



哈爾濱工業大學(威海)

Harbin Institute of Technology at Weihai

Linux 操作系统

实验报告

院 系： 计算机科学与技术学院

班 级： 2104102

姓 名： 蒲海博

学 号： 2021211041

指导教师： 闫健恩

哈尔滨工业大学（威海）

报告撰写要求

- 1) 正文使用五号字，单倍行距，中文使用宋体，英文使用 **times new roma** 字体；
- 2) 注意行文的规范，段落空行等；
- 3) 实验报告电子版请提供 **pdf** 格式；
- 4) 报告采用**双面打印**，使用左侧装订方式；

(此页需要打印)

哈尔滨工业大学（威海）计算机学院

《Linux 操作系统》实验评定表

| 实验一 Linux 系统基本管理与应用（共 40 分） | | | |
|-----------------------------|--|------|----|
| 评审项目 | 评审内容 | 成绩标准 | 成绩 |
| 实验内容完成情况 | 实验内容所有项目是否完成，实验结果是否准确，结果说明是否完整等 | 30 分 | |
| 报告内容完整性 | 文档格式是否规范；文字是否规范；图表是否规范；术语是否准确等 | 10 分 | |
| 实验二 shell 程序设计（共 60 分） | | | |
| 评审项目 | 评审内容 | 成绩标准 | 成绩 |
| 实验内容完成情况 | 实验内容所有项目是否完成，程序是否能够运行，实验结果是否准确，结果分析是否清晰明确等 | 50 分 | |
| 报告内容完整性 | 文档格式是否规范；文字是否规范；图表是否规范；术语是否准确等 | 10 分 | |
| 总体评价 | | 总成绩 | |

实验1 Linux 系统基本管理与应用

1.实验目的

掌握 Linux 系统常用的 shell 命令，便于系统的管理与应用。

2.实验内容

熟悉 Linux 操作系统的常用 shell 命令、控制台基本操作

3.实验题目：

- 以 hit+个人学号建立用户账号(例如建立 hit150410101 的账户)，并以此账号进行课程所有相关实验操作。
- 文件管理操作（4 个以上常用命令）
- 目录管理操作（4 个以上常用命令）
- 管理使用与权限设置（3 个以上常用命令）
- 文件阅读与查找（find 与 grep 命令）
- 系统用户和组的管理，包括增加修改删除等基本管理方法（4 个以上常用命令）
- 网络配置与网络状态分析和检查（4 个以上常用命令）

4.实验报告内容：

以书写形式记录下每个命令操作的情况，尝试不同命令的参数设置，掌握不同参数的功能，对比不同参数使用获得的结果有何异同以及遇到的问题和解决方法。

参考如下撰写方式：

1) 文件管理操作

(1) cat 命令

cat 命令的用途是连接文件或标准输入并打印。这个命令常用来显示文件内容，或者将几个文件连接起来显示，或者从标准输入读取内容并显示，它常与重定向符号配合使用。

a. 命令格式：

cat [选项] [文件]...

b. 命令功能：

cat 主要有三大功能：

1.一次显示整个文件:cat filename

2.从键盘创建一个文件:cat > filename 只能创建新文件,不能编辑已有文件.

3.将几个文件合并为一个文件:cat file1 file2 > file

c. 命令参数：

| | |
|------------------------|------------------------------|
| -A, --show-all | 等价于 -vET |
| -b, --number-nonblank | 对非空输出行编号 |
| -e | 等价于 -vE |
| -E, --show-ends | 在每行结束处显示 \$ |
| -n, --number | 对输出的所有行编号,由 1 开始对所有输出的行数编号 |
| -s, --squeeze-blank | 有连续两行以上的空白行，就代换为一行的空白行 |
| -t | 与 -vT 等价 |
| -T, --show-tabs | 将跳格字符显示为 ^I |
| -u | (被忽略) |
| -v, --show-nonprinting | 使用 ^ 和 M- 引用，除了 LFD 和 TAB 之外 |

```

[hit2021211041@localhost ~]$ cat>file
I'm a HITer!
Are you ok?
I'm fine!
Let's begin!
^Z
[1]+  Stopped                  cat > file
[hit2021211041@localhost ~]$ cat file
I'm a HITer!
Are you ok?
I'm fine!
Let's begin!
[hit2021211041@localhost ~]$ cat -A file
I'm a HITer!$
Are you ok?$
I'm fine!$
Let's begin!$
[hit2021211041@localhost ~]$ cat -b file
 1 I'm a HITer!
 2 Are you ok?
 3 I'm fine!
 4 Let's begin!
[hit2021211041@localhost ~]$ cat -e file
I'm a HITer!$
Are you ok?$
I'm fine!$
Let's begin!$
[hit2021211041@localhost ~]$ cat -n file
 1 I'm a HITer!
 2 Are you ok?
 3 I'm fine!
 4 Let's begin!
[hit2021211041@localhost ~]$

