1 Linux 常用命令

在命令行中要输入的内容，或者是终端上显示的内容，会另起一行来说明

1.1 环境设置

1.1.1 调整网络设置

将虚拟机的网络适配器修改为桥接模式（虚拟机-->设置），同时要勾选“复制网络状态”

在 shell 里面使用 ifconfig 查看自己的 IP，如果提示没有此软件，则键入 apt install 命令来安装，输入命令之后要键入自己用户的密码

```
$ sudo apt install net-tools
```

如果使用 ifconfig 命令能够看到自己的 IP，那么接下来就键入 ping www.baidu.com。如果能够成功连接，那么按下 ctrl+c 即可以终止连接

1.1.2 调整时间

选项 --> 详细 --> 时间和日期，将时区调整到上海即可

1.1.3 使用 xshell

xshell 是一款终端模拟软件，它可以使用 ssh 协议连接到远程服务器上。在本课程，我们使用 xshell 连接到我们的虚拟机。在将来的工作中，服务器通常放

置在机房当中，程序员们通常是使用 xshell、putty 等 ssh 软件来连接到服务器，从而能够在服务器上面编程

使用 xshell 之前，需要保证虚拟机和主机能连接上网络

先在虚拟机上 ping 百度，确定虚拟机能连接上网络

然后需要在虚拟机上安装 ssh 协议的客户端

```
$ sudo apt install ssh
```

使用 ps 命令，可以确认 ssh 客户端有没有运行（ps 是显示进程的命令）

```
$ ps -elf|grep ssh
```

使用 apt install 命令来安装 vim 编辑器

```
$ sudo apt install vim
```

键入 vim，再键入 tab 键（注意不要使用回车键！），如果会联想 vim 相关命令，就说明安装成功

配置静态 IP

1. 选择一个合适的静态 ip，不要和其他 Windows 和 Linux 设备的 IP 冲突
2. 如何判断 IP 冲突没有？在 shell 或者是 Windows 下的命令行里面使用 ping 命令测试你想要的 IP 地址，如果能 ping 通，那么就不要使用这个 IP 地址了
3. 打开虚拟机的网络设定，进入 IPV4 分页，选择 manual。修改 IP，网络掩码（255.255.255.0）和网关（这个保持和自动情况下一致即可）
4. 修改 DNS 让它和 Windows 里面的 DNS 一致（例如：202.103.44.150）

5. 关闭网络，然后重新打开，再使用 ifconfig 来检查是否设置成功

定位网络问题

1. ping 本机 IP，如果 ping 不通，则将 IP 修改为 DHCP 后，再重新查看 IP
2. 检查虚拟机是否使用桥接模式
3. 修改虚拟网络编辑器，然后还原默认设置，然后选择一个正确的网络适配器

使用 xshell

1. 打开 xshell，点击“新建”，将名称修改为自己想要的名字，将主机名修改为虚拟机的 IP 地址，点击“连接”
2. 输入账户和密码，并且勾选“记住密码”，就可以连上虚拟机了

1.2 Linux（bash）命令

1.2.1 用户配置

用户是 Linux 的使用者

- Linux 是多用户的系统，一个用户可以多次（同时）登录一个 Linux 系统，多个用户也可以（同时）登录一个 Linux 系统。（与 Windows 桌面版本不一样）
- 使用 Linux 系统就像是入住一间豪宅。豪宅有它的主人，家人和客人；类似地，Linux 系统的用户也分为普通用户和特权用户

- 普通用户有很多个，它们只能使用系统的部分功能，而超级用户只有一个，它的名字是 root，它可以使用系统的所有功能--包括删除另外一个用户
- 对于部分普通用户，它们可以使用 sudo 命令来临时提高自己的权限，从而可以执行一些（而不是所有）特权命令，比如想要使用 apt install 命令在系统上安装软件，使用 shutdown 命令关机等等
- 普通用户的 shell 命令提示符是$，root 的 shell 命令提示符是#

配置密码

Ubuntu 安装的时候，root 用户是没有密码的。所以需要首次进入的账户里面键入如下命令

```
$ sudo passwd root
```

先输入本账户的密码，以获取 sudo 权限（注意输入密码和 Windows 不一样，它不会显示键入了多少字符） 然后输入 root 用户的密码即可

切换用户

切换到根用户，执行 su 命令

```
$ su
```

- 切换到其他用户，在 su 命令后面添加用户名即可。从 root 用户进入其他用户的时候不需要密码

```
$ su 用户名
```

显示所有用户

所有用户的信息是存放在文件/etc/passwd 里面，所以使用查看文件内容的命令 cat 就能显示所有用户的信息

```
$ cat /etc/passwd
```

