shutdown (implies -h)
  -P                power off after shutdown (implies -h)
  -c                cancel a running shutdown
  -k                only send warnings, don't shutdown
  -q, --quiet       reduce output to errors only
  -v, --verbose     increase output to include informational messages
  --help            display this help and exit
  --version         output version information and exit

时间 可以使用不同的格式。最常用的是简单的一个单词
"now", 其使系统立即关闭。 其它可用的格式有 +m, 此 m
是关机前等待的分钟数; hh:mm 其指定以 24 小时制中的时间。

已登录的用户被一条发送到他们终端的一条消息警告, 您可
以包含一条可选的 消息 到此项中。 使用 -k
选项可以发送警告而不真的关机。

如果给出 时间, 此命令将留在前端指导关机发生。
可以使用 Control-C 取消它, 或者被其他用户以 -c 选项取消。

系统默认进入维护状态(单用户)模式, 你可以使用 -r 或 -h
更改此行为, 其分别指定系统重启或关闭。-h
选项可以进一步由 -H 或 -P
来指定何时是关机还是切断电源。 默认动作由 shutdown
脚本决定。

Report bugs to <upstart-devel@lists.ubuntu.com>
fouder@a-44:~$
```

(a)

```
fouder@a-44:~$ halt --help
用法: halt [OPTION]...
关闭系统

Options:
  -n, --no-sync      don't sync before reboot or halt
  -f, --force         force reboot or halt, don't call shutdown(8)
  -p, --poweroff      switch off the power when called as halt
  -w, --wtmp-only     don't actually reboot or halt, just write wtmp
                      record
  -q, --quiet         reduce output to errors only
  -v, --verbose       increase output to include informational messages
  --help             display this help and exit
  --version           output version information and exit

此命令将指示内核重启或关闭系统; 在没有 -f
选项运行时, 或当处于系统运行级别不是 0 或 6
时, 其将真的执行 /sbin/shutdown。

Report bugs to <upstart-devel@lists.ubuntu.com>
fouder@a-44:~$
```

(b)

```
fouder@a-44:~$ reboot --help
用法: reboot [OPTION]...
重新启动

Options:
  -n, --no-sync      don't sync before reboot or halt
  -f, --force         force reboot or halt, don't call shutdown(8)
  -p, --poweroff      switch off the power when called as halt
  -w, --wtmp-only     don't actually reboot or halt, just write wtmp
                      record
  -q, --quiet         reduce output to errors only
  -v, --verbose       increase output to include informational messages
  --help             display this help and exit
  --version           output version information and exit

此命令将指示内核重启或关闭系统; 在没有 -f
选项运行时, 或当处于系统运行级别不是 0 或 6
时, 其将真的执行 /sbin/shutdown。

Report bugs to <upstart-devel@lists.ubuntu.com>
fouder@a-44:~$
```

(c)

```
fouder@a-44:~$ init --help
用法: init [OPTION]...
管理守护进程

Options:
  --confdir=DIR      specify alternative directory to load
                     configuration files from
  --default-console=VALUE
                     default value for console stanza
  --no-inherit-env    jobs will not inherit environment of init
  --logdir=DIR        specify alternative directory to store job output
                     logs in
  --no-log            disable job logging
  --no-sessions        disable chroot sessions
  --no-startup-event  do not emit any startup event (for testing)
  --restart            flag a re-exec has occurred
  --state-fd=FD       specify file descriptor to read serialisation
                     data from
  --session           use D-Bus session bus rather than system bus (for
                     testing)
  --startup-event=NAME
                     specify an alternative initial event (for
                     testing)
```

(d)

```
fouder@a-44:~$ sudo shutdown -h 10
sudo: 无法解析主机: a-44

来自fouder@a-44的广播信息
(/dev/pts/0) 于 14:04 ...

