

电子科技大学
计算机科学与工程学院

标准实验报告

(实验) 课程名称 Unix 操作系统

电子科技大学教务处制表

电子科技大学

电子科技大学

实验报告

学生姓名：刘芷溢

学号：2020080907009

指导教师：聂晓文

实验地点：主楼 A2-412

实验时间：2022/5/6

一、实验室名称：计算机学院实验中心

二、实验项目名称：UNIX 文件系统相关操作

三、实验学时：4 学时

四、实验原理：Unix 操作系统和文件基本操作以及 shell 基本原理

五、实验目的：

理解 Unix 文件操作，掌握基本的 shell 命令；使用文件系统模型理解文件操作，实现书上的各类命令。

六、实验内容：

1. 文件系统基本操作，并用文件系统模型解释：cd、cp、ls、cd、ln、ln -s、touch、mkdir、rmdir；
ERE：扩展的正则表达式
2. 第八章的命令进行操作；
3. 元字符与 find 命令，测试 shell 对元字符的解释，并说明现象；
4. Cat 与重定向，并利用目录及文件的基本思想和文件结构 inode、data 等现象做解释；
5. 管道，并对现象做解释，wc、cut 命令；
6. 解释命令序列，带括号和不带括号；
7. 解释 tee 命令过程 ls -c|tee dir,list;
8. HISTORY 命令和 sort 命令、fc 命令、alias 命令
9. Grep 使用 BRE 正则表达式，egrep 使用 ERE 正则表达式

七、实验器材（设备、元器件）：

PC 微机一台；

八、实验步骤：

1. 下载 VMware 虚拟机；
2. 在虚拟机上安装 Ubuntu 服务器版本，进行环境配置；
3. 执行实验要求的各项操作；

4.使用文件系统模型解释操作原理。

九、实验数据及结果分析：

文件系统由 BOOT、superblock、inode、data 构成；Inode 中有 blockpoint 指向 data 中的某一块，还有有引用计数器 refcount、权限、日期、类型、长度、所属组别、用户 id 等；

ERE:

.	匹配任意单个字符
[]	匹配指定范围内的任意字符
[^]	匹配指定范围外的任意字符
[:alnum:]	字母和数字
[:alpha:]	代表任何英文大小写字符，亦即A-Z,a-z
[:lower:]	小写字母
[:upper:]	大写字母
[:blank:]	空白字符（空格和制表符）
[:space:]	水平和垂直的空白字符（比[:blank:]包含的范围广）
[:cntrl:]	不可打印的控制字符（退格、删除、警铃...）
[:digit:]	十进制数字
[:xdigit:]	十六进制数字
[:graph:]	可打印的非空白字符
[:print:]	可打印字符
[:punct:]	标点符号

ERE 次数匹配:

*	匹配前面字符任意次
?	0次或1次
+	1次或多次
{m}	匹配m次
{m,n}	至少m,之多n次

需要 \ 使?、*、+来消除特殊含义（BRE：需要 \ 使?、*、+具有特殊含义）；

1. mkdir: 创建目录

分配新的 inode，在新建目录文件中，既有自身的 inode，也有父目录的 inode；同时 refcount 为 2（自身引用、父目录引用）；

```
unix@unix:~$ mkdir aaa
+ mkdir aaa
```

Cd 命令:

进入目录, bash 内置命令, 直接取出对应目录对应的 inode 列表中的 inode;

```
unix@unix:~$ cd aaa
+ cd aaa
unix@unix:~/aaa$ mkdir bbb
+ mkdir bbb
unix@unix:~/aaa$ cd bbb
+ cd bbb
unix@unix:~/aaa/bbb$
```

Ls 命令+Cp 命令:

Ls 命令是将该目录下 inode 列表中所有文件或子目录名显示出来;