使用之后会发现打印了大量信息，可以直接看最后面，因为最后面几行才是自定义的用户。打印的信息中每一行代表一个用户，一行里面依次显示的信息是：登录名、加密后的密码、用户 ID、组 ID、注释字段、家目录和 shell 程序的位置

可以看出，root 用户的家目录是/root，而普通用户的家目录是/home/用户名

退出当前用户

使用 exit 指令可以退出当前用户。

```
$ exit
```

用户使用 su 指令会进入其他用户，使用 exit 指令会回到当前的用户

当嵌套使用 su 指令来依次进入多个用户时候，多个用户是使用 **栈结构** 来管理的。执行 su 指令相当于将新用户压入栈顶，执行 exit 指令相当于弹出栈顶

当用户在上述的 **栈结构** 中存在的时候，那个用户是不能够被删除的

添加用户

使用 useradd 命令来添加用户

```
$ useradd 用户名
```

只有 root 用户或者拥有 sudo 权限的用户使用 sudo 命令后才能添加用户

- 什么是拥有 sudo 权限的用户？

如果把 root 看成是宫中的皇帝，root 作为皇帝自然是拥有至高无上的权力，但是这种权力往往也会带来大量的风险（比如下了一个错误的指令，例如删除所有的内容），另一方面，root 也不希望事必躬亲。这个情况下，root 可以将部分权力委托给一些普通用户（类似宫中的太监），这些普通用户只要拿到“圣旨”以后，就能以 root 的身份来处理事情。我们会在将来说明查看一个用户是否拥有 sudo 权限和设置 sudo 权限

在 Linux 系统中，没有消息就是最好的消息。通常命令执行成功时是不会有任何提示的，只有失败的时候才会在 shell 里面打印错误信息

添加用户并指定家目录

给 useradd 命令添加参数，在用户名之前使用-m，可以为用户添加默认家目录（如果不添加家目录，这个用户将无法创建文件）。使用-m 参数的同时还需要使用-s 参数来指定 shell 的位置（如果不添加 shell 的位置，用户的默认 shell 使用的是 sh，它的功能比较弱）

```
$ useradd -m  用户名  -s /bin/bash
```

执行完命令，先使用 pwd 命令获取当前工作目录，使用 cd 命令进入/home 目录，再使用 ls 命令显示当前目录下的所有文件。就会发现 home 的下面新建了一个新的目录，目录的名字就是用户名，这里就是新用户的家目录

```
$ pwd
$ cd /home
$ ls
```

- ~是家目录的简写，它和/home/用户名/是等价的

但是，目前的新用户是没有配置密码的，这也意味我们暂时不能够通过用户+密码的方式来登录用户。使用 passwd 命令可以给新用户配置密码，配置完成以后就能够正常登录了。

```
passwd 用户名
```

删除用户

- 使用 userdel 命令来删除用户

```
userdel 用户名
```

- 如果用户正在使用中，那么这个用户就不能被删除
- 如果用户在 su 命令和 exit 命令构成的用户 **栈结构** 当中的时候，那么这个用户也不能被删除
- 在 userdel 后面添加-r 选项，可以删除用户家目录下的文件

```
$ userdel -r 用户名
```

安全设置

无法直接使用 root 的用户名来登录 root 用户，因为 root 用户的用户名是固定的，它被穷举法找出密码的概率更大

- 服务器可以设置安全措施，例如对连续输入 3 次错误密码的 IP 进行封禁处理
- 我们所使用的 xshell 就使用了 ssh 协议。ssh 就是 secure shell 的简称，相较于其他的不安全的协议（这些协议传递口令都是明文传递的），比如 telnet，tftp 之类的，它提供了一种安全的远程登录协议

在日常工作当中，一般是不会使用 root 用户进行操作的，root 用户拥有非常高的权限，它可以在任意位置去添加和删除文件，一方面不方便工作内容管理，另外很容易误操作带来严重的后果

1.2.2 文件相关命令

文件是存放磁盘上的数据

- 在 Linux 的文件系统是目录和普通文件的一种 **层次结构**。
- 所有东西的起点（也就是 **层次结构** 的根结点）是被称为 **根 root** 的目录，它写作"/"
- 目录是包含目录项的文件（类似于 Windows 的文件夹）。目录项里面有多个文件以及文件的属性
- 文件名是文件或者目录的名字，里面不能有空格或者"/"符号。
- 连续的文件名组成的序列称为路径，根目录开始的路径称为绝对路径，当前目录下的路径称为相对路径

当前工作目录

在 shell 的命令输入栏左边，会提示当前目录的路径。其中，如果路径的最左边是~，那么路径的根是用户的家目录（即/home/用户名）；如果路径的最左边是/，那么路径的根是 root 目录

查看当前工作目录下的所有文件

使用 pwd 命令可以显示当前工作目录

```
$ pwd
```

使用 ls 命令可以显示当前工作目录下的所有文件的文件名

```
$ ls
```

无论是在哪个目录下面使用 ls，都会发现显示结果必有两个目录文件，依次是 . 和 ..