The system is going down for halt in 10 minutes!
```

(e)

图 8: shutdown

```
fouder@a-44: ~
fouder@a-44:~$ sudo passwd root
输入新的 UNIX 密码:
重新输入新的 UNIX 密码:
passwd: 已成功更新密码
fouder@a-44:~$ su
密码:
root@a-44:/home/fouder# exit
exit
fouder@a-44:~$ █
```

图 9: su

```
fouder@a-44: ~
fouder@a-44:~$ sudo adduser ubuntu1
sudo: 无法解析主机: a-44
正在添加用户"ubuntu1"...
正在添加新组"ubuntu1" (1001)...
正在添加新用户"ubuntu1" (1001) 到组"ubuntu1"...
创建主目录"/home/ubuntu1"...
正在从"/etc/skel"复制文件...
输入新的 UNIX 密码:
重新输入新的 UNIX 密码:
passwd: 已成功更新密码
正在改变 ubuntu1 的用户信息
请输入新值, 或直接敲回车键以使用默认值
  全名 []:
  房间号码 []:
  工作电话 []:
  家庭电话 []:
  其它 []:
这些信息是否正确? [Y/n] y
fouder@a-44:~$ su ubuntu1
密码:
ubuntu1@a-44:/home/fouder$ who
fouder    tty7          2015-05-04 21:01 (:0)
fouder    pts/0           2015-05-04 13:02 (:0)
fouder    pts/1           2015-05-04 13:11 (:0)
ubuntu1@a-44:/home/fouder$ who am i
fouder    pts/0           2015-05-04 13:02 (:0)
ubuntu1@a-44:/home/fouder$ exit
exit
fouder@a-44:~$ █
```

图 10: adduser


```
fouder@a-44:~$ ifconfig
eth1      Link encap:以太网  硬件地址 00:25:11:88:0e:9e
          inet 地址:172.18.69.181  广播:172.18.69.255  掩码:255.255.255.0
          inet6 地址: fe80::225:11ff:fe88:e9e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
          接收数据包:5056 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:2882 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:1489614 (1.4 MB)  发送字节:366694 (366.6 KB)

lo        Link encap:本地环回
          inet 地址:127.0.0.1  掩码:255.0.0.0
          inet6 地址: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  跃点数:1
          接收数据包:749 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:749 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:0
          接收字节:78205 (78.2 KB)  发送字节:78205 (78.2 KB)

fouder@a-44:~$ █
```

(a)

```
fouder@a-44:~$ ping weibo.com
PING weibo.com (121.194.0.221) 56(84) bytes of data.
64 bytes from 121.194.0.221: icmp_req=1 ttl=53 time=12.6 ms
64 bytes from 121.194.0.221: icmp_req=2 ttl=53 time=11.3 ms
64 bytes from 121.194.0.221: icmp_req=3 ttl=53 time=11.4 ms
64 bytes from 121.194.0.221: icmp_req=4 ttl=53 time=11.6 ms
64 bytes from 121.194.0.221: icmp_req=5 ttl=53 time=12.0 ms
^C64 bytes from 121.194.0.221: icmp_req=6 ttl=53 time=11.9 ms

--- weibo.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 25123ms
rtt min/avg/max/mdev = 11.346/11.844/12.649/0.440 ms
fouder@a-44:~$ ping 172.18.69.181 -c 4
PING 172.18.69.181 (172.18.69.181) 56(84) bytes of data.
64 bytes from 172.18.69.181: icmp_req=1 ttl=64 time=0.040 ms
64 bytes from 172.18.69.181: icmp_req=2 ttl=64 time=0.027 ms
64 bytes from 172.18.69.181: icmp_req=3 ttl=64 time=0.027 ms
64 bytes from 172.18.69.181: icmp_req=4 ttl=64 time=0.029 ms

--- 172.18.69.181 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2998ms
rtt min/avg/max/mdev = 0.027/0.030/0.040/0.008 ms
fouder@a-44:~$ █
```

(b)

图 11: ifconfig & ping

```
fouder@a-44:~$ more haut/linux/linuxexp/exp2.tex
```

(a)

```
\documentclass[cs4size,a4paper,nofonts]{ctexart}
\usepackage[utf8]{inputenc}
\def\ttf{{\tt{田劲锋}}}
\def\ttitle{Linux常用命令 (一)}
\usepackage[a4paper,margin=2.2cm]{geometry} % 页面设置
\usepackage[unicode,breaklinks=true,
colorlinks=true,linkcolor=black,anchorcolor=black,citecolor=black,urlcolor=black]
{
pdftitle={\title},pdfauthor={\ttf}}{\hyperref}
%\CTEXsetup[number=\chinese{section}, format={\large\sf\bfseries}]{section}
\usepackage{latexsyn,amsmath,amssymb,bm}
\usepackage{graphicx}
\usepackage{subfigure}
\usepackage{wrapfig}
\usepackage{fancyvrb}
\DefineShortVerb{\|}
\Fvset{frame=single}