Cp 命令, 先分配 block, 拷贝数据; 再分配 inode; 最后添加目录项至目录文件;

```
unix@unix:~/hello1$ ls
hello2
unix@unix:~/hello1$ vi hello.c
unix@unix:~/hello1$ ls
hello2 hello.c
unix@unix:~/hello1$ cp hello.c hello2/hello_copy.c
unix@unix:~/hello1$ ls
hello2 hello.c
unix@unix:~/hello1$ cd hello2
unix@unix:~/hello1/hello2$ ls
hello_copy.c
unix@unix:~/hello1/hello2$
```

Mv 命令:

Mv 命令修改文件名: 修改目录文件中的目录项;

Mv 命令移动目录: 删除目录文件中该目录项; 在新的目录的目录文件下添加目录项;

```

unix@unix:~/hello1$ ls
hello2  hello.c
unix@unix:~/hello1$ mv hello.c hello2
unix@unix:~/hello1$ ls
hello2
unix@unix:~/hello1$ cd hello2
unix@unix:~/hello1/hello2$ ls
hello1  hello.c
unix@unix:~/hello1/hello2$ █

```

Ln 命令、ln-s 命令：

Ln 命令是硬链接，不分配新的 inode，仅在使用原来的 inode，但 inode 中的 refcount+1；并在目录文件中添加目录项；

Ln -s 命令是软链接，分配新的 inode，但是 blockpoint 仍指向原来的数据块；

```

unix@unix:~/hello1/hello2$ ln hello.c yyy
unix@unix:~/hello1/hello2$ ls
hello1  hello.c  yyy
unix@unix:~/hello1/hello2$ cat yyy

by most of the UNIX operating systems.
hello world
unix@unix:~/hello1/hello2$ ln -s hello.c zzz
unix@unix:~/hello1/hello2$ ls
hello1  hello.c  yyy  zzz
unix@unix:~/hello1/hello2$ cat zzz

by most of the UNIX operating systems.
hello world

```

Touch 命令：

创建一个空文件，分配 inode，但文件大小为 0，blockpoint 不指向数据；

```

unix@unix:~/hello1/hello2$ touch xxx
unix@unix:~/hello1/hello2$ ls
hello1  hello.c  xxx  yyy  zzz

```

2. 书上第八章的一些命令：

Cat+重定向实现文件的复制：

```
unix@unix:~/hello1/hello2$ cat hello2.c  
by most of the UNIX operating systems.  
hello world  
unix@unix:~/hello1/hello2$
```

追加多个文件到一个文件中:

```
unix@unix:~/hello1/hello2$ cat hello2.c hello.c hello1 >> xyz  
unix@unix:~/hello1/hello2$ cat xyz  
  
by most of the UNIX operating systems.  
hello world  
  
by most of the UNIX operating systems.  
hello world  
  
by most of the UNIX operating systems.  
hello world
```

Head 命令:

显示文件头部:

```
unix@unix:~/hello1/hello2$ head -5 xyz  
  
by most of the UNIX operating systems.  
hello world  
  
by most of the UNIX operating systems.
```

Tail 命令:

显示文件后 5 行:

```
root@unix:~# tail -5 world  
The vim is a powerful tool.  
The vim is a good tool.  
The vim is a good tool.  
THE VIM IS A GOOD TOOL.  
The vim is a good tool.
```

显示文件前 5 行:

```
root@unix:~# head -5 world  
This is a new file.  
Vim is a powerful editor.  
The vim is a powerful tool.  
The vim is a good tool.  
The vim is a good tool.  
root@unix:~#
```

Paste 命令：逐行连接文件；

-d 选项：指定域分隔符；

```
root@unix:~# paste -d : first last
David:Back
Daniel:Knee
Gabriel:Smart
:
root@unix:~# paste -d " " first last
David Back
Daniel Knee
Gabriel Smart
```

More 命令：

-10：每屏显示 10 行；

-c：显示每页前清屏；

-d：显示提示；

More -10 -cd who:

```

NAME
    who - show who is logged on

SYNOPSIS
    who [OPTION]... [ FILE | ARG1 ARG2 ]

DESCRIPTION
    Print information about users who are currently logged in.
--More--(11%)[Press space to continue, 'q' to quit.]
```