- .代表的是当前目录，.. 代表的是上级目录
- 在 ls 后面可以添加路径名，这样可以打印对应路径下的所有文件的文件名

```
$ ls /
```

与 Windows 不一样，Linux 没有盘和盘符的概念，如果在物理上确实添加了一块磁盘（磁盘上也有文件系统），为了访问磁盘内容，那么 Linux 会将磁盘的文件系统的根目录“添加”（实际上是将磁盘内容映射到 Linux 文件系统里面）到 Linux 根目录下的某个目录上，这个过程称为挂载。使用 mount 命令可以查看当前系统的挂载信息。

```
$ mount
```

- **熟练使用 tab 键的自动联想功能**。当你忘记想要输入的路径的时候，使用 tab 键会给你新的灵感。在某些情况下，如果你输入了路径的前缀，使用 tab 键会自动补全路径

在 ls 命令后面添加-a 参数，可以显示隐藏文件（文件名以.开头的文件）

```
$ ls -a
```

在 ls 命令后面添加-l 参数，可以显示文件的详细信息

```
$ ls -l
```

我们使用两行信息作为例子，来展示详细信息里面的具体内容

```
drwxr-xr-x  2 liao liao 4096 Nov  5 10:01 Public
-rw-r--r--  1 liao liao    0 Nov  5 10:02 .sudo_as_admin_successful
```

- 第一个字母用来说明文件的类型，这种方法是和 Windows 使用后缀来代表文件类型不同的。如果第一字母是 d，那么说明这是一个目录文件，如果是-，那么说明时普通文件（包括文本文件和二进制文件）。文件类型还有很多种，例如：块特殊文件、字符特殊文件、命名管道（FIFO）、套接字（socket）和符号链接等
- 后续的九个字母代表的是权限。依次是代表拥有者的读，写，执行、拥有组的读，写，执行和其他用户的读，写，执行权限。如果文件可读则显示 r，可写则显示 w，可执行则显示 x，没有相关权限则显示-
- 后续的数字代表的是指向这个文件的硬链接数
- 后续的两个字符串依次代表文件拥有者和拥有组
- 随后的数字代表了文件的所占用的空间，这个属性只对普通文件，目录文件和符号链接有意义。普通文件的大小可以是 0。目录的大小是固定的，在本机中是 4096，它和目录下面的文件大小无关，目录文件主要是存放目录下的文件的文件名。符号链接的大小是它指向的文件的路径名长度
- 接着就是一个时间和日期，说明的是最后修改的时间
- 最后的字符串说明了文件名字，如果是符号链接，它还会显示指向文件的名字

在 ls 命令后面添加-lh 参数，可以将文件详细信息以“人类可读”的方式来显示，也就是文件的占用空间会自动根据大小来按照 K，M，G 的单位显示

```
$ ls -lh
```

获取当前的目录

使用 pwd 命令可以获取当前的路径

```
$ pwd
```

改变目录

使用 cd 命令可以改变工作目录

```
$ cd /dev
```

- 在 dev 目录上面使用 ls -l 指令以后，会发现这个目录下有大量的块特殊文件和字符特殊文件。其中块特殊文件用 b 来表示，字符特殊文件用 c 来表示。特殊文件实际上是硬件设备在文件系统上的抽象，所以特殊文件也叫设备文件
- 块特殊文件的特点是每次读取都读取一“块”数据，也就是说会设置一个专门的缓冲区来存放数据，典型的块特殊文件有磁盘。字符特殊文件的特点是每次读取都读取一个字符，典型的有键盘、鼠标、打印机等等

在 cd 后面添加-，可以回到上次 cd 之前的目录（连续使用 cd - 只能在两个目录之间切换，因为之前目录是存放在环境变量里面，使用 env 命令可以查看，每次使用 cd 命令都会修改环境变量）

```
$ cd -
```

- 如何使用 cd 命令，使工作目录转移到父目录，根目录，家目录？

```
$ cd ..
$ cd /
$ cd ~ 或者 cd
```

创建目录

使用 mkdir 可以创建一个目录

```
$ mkdir  目录名
```

如果直接去/home 下面使用 mkdir，它会提示没有权限。能否在一个目录下去添加新的文件，依赖于用户对目录的写权限，因为在目录下添加文件，实际上是修改了目录文件里面的内容。