```

(2) more 命令

more 命令，功能类似 **cat**，**cat** 命令是整个文件的内容从上到下显示在屏幕上。**more** 会以一页一页的显示方便使用者逐页阅读，而最基本的指令就是按空白键（space）就往下 一页显示，按 **b** 键就会往回（back）一页显示，而且还有搜寻字串的功能。**more** 命令从前向后读取文件，因此在启动时就加载整个文件。

a. 命令格式：

more [-dlfpsu] [-num] [+/- pattern] [+ linenum] [file ...]

b. 命令功能：

more 命令和 **cat** 的功能一样都是查看文件里的内容，但有所不同的是 **more** 可以按页来查看文件的内容，还支持直接跳转行等功能。

c. 命令参数：

| | |
|-------------------|--|
| +n | 从第 n 行开始显示 |
| -n | 定义屏幕大小为 n 行 |
| +/-pattern | 在每个档案显示前搜寻该字串,然后从该字串前两行之后开始显示 |
| -c | 从顶部清屏，然后显示 |
| -d | 提示“Press space to continue, 'q' to quit（按空格键继续，按 q 键退出）”，禁用响铃功能 |
| -l | 忽略 Ctrl+I （换页）字符 |
| -p | 通过清除窗口而不是滚屏来对文件进行换页，与 -c 选项相似 |
| -s | 把连续的多个空行显示为一行 |
| -u | 把文件内容中的下画线去掉 |

```

I'm a HITer!
Are you ok?
I'm fine!
Let's begin!
[hit2021211041@localhost ~]$ more file
I'm a HITer!
Are you ok?
I'm fine!
Let's begin!
[hit2021211041@localhost ~]$ more +2 file
Are you ok?
I'm fine!
Let's begin!
[hit2021211041@localhost ~]$ more -3 file
I'm a HITer!
Are you ok?
I'm fine!
--More--(72%)

```

(3) less 命令

less 命令也是对文件或其它输出进行分页显示的工具。less 的用法的功能与 more 指令类似但是比起 more 更加强大，支持各种显示终端。在 more 的时候，我们并没有办法向前面翻，只能往后面看，但若使用了 less 时，就可以使用 [pageup] [pagedown] 等按键的功能来往往后翻看文件，更容易用来查看一个文件的内容。除此之外，在 less 里可以拥有更多的搜索功能，不止可以向下搜，也可以向上搜。less 指令在显示文件内容时，并不是一次将整个文件加载之后才显示，而是根据显示需要加载内容，对于显示大型文件具有较高的效率。

a.命令格式:

less [选项] 文件名

b.命令功能:

less 实际上是 more 的改进版，其命令的直接含义是的 more 反义。它的功能比 more 命令更灵活

b 向后翻一页 d 向后翻半页
 h 显不帮助界面 Q 退出 less 命令
 u 向前滚动半页 y 向前滚动一行
 空格键 滚动一行 回车键 滚动一页
 [pagedown] 向下翻动一页 [pageup] : 向上翻动一页

c.命令参数:

-b <缓冲区大小>设置缓冲区的大小
 -e 当文件显示结束后，自动离开
 -f 强迫打开特殊文件，例如外围设备代号、目录和二进制文件
 -g 只标志最后搜索的关键词
 -i 忽略搜索时的大小写
 -m 显示类似 more 命令的百分比
 -N 显示每行的行号
 -o <文件名> 将 less 输出的内容在指定文件中保存起来
 -Q 不使用警告音
 -s 显示连续空行为行
 -S 行过长时间将超出部分舍弃

```
[hit2021211041@localhost ~]$ less file
I'm a HITer!
Are you ok?
I'm fine!
Let's begin!

[5]+  Stopped                  less file
[hit2021211041@localhost ~]$ less -b file
Number is required after -b (--buffers)
Missing filename ('less --help' for help)
[hit2021211041@localhost ~]$ less -N file
  1 I'm a HITer!
  2 Are you ok?
  3 I'm fine!
  4 Let's begin!
file (END)
```

(4) head 命令

head 命令是用来显示开头某个数量的文字区块，显示档案的开头至标准输出中。

a.命令格式:

head [参数] 文件名

b.命令功能:

head 命令在屏幕上显示指定文件的开头若干行，行数由参数值来确定。显示行数的默认值是 10

c.命令参数:

-n<数字> 指定显示头部内容的行数
 -c<字符数> 指定显示头部内容的字符数
 -v 总是显示文件名的头信息
 -q 不显示文件名的头信息。
 -x <数字>将“tab”键显示为规定的数字空格

```
[hit2021211041@localhost ~]$ head file
I'm a HITer!
Are you ok?
I'm fine!
Let's begin!
[hit2021211041@localhost ~]$ head -n2 file
I'm a HITer!
Are you ok?
[hit2021211041@localhost ~]$ head -c11 file
I'm a HITer[hit2021211041@localhost ~]$ head -v file
==> file <==
I'm a HITer!
Are you ok?
I'm fine!
Let's begin!
[hit2021211041@localhost ~]$ head -q file
I'm a HITer!
Are you ok?
I'm fine!
Let's begin!
[hit2021211041@localhost ~]$
```

(5) mv 命令

mv 是 move 的缩写，可以用来移动文件或者将文件改名（move (rename) files）

a.命令格式:

mv [选项] 源文件或目录 目标文件或目录

b.命令功能:

视 mv 命令中第二个参数类型的不同（是目标文件还是目标目录），mv 命令将文件重命名或将其移至一个新的目录中。当第二个参数类型是文件时，mv 命令完成文件重命名，此时，源文件只能有一个（也可以是源目录名），它将所给的源文件或目录重命名为给定

的目标文件名。当第二个参数是已存在的目录名称时，源文件或目录参数可以有多个，mv 命令将各参数指定的源文件均移至目标目录中。在跨文件系统移动文件时，mv 先拷贝，再将原有文件删除，而链至该文件的链接也将丢失。即 mv 可移动文件或目录，或是更改文件或目录的名称。

c. 命令参数:

- b 或--backup 若需覆盖文件，则覆盖前先行备份。
- f 或--force 若目标文件或目录与现有的文件或目录重复，则直接覆盖现有的文件或目录且不会询问。
- i 或--interactive 覆盖前先行询问用户。
- n 或--no-clobber 不覆盖已存在文件，如果您指定了-i、-f、-n 中的多个，仅最后一个生效。

-S<附加字尾>或--suffix=<附加字尾> 与-b 参数一并使用，可指定备份文件的所要附加的字尾，即替换备份文件后缀。

-t 或--target-directory=DIRECTORY 将所有参数指定的源文件或目录移动至指定目录。

-T 或--no-target-directory 将目标文件视作普通文件处理。

-u 或--update 在移动或更改文件名时，若目标文件已存在，且其文件日期比源文件新，则不覆盖目标文件。

-v 或--verbose 执行时显示详细的信息。

-V=<方法>或--version-control=<方法> 与-b 参数一并使用，可指定备份的方法。

--help 在线帮助。查看该命令功能和可带参数等。

```
[hit2021211041@localhost ~]$ mv file file1
[hit2021211041@localhost ~]$ cat file1
I'm a HITer!
Are you ok?
I'm fine!
Let's begin!
[hit2021211041@localhost ~]$ _
```

2) 目录管理操作

(1) mkdir 命令

a. 命令格式

mkdir [选项] 目录名

b. 命令功能

创建由目录名命令的目录

c. 命令参数

-m, -mode <目录属性> 建立目录时同时设置目录的权限。

_p, -parents 可以是一个路径名称。此时若路径中的某些目录尚不存在，加上此选项后，系统将自动建立好那些尚不存在的目录，即一次可以建立多个目录；

-v, -verbose 每次创建新目录都显示信息

—help 显示此帮助信息并退出

—version 输出版本信息并退出

```
[hit2021211041@localhost ~]$ mkdir linux
[hit2021211041@localhost ~]$ mkdir -v linux
mkdir: cannot create directory 'linux': File exists
[hit2021211041@localhost ~]$ mkdir -v CS
mkdir: created directory 'CS'
```

(2) rmdir 命令

a. 命令格式

rmdir [选项] 目录名

b. 命令功能

从一个目录中删除一个或多个空的子目录。

c. 命令参数

-P: 递归删除目录 dir name,当子目录删除后其父目录为空时，

也一同被删除。如果整个路径被删除或者由于某种原因保留部分路径，则系统在标准输出上显示相应的信息

-v 或-verbose: 显示指令执行过程
-vp: 显示删除多个目录的信息
—help: 显示此帮助信息并退出
—version: 输出版本信息并退出

```
[hit2021211041@localhost ~]$ rmdir -v linux  
rmdir: removing directory, 'linux'  
[hit2021211041@localhost ~]$
```

(3) cd 命令

a.命令格式:

cd [目录名]

b.命令功能:

该命令用来改变工作目录。当不带任何参数时，返回到用户的主目录

c.命令参数:

cd 进入用户主目录;
cd ~ 进入用户主目录;
cd - 返回进入此目录之前所在的目录;
cd .. 返回上级目录(若当前目录为"/", 则执行完后还在"/"; ".."为上级目录的意思);
cd ../.. 返回上两级目录;
cd !\$ 把上个命令的参数作为 cd 参数使用。

```
[hit2021211041@localhost ~]$ cd  
[hit2021211041@localhost ~]$ ls  
CS file1 folder test text text.txt txt  
[hit2021211041@localhost ~]$ cd CS  
[hit2021211041@localhost CS]$
```

(4) pwd 命令

a.命令格式:

Pwd

b.命令功能:

不带任何选或参数，用于显示出当前工作目录的绝对路径。

```
[hit2021211041@localhost ~]$ pwd  
/home/hit2021211041  
[hit2021211041@localhost ~]$
```

(5) ls 命令

a.命令格式:

ls [选项] [目录或文件]

b.命令功能:

列出指定目录的内容，默认情况下，输出条目按字母顺序排列

c.命令参数:

-a 显示所有文件，包含隐藏文件
-A 显示所有文件，包含隐藏文件，但不包含.及..
-l 显示为 long format (长格式)，列出文件的类型、权限、链接数、owner、group、大小，时间，名字
-d 不展开目录，只显示目录自身，一般与-l 配合使用以显示目录自身的

属性信息（只显本当前目录的内容）

- l 数字 1,成列显示内容
- s 以文件大小排序显示，默认从大到小-r 后，从小到大
- u 按存放顺序排序显示
- X 按扩展名的首字母来排序
- t 按 mtime 排序（先显示时间最近的）
- ul 按 atime 排序（先显示时间最近的）
- ct 按 ctime 排序（先显示时间最近的）

```
[hit2021211041@localhost ~]$ ls -a
.  ..  .bash_logout  .bash_profile  .bashrc  CS  file1  folder  test  text  text.txt  txt
[hit2021211041@localhost ~]$ ls -A
.bash_logout  .bash_profile  .bashrc  CS  file1  folder  test  text  text.txt  txt
[hit2021211041@localhost ~]$ ls -d
.
[hit2021211041@localhost ~]$ ls -l
CS
file1
folder
test
text
text.txt
txt
[hit2021211041@localhost ~]$ ls -s
total 28
4 CS  4 file1  4 folder  4 test  4 text  4 text.txt  4 txt
[hit2021211041@localhost ~]$ ls -u
```

3) 管理使用与权限设置

(1)ls_l 命令

a.命令格式:

ls-l 文件和目录列表

b.命令功能:

查看文件或目录的存取权限、读写执行的权限。

```
[hit2021211041@localhost ~]$ ls -l
total 28
drwxr-xr-x. 2 hit2021211041 hit2021211041 4096 Jul 11 22:27 CS
-rw-r--r--. 1 hit2021211041 hit2021211041  48 Jul 11 19:46 file1
drwxr-xr-x. 2 hit2021211041 hit2021211041 4096 Jul 11 22:19 folder
-rw-r--r--. 1 hit2021211041 hit2021211041  33 Jul 11 05:13 test
-rw-r--r--. 1 hit2021211041 hit2021211041  30 Jul 11 05:04 text
-rw-r--r--. 1 hit2021211041 hit2021211041   9 Jul 11 04:40 text.txt
-rw-r--r--. 1 hit2021211041 hit2021211041  26 Jul 11 05:26 txt
[hit2021211041@localhost ~]$
```

(2)chmod 命令

a.命令格式:

chmod [选项] 文件和目录列表

b.命令功能:

改变或设置文件或目录的存取权限、读写执行的权限。

c.命令参数:

u:user5 文件的属主

g:group5 文件的属主所在的组，属组

o: other, 其他的用户

a:all5 以上三种人

chmod [u/g/o/a] [+/-/=] [权限] FileName/DirecName

其中: +: 增加指定的权限

-: 去掉指定的权限

=: 清除所选的权限并将其设置为制定的权限

ft: chmod 也允许使用八进制符号作为模式，改变权限。

```

[hit2021211041@localhost ~]$ chmod a+w file1
[hit2021211041@localhost ~]$ ls -l
total 28
drwxr-xr-x. 2 hit2021211041 hit2021211041 4096 Jul 11 22:27 CS
-rw-rw-rw-. 1 hit2021211041 hit2021211041 48 Jul 11 19:46 file1
drwxr-xr-x. 2 hit2021211041 hit2021211041 4096 Jul 11 22:19 folder
-rw-r--r--. 1 hit2021211041 hit2021211041 33 Jul 11 05:13 test
-rw-r--r--. 1 hit2021211041 hit2021211041 30 Jul 11 05:04 text
-rw-r--r--. 1 hit2021211041 hit2021211041 9 Jul 11 04:40 text.txt
-rw-r--r--. 1 hit2021211041 hit2021211041 26 Jul 11 05:26 txt
[hit2021211041@localhost ~]$ _