3.Find 命令：

```

unix@unix:~$ find /usr/include -name "s*.h" -type f
/usr/include/linux/sonypi.h
/usr/include/linux/string.h
/usr/include/linux/seg6_ip tunnel.h
/usr/include/linux/sched.h
/usr/include/linux/sockios.h
/usr/include/linux/securebits.h
/usr/include/linux/selinux_netlink.h
/usr/include/linux/shm.h
/usr/include/linux/swab.h
/usr/include/linux/sock_diag.h
/usr/include/linux/sync_file.h
/usr/include/linux/suspend_ioctls.h
/usr/include/linux/smc.h
/usr/include/linux/sdla.h
/usr/include/linux/serio.h
/usr/include/linux/serial.h
/usr/include/linux/scc.h

```

使用通配符搜索文件：

*：匹配任意个任意字符；

[]：匹配括号中的字符；

?：替换单个字符；

.：作为隐藏文件的开头；

\：消除 shell 元字符的语义；

.*：将其识别为所有文件，fork 出的子进程中命令的参数就改变为所有文件；

```

unix@unix:~/hello1/hello2$ ls -al hello?.c
-rw-rw-r-- 1 unix unix 52 Jun 19 07:49 hello2.c
unix@unix:~/hello1/hello2$ ls
hello1 hello2.c hello.c xxx xyz yyy zzz
unix@unix:~/hello1/hello2$ ls -c hello*
hello2.c hello.c hello1
unix@unix:~/hello1/hello2$ ls -c hello[l2]*
hello2.c hello1
unix@unix:~/hello1/hello2$ cd
unix@unix:~$ ls -c .*
.viminfo .bash_history .Xauthority .sudo_as_admin_successful .bash_logout .bashrc .profile

.:
hello1 test2 exam3 test1.sh whileadd.sh aaa myfirst q
dir.list exam4 ulib.sh untiladd.sh xx.sh bbb 'ps -elH' hello.c

.vim:

.:
unix

.cache:
motd.legal-displayed

.ssh:
authorized keys

```


正则表达式：

.: 解释为任意的元字符；

*: 解释为出现的任意次数；

^: 解释为正则式的开始

\$: 解释为正则式的结束；

[]: 匹配括号中的字符；

[^]: 不匹配括号中的字符；

将文件中全部的 **hello** 均替换为 **world**；

```
unix@unix:~/hello1/hello2$ sed '1,$s/hello/world/g' hello.c
by most of the UNIX operating systems.
world world
```

删除文件中第一和第二行内容：

```
unix@unix:~/hello1/hello2$ sed '1,2s/.*/g' hello.c

hello world
unix@unix:~/hello1/hello2$
```

4.Cat 命令：

```
unix@unix:~/hello1/hello2$ cat yyy
by most of the UNIX operating systems.
hello world
```

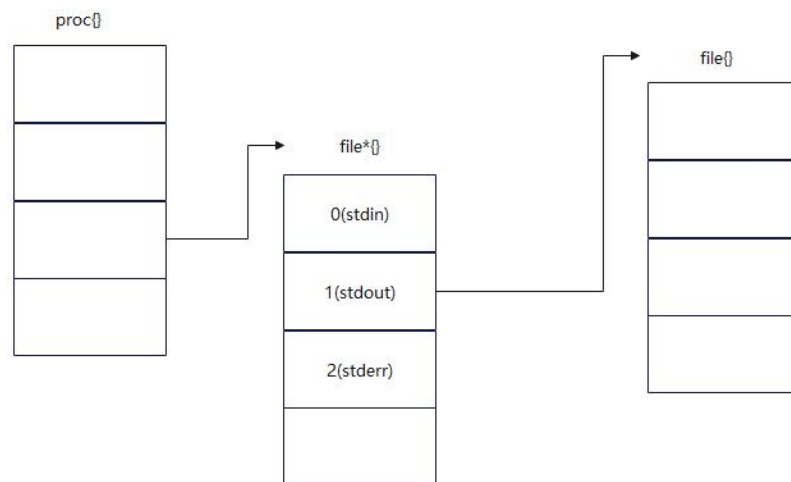
重定向：

进程会指向一个文件指针数组（file*），file*[0]为标准输入；file*[1]为标准输出；file*[2]为标准错误输出；每个指针都指向一个 file 列表；