删除目录

使用 rmdir 命令可以删除一个空目录

```
$ rmdir  目录名
```

- 注意：rmdir 不能删除非空的目录

拷贝文件或者目录

使用 cp 命令可以拷贝文件和目录

```
$ cp [选项]  源文件  目标路径|目标文件

#  当目标是目录的时候，cp 命令会直接在目标目录下拷贝一份副本，文件名和源文件一致

#  当目标是文件的时候，cp 命令会拷贝源文件的内容，重命名为目标文件的名字
```

- Ubuntu 当中，如果 cp 命令中目标路径已经存在同名文件，那么就会直接覆盖（其他发行版中在选项中添加-f(force 强制)也可以实现同样的效果）

如果希望在覆盖之间进行警告，则应该在选项中添加-i(interactive 交互式)。比如，如果 dir1/text1 已经存在，则键入下面的命令：

```
$ cp -i text1 dir1
```

```
# 那么就会出现提示：

cp: overwrite 'dir1/text1'?

# 如果输入 n 或者 no 就会不覆盖，输入其他则会覆盖
```

如果源文件是一个目录，那么在选项中必须要添加-r，这里 r 是 recursive（循环递归）的意思，也就是说添加-r 以后，cp 命令会递归地将目录下的所有文件都进行拷贝

```
$ cp -r dir1 path

# 执行完毕后会出现 dir1 的副本 path/dir1
```

移动文件或者目录

使用 mv 命令可以移动文件或者目录

```
$ mv [选项] 源文件 目标路径|目标文件
```

- 执行 mv 命令的效果和 Windows 下的剪切-->粘贴的效果一样
- 和 cp 命令一样，mv 命令会出现覆盖现存文件的情况，mv 也可以使用-f 或者-i 选项来关闭或者开启提醒
- Linux 里面没有专门的重命名命令，使用 mv 命令移动到原来的目录下面就可以实现重命名了

删除文件

使用 rm 命令可以删除文件

```
$ rm [选项] 文件|目录
```

- 使用 rm，cp 和 mv 导致文件被覆盖或者删除都是 **不可恢复** 的！

- 不要以 root 的身份或者是 sudo 权限来执行 rm -rf，这会造成不可恢复的后果
- 和 rm 以及 cp 命令一样，rm 也可以使用-f 选项来实现强制删除，使用-i 选项来实现删除前交互提示，使用-r 选项来实现递归删除目录及目录下文件

树状目录结构显示

首次使用之前需要先安装 tree

```
$ sudo apt install tree
```

使用 tree 命令就可以显示目录的树状结构

```
$ tree  路径名
```

- 如果对 tree 命令的详细命令不了解，可以在安装完 tree 以后再使用 man 命令查看命令帮助，下面是使用 tree 以后的输出结果

```
.
├── dir1
│   ├── text1
│   └── text3
├── dir2
│   ├── text1
│   └── text2
├── text1
└── text2
```

显然，tree 命令可以将目录的树形结构用图形化的方式打印出来

在 tree 命令之后添加-h 选项，可以增加文件大小输出

1.2.3 权限相关

修改权限

- 文件的权限有三种，依次是读、写和执行
- 从命令 ls -l 的打印结果中可以得知，这种三种权限还会根据用户的身份而有所不同。文件的拥有者、文件拥有者的同组其他成员以及其他用户对文件的权限会有所区别

1. 如何判断某个用户能否读取/修改/执行文件？

> 先判断当前用户和文件之间的身份关系，然后再检查文件的权限位

2. 如何判断某个用户能否删除文件？

> 先找到文件所在目录，判断当前用户是否拥有目录的写权限

使用 chmod 可以修改文件的权限，chmod 命令比较灵活，它拥有比较多种使用模式，接下来会依次介绍：

1. 文字设定法

```
$ chmod [who][+|-|=][mode]  文件名

# who 代表用户类型：u 表示用户，g 表示组成员，o 表示其他用户，a 或者不写代表所有用户（ugo 三个字母可以组合）

# +|-|=代表操作符：+表示添加某个权限，-表示取消某个权限，=表示设置成某种权限

#mode 代表模式，使用 rwx 三个字母的组合，r 表示可读，w 表示可写，x 表示可执行
```

可以在一行语句中对某个文件调整多次权限，使用逗号来分隔用户操作类型即可，例如：

```
$ chmod g+r,o+r example
```

执行完成以后，增加了组成员和其他用户的读权限

另外一个例子，删除 example 文件所有用户的执行权限（三种写法等价）

```
$ chmod a-x example

$ chmod -x example

$ chmod ugo-x example
```