\setmainfont{Times New Roman}
\setCJKmainfont[BoldFont={SimHei}]{SimSun} % 主要字体: 宋体、黑体
\setCJKsansfont[BoldFont={STZhongsong}]{STFangsong} % 次要字体: 仿宋、中宋
\setCJKmonofont{KFKai} % 等宽字体: 楷体

--More--(10%)
```

(b)

```
\end{Verbatim}
\end{enumerate}
实质上只是一个通配符~|*|~的区别而已。注意上面的~|newdir2/newdir1|~目录是上一次复制的遗留产物。

\begin{figure}[htp]
%\subimage{2015-04-27-(7)-1}{复制目录}{1}
\subimage{2015-04-27-(7)-2}{复制目录}{2}
\subimage{2015-04-27-(7)-3}{复制目录}{3}
\caption{复制目录}\label{fig:复制目录}
\end{figure}

\end{enumerate}

\paragraph{实验体会: }\quad

这次实验是练习Shell中非常基本的几个文件和目录操作命令。这些命令在日常应用中都是很常用的。

最有趣的其实是~|cat|~, 明明只是简单把输入的内容输出出来, 却由于重定向而变得非常强大。我最喜欢用~|cat|~来复制粘贴文本了。

\end{document}
fouder@a-44:~$
```

(c)

```
fouder@a-44:~$ less haut/linux/linuxexp/exp1.tex
```

(d)

```
\documentclass[cs4size,a4paper,nofonts]{ctexart}
\usepackage[utf8]{inputenc}
\def\ttf{{\tt{田劲锋}}}
\def\ttitle{虚拟机上安装Linux系统}
\usepackage[a4paper,margin=2.2cm]{geometry} % 页面设置
\usepackage[unicode,breaklinks=true,
colorlinks=true,linkcolor=black,anchorcolor=black,citecolor=black,urlcolor=black]
{
pdftitle={\title},pdfauthor={\ttf}}{\hyperref}
%\CTEXsetup[number=\chinese{section}, format={\large\sf\bfseries}]{section}
\usepackage{latexsyn,amsmath,amssymb,bm}
\usepackage{graphicx}

\setmainfont{Times New Roman}
\setCJKmainfont[BoldFont={SimHei}]{SimSun} % 主要字体: 宋体、黑体
\setCJKsansfont[BoldFont={STZhongsong}]{STFangsong} % 次要字体: 仿宋、中宋
\setCJKmonofont{KFKai} % 等宽字体: 楷体

\CJKsetecglue{\hspace{0.1em}}
\renewcommand{\CJKglue}{\hspace{-0.3pt plus 0.08\baselineskip}}
\Frenchspacing
\widowpenalty=10000
\linespread{1.2} % 行距
haut/linux/linuxexp/exp1.tex
```

(e)

```
\item 最后我们关闭计算机, 如图~\ref{fig:关机}。

\begin{figure}[htp]
\image{Ubuntu-2015-04-13-21-21-16}{系统进程占用}
\image{Ubuntu-2015-04-13-21-21-53}{关机}
\end{figure}

\end{enumerate}

{\bf 实验体会: }(200字以内)

Linux 桌面操作系统, 尤其是由 Ubuntu 领导的 Debian 系发行版的流行, 近年来为 Linux 争取到了大量用户。
% 然而自从 Unity 以及 Gnome 3 的出现, Ubuntu 的图形界面变得越来越慢且难于使用。Ubuntu Kylin 的发行更是以一种不伦不类的形式进入了中国市场。反倒 Linux Deepin 在图形界面上狠下功夫, 做得越来越好了。
% 不过由于 X Window 的本质设计问题, Linux 的 GUI 一直都不怎样, 加上应用软件开发商不愿意为 Linux 开发优质的商业软件 (如 QQ), Linux 并不适合桌面应用。但是 Linux 的强大命令行是毋庸置疑的。
这次安装 Ubuntu 的最新版花了不少时间, 一个是安装包过大下载了很久, 一个是安装启动过程有些缓慢和卡顿。虽然缺点仍然不少, 但也不能忽视这几年 Ubuntu 的进步就是了。