输出重定向就是将原来的标准输出指针从指向屏幕到指向一个指定的文件；

同时 inode 中分配新的 blockpoint 指向新的 data；

输入重定向就是将原来的标准输入指针的指向从屏幕输入到一个指定的文件；



重定向工作原理

```

unix@unix:~/hello1/hello2$ ls > xxx
unix@unix:~/hello1/hello2$ cat xxx
hello1
hello.c
xxx
yyy
zzz
unix@unix:~/hello1/hello2$ cat < zzz

by most of the UNIX operating systems.
hello world

```

5.管道命令:

由 bash 进程给出 stdin、stdout、stderr;

给出 pipe{0,1,2,3(input),4(output)}

分别 fork 出一个进程执行管道前后的命令;

管道前面的命令由 bash1 子进程执行, 关闭标准输出和 pipe[3], 执行 dup(4), 将输出指向管道 pipe{}, 与后面的命令产生联系;

管道后面的命令由 bash2 执行, 只保留 pipe[0]、pipe[1]、pipe[2]; 标准输入来自管道 pipe{}; 输出为此命令的标准输出;

xargs: 将前面命令的内容作为后面的参数;

求主目录下的所有子目录的个数；

在/usr/include 目录下搜索包含'EPOLLIN'的头文件；

```
unix@unix:~$ ls -al | grep -e '^d' | wc -w
90
unix@unix:~$ find /usr/include -name "*.h" -type f | xargs grep -e 'EPOLLIN'
/usr/include/linux/eventpoll.h:#define EPOLLIN      (__poll_t)0x00000001
/usr/include/x86_64-linux-gnu/sys/epoll.h:    EPOLLIN = 0x001,
/usr/include/x86_64-linux-gnu/sys/epoll.h:#define EPOLLIN EPOLLIN
```

Wc 命令：

-l: 报告行数；

-w: 报告字数；

-c: 报告字符数；

```
unix@unix:~/hello1/hello2$ cat xxx | wc -l
5
unix@unix:~/hello1/hello2$ cat xxx
hello1
hello.c
xxx
yyy
zzz
unix@unix:~/hello1/hello2$ cat xxx | wc -w
wc -w: command not found
unix@unix:~/hello1/hello2$ cat xxx | wc -w
5
unix@unix:~/hello1/hello2$ cat xxx | wc -c
27
unix@unix:~/hello1/hello2$
```

Cut 命令：

-f: 指定域位置；

-c: 指定字符位置；

-d: 指定域分隔符；

在 date 命令下：

以空格为域分隔符，求第一个域的值；

求第 12 到第 13 个字符；

```

unix@unix:~/hello1/hello2$ date | cut -d " " -f 1
Sun
unix@unix:~/hello1/hello2$ date | cut -c 12-13
20
unix@unix:~/hello1/hello2$ █