```

(3) umask 命令

a.命令格式:

umask [选项] [掩码]

b.命令功能:

用来设置限制新文件权限的掩码。

```

[hit2021211041@localhost ~]$ umask
0022
[hit2021211041@localhost ~]$ umask g=rw
[hit2021211041@localhost ~]$ umask
0012
[hit2021211041@localhost ~]$ umask 011
[hit2021211041@localhost ~]$ umask
0011
[hit2021211041@localhost ~]$ _

```

(4) chown 命令

a.命令格式:

chown [选项]... [所有者][:[组]] 文件...

b.命令功能:

通过 chown 改变文件的拥有者和群组。在更改文件的所有者或所属群组时，可以使用用户名和用户识别码设置。普通用户不能将自己的文件改变成其他的拥有者。其操作权限一般为管理员。

c.命令参数:

-c 或--changes 效果类似"-v"参数，但仅显示更改的部分。

-f 或--quite 或--silent 不显示错误信息。

-h 或--no-dereference 之对符号连接的文件作修改，而不更动其他任何相关文件。

-R 或--recursive 递归处理，将指定目录下的所有文件及子目录一并处理。

-v 或--version 显示指令执行过程。

--dereference 效果和"-h"参数相同，作用于符号链接的指向，而不是链接文件本身。

--reference=<参考文件或目录> 把指定文件或目录的拥有者与所属群组全部设成和参考文件或目录的拥有者与所属群组相同。

--from=<当前用户: 当前群组> 只有当前用户和群组跟指定的用户和群组相同时才进行改变。

```

[hit2021211041@localhost ~]$ chown -R hit2021211041 file1
[hit2021211041@localhost ~]$

```

4) 文件阅读与查找

(1) find 命令

a.命令格式:

find [路径] [表达式]

b.命令功能:

find 命令从左向右分析各个参数，然后依次搜索目录。

c.命令参数:

-amin n : 在过去 n 分钟内被读取过

-anewer file : 比文件 file 更晚被读取过的文件-

atime n : 在过去 n 天内被读取过的文件
-cmin n : 在过去 n 分钟内被修改过
-cnewer file : 比文件 file 更新的文件
-ctime n : 在过去 n 天内被修改过的文件注: 以上+n 表示 n 以 前, -n 表示 n 以内
-empty : 空的文件
-gid n or -groupname : gid 是 n 或是 group 名称是 name
-ipath p, -path p : 路径名称符合 p 的文件, ipath 会忽略大小写
-name name, -iname name : 文件名称符合 name 的文件。iname 会忽略大小写
-size n : 文件大小是 n,单位,b 代表 512 位元组的区块,c 表示字元数,k 表示 kilo bytes。
-type c : 文件类型是 c 的文件。

```

[hit2021211041@localhost ~]$ find -amin -5
./CS
./folder
./text
./file1
[hit2021211041@localhost ~]$ find -atime -3
.
./CS
./bashrc
./folder
./text
./test
./file1
./text.txt
./txt
./bash_profile
[hit2021211041@localhost ~]$ find -ctime -4
.
./CS
./bashrc
./folder
./bash_logout
./text
./test
./file1
./text.txt
./txt
./bash_profile
[hit2021211041@localhost ~]$
  
```

(2) grep 命令

a.命令格式

grep [参数] 文件名

b.命令功能

grep 命令用于在指定文件中搜索特定模式及定位特定主题。

c.命令参数

- i 搜索时, 忽略大小写
- c 只输出匹配行的数量
- l 只列出符合匹配的文件名, 不列出具体的匹配行
- n 列出所有的匹配行, 显示行号
- h 查询多文件时不显示文件名
- s 不显示不存在、没有匹配文本的错误信息
- V 显示不包含匹配文本的所有行
- W 匹配整词
- X 匹配整行
- r 递归搜索
- q 禁止输出任何结果, 已退出状态表示搜索是否成功
- b 打印匹配行距文件头部的偏移量, 以字节为单位
- o 与 -b 结合使用, 打印匹配的词语距文件头部的偏移量, 以字节为单位

```
[hit2021211041@localhost ~]$ grep -ni I file1
1:I'm a HITer!
3:I'm fine!
4:Let's begin!
[hit2021211041@localhost ~]$ _
```