\end{document}
(END)
```

(f)

图 12: more & less

```
fouder@a-44: ~  
fouder@a-44:~$ ls  
haut 公共的 模板 视频 图片 文档 下载 音乐 桌面  
fouder@a-44:~$ mkdir perm  
fouder@a-44:~$ ls -l  
总用量 40  
drwxrwxr-x 5 fouder fouder 4096 5月 4 13:02 haut  
drwxrwxr-x 2 fouder fouder 4096 5月 4 13:37 perm  
drwxr-xr-x 2 fouder fouder 4096 11月 4 18:08 公共的  
drwxr-xr-x 2 fouder fouder 4096 11月 4 18:08 模板  
drwxr-xr-x 2 fouder fouder 4096 11月 4 18:08 视频  
drwxr-xr-x 2 fouder fouder 4096 5月 4 13:34 图片  
drwxr-xr-x 2 fouder fouder 4096 11月 4 18:08 文档  
drwxr-xr-x 2 fouder fouder 4096 11月 4 18:08 下载  
drwxr-xr-x 2 fouder fouder 4096 11月 4 18:08 音乐  
drwxr-xr-x 2 fouder fouder 4096 5月 4 13:04 桌面  
fouder@a-44:~$ touch perm/newfile  
fouder@a-44:~$ find perm  
perm  
perm/newfile  
fouder@a-44:~$ █
```

(a)

```
fouder@a-44:~$ chmod 777 perm  
fouder@a-44:~$ ls -l  
总用量 40  
drwxrwxr-x 5 fouder fouder 4096 5月 4 13:02 haut  
drwxrwxrwx 2 fouder fouder 4096 5月 4 13:38 perm  
drwxr-xr-x 2 fouder fouder 4096 11月 4 18:08 公共的  
drwxr-xr-x 2 fouder fouder 4096 11月 4 18:08 模板  
drwxr-xr-x 2 fouder fouder 4096 11月 4 18:08 视频  
drwxr-xr-x 2 fouder fouder 4096 5月 4 13:38 图片  
drwxr-xr-x 2 fouder fouder 4096 11月 4 18:08 文档  
drwxr-xr-x 2 fouder fouder 4096 11月 4 18:08 下载  
drwxr-xr-x 2 fouder fouder 4096 11月 4 18:08 音乐  
drwxr-xr-x 2 fouder fouder 4096 5月 4 13:04 桌面  
fouder@a-44:~$ su ubuntu1  
密码:  
ubuntu1@a-44:/home/fouder$ rm perm/newfile  
rm: 是否删除有写保护的普通空文件 "perm/newfile"? y  
ubuntu1@a-44:/home/fouder$ find perm  
perm  
ubuntu1@a-44:/home/fouder$ exit  
fouder@a-44:~$ find perm  
perm  
fouder@a-44:~$ █
```

(b)

图 13: chmod & su

```
fouder@a-44: ~  
fouder@a-44:~$ ls -l  
总用量 40  
drwxrwxr-x 5 fouder fouder 4096 5月 4 13:02 haut  
drwxrwxrwx 2 fouder fouder 4096 5月 4 13:41 perm  
drwxr-xr-x 2 fouder fouder 4096 11月 4 18:08 公共的  
drwxr-xr-x 2 fouder fouder 4096 11月 4 18:08 模板  
drwxr-xr-x 2 fouder fouder 4096 11月 4 18:08 视频  
drwxr-xr-x 2 fouder fouder 4096 5月 4 13:39 图片  
drwxr-xr-x 2 fouder fouder 4096 11月 4 18:08 文档  
drwxr-xr-x 2 fouder fouder 4096 11月 4 18:08 下载  
drwxr-xr-x 2 fouder fouder 4096 11月 4 18:08 音乐  
drwxr-xr-x 2 fouder fouder 4096 5月 4 13:04 桌面  
fouder@a-44:~$ touch newfile2  
fouder@a-44:~$ ls  
haut newfile2 perm 公共的 模板 视频 图片 文档 下载 音乐 桌面  
fouder@a-44:~$ mv newfile2 perm/file01  
fouder@a-44:~$ ls -l perm/  
总用量 0  
-rw-rw-r-- 1 fouder fouder 0 5月 4 13:41 file01  
fouder@a-44:~$ chmod 764 perm/file01  
fouder@a-44:~$ ls -l perm/  
总用量 0  
-rwxrw-r-- 1 fouder fouder 0 5月 4 13:41 file01  
fouder@a-44:~$ █
```

图 14: chmod