```

6.命令序列（不带括号）：

按序执行，以分号为间隔；

```

unix@unix:~$ date;pwd;ls -al;
Sun 19 Jun 2022 07:26:51 AM UTC
/home/unix
total 168
drwxr-xr-x 10 unix unix 4096 Jun 19 07:25 .
drwxr-xr-x  3 root root 4096 Mar 25 13:31 ..
drwxrwxr-x  3 unix unix 4096 May  6 13:53 aaa
-rw-----  1 unix unix 5023 Jun 18 16:03 .bash_history
-rw-r--r--  1 unix unix  220 Feb 25  2020 .bash_logout
-rw-r--r--  1 unix unix 3771 Feb 25  2020 .bashrc
drwxrwxr-x  2 unix unix 4096 May  6 13:52 bbb
drwx-----  2 unix unix 4096 Mar  3 09:58 .cache
drwxrwxr-x  2 unix unix 4096 May 28 13:56 exam3
drwxrwxr-x  2 unix unix 4096 May 28 13:56 exam4
drwxrwxr-x  3 unix unix 4096 Jun 19 03:49 hello1
-rw-rw-r--  1 unix unix    1 Mar  4 14:41 hello.c
-rw-rw-r--  1 unix unix  116 Mar 25 11:11 myfirst
-rw-r--r--  1 unix unix   807 Feb 25  2020 .profile
-rw-r--r--  1 unix unix 55217 Mar 25 10:53 'ps -elH'
-rw-rw-r--  1 unix unix    1 Mar 18 12:19 q
drwx-----  2 unix unix 4096 Mar  3 09:57 .ssh
-rw-r--r--  1 unix unix    0 Mar  3 09:58 .sudo_as_admin_successful
-rwxrwxr-x  1 unix unix  115 May 22 12:50 test1.sh
-rwxrwxr-x  1 unix unix   822 May 22 12:53 ulib.sh
-rw-rw-r--  1 unix unix  177 May 22 03:17 untiladd.sh
drwxr-xr-x  2 unix unix 4096 May 28 13:56 .vim
-rw-----  1 unix unix 12419 Jun 19 04:02 .viminfo
-rw-rw-r--  1 unix unix   180 May 22 03:11 whileadd.sh
-rw-----  1 unix unix   100 Jun 19 07:25 .Xauthority
-rwxrwxr-x  1 unix unix   152 May 21 16:17 xx.sh

```

命令序列（带括号）进行重定向：

将括号中的所有命令当作一组命令；统一执行；

```

unix@unix:~$ (ls -al;date;pwd)>test2
unix@unix:~$ ls
aaa bbb exam3 exam4 hello1 hello.c myfirst 'ps -elH' q test1.sh test2
unix@unix:~$ cat test2
total 168
drwxr-xr-x 10 unix unix 4096 Jun 19 07:28 .
drwxr-xr-x  3 root root 4096 Mar 25 13:31 ..
drwxrwxr-x  3 unix unix 4096 May  6 13:53 aaa
-rw-----  1 unix unix 5023 Jun 18 16:03 .bash_history
-rw-r--r--  1 unix unix  220 Feb 25 2020 .bash_logout
-rw-r--r--  1 unix unix 3771 Feb 25 2020 .bashrc
drwxrwxr-x  2 unix unix 4096 May  6 13:52 bbb
drwx-----  2 unix unix 4096 Mar  3 09:58 .cache
drwxrwxr-x  2 unix unix 4096 May 28 13:56 exam3
drwxrwxr-x  2 unix unix 4096 May 28 13:56 exam4
drwxrwxr-x  3 unix unix 4096 Jun 19 03:49 hello1
-rw-rw-r--  1 unix unix   1 Mar  4 14:41 hello.c
-rw-rw-r--  1 unix unix  116 Mar 25 11:11 myfirst
-rw-r--r--  1 unix unix  807 Feb 25 2020 .profile
-rw-r--r--  1 unix unix 55217 Mar 25 10:53 ps -elH
-rw-rw-r--  1 unix unix   1 Mar 18 12:19 q
drwx-----  2 unix unix 4096 Mar  3 09:57 .ssh
-rw-r--r--  1 unix unix   0 Mar  3 09:58 .sudo_as_admin_successful
-rwxrwxr-x  1 unix unix  115 May 22 12:50 test1.sh
-rw-rw-r--  1 unix unix   0 Jun 19 07:28 test2
-rwxrwxr-x  1 unix unix  822 May 22 12:53 ulib.sh
-rw-rw-r--  1 unix unix  177 May 22 03:17 untiladd.sh
drwxr-xr-x  2 unix unix 4096 May 28 13:56 .vim
-rw-----  1 unix unix 12419 Jun 19 04:02 .viminfo
-rw-rw-r--  1 unix unix  180 May 22 03:11 whileadd.sh
-rw-----  1 unix unix  100 Jun 19 07:25 .Xauthority
-rwxrwxr-x  1 unix unix  152 May 21 16:17 xx.sh
Sun 19 Jun 2022 07:28:22 AM UTC
/home/unix