5) 系统用户和组的管理，包括增加修改删除等基本管理方法

(1) adduser 命令

a.命令格式:

`sudo adduser [用户名]`

b.命令功能:

用于新建用户

c.命令参数:

- C: 加上备注文字，备注文字保存在 `passwd` 的备注栏中。
- d: 指定用户登入时的主目录，替换系统默认值 `/home/v 用户名`
- D: 变更预设值。
- e: 指定账号的失效日期，日期格式为 `MM/DD/YY`,例如 `06/30/12` 缺省表示永久有效。
- f: 指定在密码过期后多少天即关闭该账号。如果为 `0` 账号立即被停用； 如果为 `-1` 则账号一直可用。默认值为 `-1`。
- g: 指定用户所属的群组。值可以使组名也可以是 `GID`。用户组必须已经存在的，期默认值为 `100`,即 `users`
- G: 指定用户所属的附加群组。
- m: 自动建立用户的登入目录。
- M: 不要自动建立用户的登入目录。
- n: 取消建立以用户名称为名的群组。
- r: 建立系统账号。
- s: 指定用户登入后所使用的 `shell` 默认值为 `/bin/bash`
- u: 指定用户 ID 号。该值在系统中必须是唯一的。`0~499` 默认是保留给系统用户账号使用的，所以该值必须大于 `499`。

```
[hit2021211041@localhost ~]$ sudo adduser hit01

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for hit2021211041:
[hit2021211041@localhost ~]$ sudo adduser hit
[hit2021211041@localhost ~]$ sudo adduser -m HIT
[hit2021211041@localhost ~]$ sudo adduser hit11
[hit2021211041@localhost ~]$ sudo adduser hit
adduser: user 'hit' already exists
[hit2021211041@localhost ~]$ _
```

(2) userdel 命令

a.命令格式:

`userdel [-r] [M 户账号]`

b.命令功能:

用于删除指定的用户账号

c.命令参数:

- f: 强制删除用户，即使用户当前已登录

-r: 删除用户的同时删除与用户相关的所有文件

```
[hit2021211041@localhost ~]$ sudo userdel hit
[hit2021211041@localhost ~]$ sudo userdel -r hit01
userdel: user 'hit01' does not exist
[hit2021211041@localhost ~]$ sudo userdel hit
userdel: user 'hit' does not exist
[hit2021211041@localhost ~]$
```

(3)groups 命令

a.命令格式:

groups 用户名

b.命令功能:

用于显示指定用户所属的组, 如未指定用户则显示当前用户所属的组

```
[hit2021211041@localhost ~]$ groups hit2021211041
hit2021211041 : hit2021211041 wheel
[hit2021211041@localhost ~]$ groups root
root : root
[hit2021211041@localhost ~]$ _
```

(4)groupadd 命令

a.命令格式:

groupadd [参数] 组名

b.命令功能:

用于创建一个组

c.命令参数:

| | |
|------------------------|------------------------------|
| _f, -force | 如果组已经存在则成功退出; 如果 GID 已经存在则取消 |
| -g, -gid GID | 为新组使用 GID; |
| _h, -help | 显示此帮助信息并推出; |
| -K, -key KEY=VALUE | 不使用/etc/login.defs 中的默认值; |
| -o, -non-unique | 允许创建有重复 GID 的组; |
| -p, -password PASSWORD | 为新组使用此加密过的密码; |
| -r, -system | 创建一个系统账户; |

```
[hit2021211041@localhost ~]$ sudo groupadd newGroup
[hit2021211041@localhost ~]$ sudo groupadd newGroup
groupadd: group 'newGroup' already exists
[hit2021211041@localhost ~]$
```

(5)groupdel 命令

a.命令格式:

groupdel [群组名称]

b.命令功能:

用于删除指定的组账号。若该群组中仍包括某些用户, 则必须先删除这些用户后, 方能删除群组

```
[hit2021211041@localhost ~]$ sudo groupdel newGroup
[hit2021211041@localhost ~]$ sudo groupdel newGroup
groupdel: group 'newGroup' does not exist
[hit2021211041@localhost ~]$
```

6) 网络配置与网络状态分析和检查

(1) ifconfig 命令

a.命令格式:

ifconfig [网络设备][down up -allmulti -arp -promise] [add<地址>][delv 地址] [[vhv 网络设备类型>v 硬件地址>][io_addr<I/O 地址>][irq<IRQ 地址>][media<网络媒介类型>][mem_start<内存地址>][metric<数目>][mtu<字节>][netmask<子网掩码>][tunnelv 地址>][broadcastv 地址] [[pointpointv 地址]][IP 地址]

b.命令功能:

用来显示和配置网络设备

```
[hit2021211041@hecs-5357 ~]$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.144 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::f816:3eff:fe49:a131 prefixlen 64 scopeid 0x20<link>
    ether fa:16:3e:49:a1:31 txqueuelen 1000 (Ethernet)
    RX packets 14398 bytes 16577636 (15.8 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3748 bytes 294167 (287.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[hit2021211041@hecs-5357 ~]$ _
```

(2)route 命令

a.命令格式:

route [-f][-p][Command][Destination][maskNetmask][Gateway][metricMetric][[if Interface]]

b.命令功能:

用来查看和设置 Linux 系统的路由信息，以实现与其他网络的通讯。

c.命令参数:

- A: 设置地址类型
- C: 打印 Linux 内核的路由缓存
- V: 显示详细信息
- n: 不执行 DNS 反向查找，直接显示数字形式的 IP 地址
- e: netstat 格式显示路由表
- net: 到一个网络的路由表
- host: 到一个主机的路由表


```
[hit2021211041@hecs-5357 ~]# route
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
default          _gateway        0.0.0.0          UG    100    0      0 eth0
169.254.169.254 _gateway        255.255.255.255 UGH    100    0      0 eth0
192.168.0.0      0.0.0.0         255.255.255.0    U     100    0      0 eth0
[hit2021211041@hecs-5357 ~]#
```

(3)ping 命令

a.命令格式:

ping [选项] 主机名 (或 IP 地址)

b.命令功能:

用于测试本主机和目标主机的连通性

c.命令参数:

- q 不显示任何传送封包的信息，只显示最后的结果
- n 只输出数值
- R 记录路由过程
- c count 总次数
- i 时间间隔
- t 存活数值：设置存活数值 TTL 的大小

```
[hit2021211041@hecs-5357 ~]# ping www.baidu.com
PING www.a.shifen.com (110.242.68.3) 56(84) bytes of data:
64 bytes from 110.242.68.3 (110.242.68.3): icmp_seq=1 ttl=47 time=17.3 ms
64 bytes from 110.242.68.3 (110.242.68.3): icmp_seq=2 ttl=47 time=17.4 ms
64 bytes from 110.242.68.3 (110.242.68.3): icmp_seq=3 ttl=47 time=17.3 ms
^Z
[1]+  Stopped                  ping www.baidu.com
[hit2021211041@hecs-5357 ~]# _
```

(4)traceroute 命令

a.命令格式:

traceroute [参数] [主机]

b.命令功能:

traceroute 指令让你追踪网络数据包的路由途径，预设数据包大小是 40Bytes，用户可另行设置。具体参数格式：traceroute [-dFlnrvx][[-f<存活数值>][[-g<网关>...][[-i<网络界面>][[-m<存活数值>][[-p<通信端口>][[-s<来源地址>][[-t<服务类型>][[-w<超时秒数>][主机名称或 IP 地址][数据包大小]

c.命令参数:

- d 使用 Socket 层级的排错功能。
- f 设置第一个检测数据包的存活数值 TTL 的大小。
- F 设置勿离断位。
- g 设置来源路由网关，最多可设置 8 个。
- i 使用指定的网络界面送出数据包。
- I 使用 ICMP 回应取代 UDP 资料信息。
- m 设置检测数据包的最大存活数值 TTL 的大小。
- n 直接使用 IP 地址而非主机名称。
- p 设置 UDP 传输协议的通信端口。

- r 忽略普通的 Routing Table，直接将数据包送到远端主机上。
- s 设置本地主机送出数据包的 IP 地址。
- t 设置检测数据包的 TOS 数值。
- v 详细显示指令的执行过程。
- w 设置等待远端主机回报的时间。
- x 开启或关闭数据包的正确性检验。

```

traceroute to www.baidu.com (118.242.68.4), 30 hops max, 60 byte packets
 1 * * *
 2 10.69.203.1 (10.69.203.1) 3.765 ms 10.69.204.49 (10.69.204.49) 4.405 ms 10.67.43.33 (10.67.43.33) 19.
 3 11.69.192.27 (11.69.192.27) 3.909 ms 11.67.104.2 (11.67.104.2) 5.212 ms 11.69.192.19 (11.69.192.19) 4
 4 11.69.192.75 (11.69.192.75) 0.889 ms 11.69.192.79 (11.69.192.79) 0.884 ms 11.69.192.77 (11.69.192.77)
 5 * * *
 6 * * *
 7 * * *
 8 172.16.64.74 (172.16.64.74) 9.520 ms 172.16.64.138 (172.16.64.138) 7.317 ms 7.356 ms
 9 61.49.137.1 (61.49.137.1) 8.056 ms 61.49.137.5 (61.49.137.5) 7.175 ms 8.409 ms
10 125.33.185.121 (125.33.185.121) 8.546 ms 125.33.185.105 (125.33.185.105) 9.280 ms 8.559 ms
11 61.149.203.117 (61.149.203.117) 10.531 ms 125.33.186.65 (125.33.186.65) 8.021 ms 61.149.203.181 (61.14
s
12 * 219.158.9.170 (219.158.9.170) 13.906 ms *
13 110.242.66.182 (110.242.66.182) 18.015 ms 110.242.66.162 (110.242.66.162) 17.211 ms 110.242.66.186 (11
6 ms
^Z
[2]+  Stopped                  traceroute www.baidu.com
[hit2021211041@hecs-5357 ~]$