```

7.Tee 命令:

命令分离输出；其产生的输出有两个指针（file*{}），一个指向屏幕（标准输出），另一个指向指定的文件（file*[3]）；

```

unix@unix:~/hello1$ ls -c | tee dir.list
dir.list
hello2
unix@unix:~/hello1$ ls -c | tee dir.list | wc -w
2
unix@unix:~/hello1$ cat dir.list
dir.list
hello2

```

8.

History 命令：显示历史命令的列表

```
unix@unix:~/exam3$ history
1  exit
2  ifconfig
3  sudo apt install net-tools
4  sudo apt install net-tools
5  ifconfig
6  sudo shutdown -P now
7  vim
8  vim hello.c
9  sudo shutdown -p now
10 sudo shutdown -P now
11 sudo apt install vim
12 /bin
13 ls -al
14 ipconfig
15 ifconfig -a
16 sudo apt install vim
17 pwd
18 ls
19 ls root
20 ls home
21 ls -at
22 cd etc
23 shutdown -P now
24 ls -l
25 ls -al
26 cd
27 ls -l
28 ls -al
29 shutdown -P now
30 sudo -i
31 sudo -i
32 ls -al
33 cd etc
34 cd home
```

Sort 命令:

- f: 忽略大小写;
- n: 按整个数的大小值排序;
- o: 将输出存储在指定的文件，而不是输出到标准设备;
- r: 按降序排序;
- b: 忽略前导空格;
- d: 按字典顺序排序;


```

root@unix:~# sort world
The vim is a good tool.
The vim is a good tool.
The vim is a good tool.
THE VIM IS A GOOD TOOL.
The vim is a powerful tool.
This is a new file.
Vim is a powerful editor.
root@unix:~# sort -fn world
The vim is a good tool.
The vim is a good tool.
The vim is a good tool.
THE VIM IS A GOOD TOOL.
The vim is a powerful tool.
This is a new file.
Vim is a powerful editor.
root@unix:~# sort -f -r -o world2 world
root@unix:~# cat world2
Vim is a powerful editor.
This is a new file.
The vim is a powerful tool.
THE VIM IS A GOOD TOOL.
The vim is a good tool.
The vim is a good tool.
The vim is a good tool.

```

Lc 命令:

显示历史的前 10 条命令:

```

unix@unix:~$ fc -l -10
375      ls
376      cat test2
377      ls -c | tee dir.list
378      ls
379      cd hello1
380      ls -c | tee dir.list
381      ls -c | tee dir.list | wc -w
382      cat dir.list
383      cd
384      sudo -i
unix@unix:~$

```

Alias 命令:

为命令取别名;

```

unix@unix:~$ alias ll="ls -al"
unix@unix:~$ ll
total 176
drwxr-xr-x 10 unix unix 4096 Jun 19 07:32 .
drwxr-xr-x  3 root root 4096 Mar 25 13:31 ..
drwxrwxr-x  3 unix unix 4096 May  6 13:53 aaa
-rw-----  1 unix unix 5023 Jun 18 16:03 .bash_history
-rw-r--r--  1 unix unix  220 Feb 25  2020 .bash_logout
-rw-r--r--  1 unix unix 3771 Feb 25  2020 .bashrc
drwxrwxr-x  2 unix unix 4096 May  6 13:52 bbb
drwx-----  2 unix unix 4096 Mar  3 09:58 .cache
-rw-rw-r--  1 unix unix  106 Jun 19 07:32 dir.list
drwxrwxr-x  2 unix unix 4096 May 28 13:56 exam3
drwxrwxr-x  2 unix unix 4096 May 28 13:56 exam4
drwxrwxr-x  3 unix unix 4096 Jun 19 07:32 hello1
-rw-rw-r--  1 unix unix   1 Mar  4 14:41 hello.c
-rw-rw-r--  1 unix unix  116 Mar 25 11:11 myfirst
-rw-r--r--  1 unix unix  807 Feb 25  2020 .profile
-rw-r--r--  1 unix unix 55217 Mar 25 10:53 'ps -elH'
-rw-rw-r--  1 unix unix   1 Mar 18 12:19 q
drwx-----  2 unix unix 4096 Mar  3 09:57 .ssh
-rw-r--r--  1 unix unix   0 Mar  3 09:58 .sudo_as_admin_successful
-rwxrwxr-x  1 unix unix  115 May 22 12:50 test1.sh
-rw-rw-r--  1 unix unix 1433 Jun 19 07:28 test2
-rwxrwxr-x  1 unix unix  822 May 22 12:53 ulib.sh
-rw-rw-r--  1 unix unix  177 May 22 03:17 untiladd.sh
drwxr-xr-x  2 unix unix 4096 May 28 13:56 .vim
-rw-----  1 unix unix 12419 Jun 19 04:02 .viminfo
-rw-rw-r--  1 unix unix  180 May 22 03:11 whileadd.sh
-rw-----  1 unix unix  100 Jun 19 07:25 .Xauthority
-rwxrwxr-x  1 unix unix  152 May 21 16:17 xx.sh

```

Unalias: 取消别名:

```

unix@unix:~$ unalias ll
unix@unix:~$ alias
alias alert='notify-send --urgency=low -i "${[ $? = 0 ] && echo te
//;s/[:&|]\s*alert$//'\''}"'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ls='ls --color=auto'

```

10. grep 查找（利用 BRE）:

查找非空行;

查找不满足第一个字母为大写，后面均为字母的行（至少有一个）;

```

unix@unix:~$ grep -vn "^$" hello
1:This is the first file
unix@unix:~$ grep -vn "^[:space:]]$" hello
1:This is the first file
unix@unix:~$ grep -vn "^[:upper:]][:alpha:]]\{1,\}$" hello
1:This is the first file

```


Egrep 查找（利用 ERE）：

?: 匹配前面字符 0 或 1 次；

*: 匹配字符任意次；

+: 匹配字符 1 次或多次；

```
unix@unix:~$ egrep -n "^[:upper:]*$" hello
3:THIS
unix@unix:~$ egrep -n "^[:upper:]?$" hello
unix@unix:~$ egrep -n "^[:upper:]+$" hello
3:THIS
unix@unix:~$ egrep -n "^[:upper:]*[^[:upper:]]*" hello
unix@unix:~$ egrep -n "^[:upper:].*$" hello
1:THIS is the first file
3:THIS
unix@unix:~$ cat hello
THIS is the first file
this
THIS
that
```

十、实验结论：

此次实验验证了书上的关于 UNIX 文件系统的基本操作和命令；还通过文件系统的模型和目录的基本原理解释了管道、文件操作的内在原理；同时测试了系统对 shell 元字符的解释，确认了 find 命令的有效性；理解了正则表达式对内容查找的重要性。

十一、总结及心得体会：

通过这次实验我了解了 UNIX 文件系统的基本操作，以及 shell 的基本编程；对管道、重定向等知识有了更深的理解；对常见的文件命令有了了解，对命令更加熟悉。

十二、对本实验过程及方法、手段的改进建议：

希望可以对文件系统的原理有更深入的认识。

报告评分：

指导教师签字：