```

实验 2 shell 程序设计

1.实验目的

掌握在 LINUX 环境下编程开发的基本过程和基本方法，能够用这些编程语言进行简单的程序设计

2.实验内容

- shell 脚本的创建和执行；
- shell 输出重定向和 shell 特殊字符的使用；
- shell 变量的使用方法；
- shell 的控制语句使用；
- shell 脚本的调试；

3.实验题目：

- 学习 shell 特殊字符的使用包括双引号、单引号、倒引号和”\”的作用；
- 学习系统环境变量的使用，包括\$HOME、\$PATH、\$PWD 等；
- 学习用户变量的定义和赋值方法；
- 学习内部变量的作用和使用方法，包括\$#、\$*、\$0、\$?；

以上为学习内容，实验完成下列题目：

- 创建一个 shell 脚本程序：输入圆半径值，输出圆面积的值；
- 编程求解 n 个数的和，要求 n 由键盘输入，使用循环语句完成；
- 利用 for 循环将当前目录下的指定扩展名（由输入决定）文件转移到指定的目录下，并按照文件大小排序，并显示移动后指定目录的内容。

4.实验报告内容

按照实验题目分别给出源程序和运行结果，分析出错原因及解决方案。

- 题目 1 创建一个 shell 脚本程序：输入圆半径值，输出圆面积的值；

1.源程序

```
[hit2021211041@hecs-5357 ~]$ cat>test.sh
#!/bin/bash
PI=3.14159
echo " input radius:"
read r
s=$(echo " $PI*$r*$r " | bc)
echo " square is :$s "
```

2.运行结果

```
[hit2021211041@hecs-5357 ~]$ ./test.sh
input radius:
3
square is :28.27431
[hit2021211041@hecs-5357 ~]$ ./test.sh
input radius:
5
square is :78.53975
[hit2021211041@hecs-5357 ~]$ _
```

3.出错原因及解决方案

- (1) 一开始输出的结果不是数值而是乘法式子，下载 bc 命令并使用得以解决
- (2) 运行程序时，要指明./在当前目录寻找，不然会报错未找到命令。

- 题目 2 编程求解 n 个数的和，要求 n 由键盘输入，使用循环语句完成

1.源程序

```
[hit2021211041@hecs-5357 ~]$ cat>test1.sh
#!/bin/bash
i=0
echo "input n:"
read n
sum=0;
while(( i<=n ))
do
(( sum=sum+i))
(( i=i+1 ))
done
echo "result is :$sum"
```

2.运行结果

```
[hit2021211041@hecs-5357 ~]$ ./test1.sh
input n:
5
result is :15
[hit2021211041@hecs-5357 ~]$ ./test1.sh
input n:
50
result is :1275
[hit2021211041@hecs-5357 ~]$ ./test1.sh
input n:
100
result is :5050
[hit2021211041@hecs-5357 ~]$ _
```

3.出错原因及解决方案

- (1) 注意语句中的括号是否对应
- (2) 注意给用到的变量赋初值

- 题目 3 利用 for 循环将当前目录下的指定扩展名（由输入决定）文件转移到指定的目录下，并按照文件大小排序，并显示移动后指定目录的内容。

1.源程序

```
[hit2021211041@hecs-5357 ~]$ cat>test.sh
#!/bin/bash
echo "input filename extension"
read str
echo "input dir:"
read dir
name=`ls *.$str`
mkdir -p $dir
for file in $name
do
    mv $file $dir/$file
done
ls -S $dir
^C
[hit2021211041@hecs-5357 ~]$ _
```

2.运行结果

```
[hit2021211041@hecs-5357 ~]$ ./test.sh
input filename extension
txt
input dir:
test
test 3.txt 2.txt 1.txt
[hit2021211041@hecs-5357 ~]$ _
```

3.出错原因及解决方案

- (1) 后缀名连接处“.”不要忘了
- (2) 一开始程序一直有错误，检查是 mv 命令中目录写错了