2. 数字设定法

- 通常文件的权限是采用类似于位图的形式来进行存储，3 位的 2 进制数字可以用来表示文件的读写执行权限，9 位的 2 进制数字就可以表示文件的所有用户权限。
- 在 3 位 2 进制数中，通常使用最高位表示读权限，随后是写权限，最低位是执行权限。然后再将拥有用户，拥有组和其他用户组合起来，就能得到一个 9 位 2 进制数字，最后按照 8 进制和 2 进制的转换方法，文件权限就能很轻松地一个用 3 位 8 进制数字来表示
- 例如 **0644** 就是 **拥有者可读可写不可执行，组用户和其他用户可读不可写不可执行**

```
$ chmod 644 example
```

执行完成以后就会将权限设置成拥有者可读可写不可执行，组用户和其他用户可读不可写不可执行

```
$chmod 775 example
```

执行完成以后就会将权限设置成拥有者和组用户可读可写可执行，其他用户可读不可写可执行（工作中经常使用）

1.2.4 文件查找

索引

- 在 Windows 的旧版本当中，想要搜索一个文件是非常缓慢的。为了实现快速查找，在 Windows 10 和 Linux 系统当中，系统会专门维护一个索引文件
- 在 Linux 当中，索引文件是存放在具体的磁盘文件前面的，索引文件里面存放了各个文件的位置信息和其他信息。有关使用索引来实现快速查找的详细算法将会在数据库阶段进行讲解

使用 find 命令可以根据查找条件来在起始目录之下来查找文件，然后再执行指定的操作

```
$ find  起始目录  查找条件  操作
```

使用 find 命令可以用来查找某个头文件的位置

```
$ sudo apt install gcc

#先安装 gcc 编译器

$ find /usr/include -name stdio.h

#  在/usr/include 目录下面查找文件 stdio.h
```

常见的查找条件

根据名称和文件属性查找

- -name '字串' 查找文件名匹配所给字串的所有文件，字串内可用通配符 *、?、[ ]
- -gid n 查找属于 ID 号为 n 的用户组的所有文件

- -uid n 查找属于 ID 号为 n 的用户的所有文件

```
$ find . -uid 1000
```

- -group '字串' 查找属于用户组名为所给字串的所有的文件
- -user '字串' 查找属于用户名为所给字串的所有的文件

```
$ find . -user liao
# 这两个查找是等价的
```

- -empty 查找大小为 0 的目录或文件。
- -perm 权限 查找具有指定权限的文件和目录，权限的表示可以如 711，644
- -size n[bckw] 查找指定文件大小的文件，n 后面的字符表示单位，缺省为 b，代表 512 字节的块
- -type x 查找类型为 x 的文件

根据时间查找

- amin n 查找 n 分钟以前被访问过的所有文件（+表示 n 分钟之前，-表示 n 分钟之内，+号和-号都不能省略）
- cmin n 查找 n 分钟以前文件状态被修改过的所有文件
- mmin n 查找 n 分钟以前文件内容被修改过的所有文件
- atime n 查找 n 天以前被访问过的所有文件
- ctime n 查找 n 天以前文件状态被修改过的所有文件
- mtime n 查找 n 天以前文件内容被修改过的所有文件

- 其他的查找条件可以在 man 帮助里面找到

应用通配符

通配符是字符串里面的特殊符号。如果将字符串里面的通配符按照一定的规则用字符来替换以后，得到的字符串和目标字符串一致，那么目标字符串就能匹配 例子，当前目录里面有文件 file，file1，file12，file2，file3，file4

```
$ find . -name "file*"

# *用来匹配 0 至多个任意字符
```

结果是

```
./file1

./file

./file12

./file2

./file3

./file4

$ find . -name "file?"

# ?用来匹配 1 个任意字符
```

结果是

```
./file1

./file2

./file3

./file4

$ find . -name "file[0-9]"

# [范围] 用来匹配 1 个范围内的字符
```

结果是

```
./file1
./file2
./file3
./file4
```

查找条件的运算

查找条件可以进行逻辑运算，比如逻辑与，逻辑或和逻辑非

1. 在命令中用-a 表示逻辑与，表示只有当所给的条件都满足时，查找条件才满足。例如在/home/user 目录下查找名为 0718 类型是一个目录的文件（使用频率最高）：

```
$ find  /home/user –name  0718  –a  -type  d
```

2. 在命令中用-o 表示逻辑或，表示只要所给的条件有一个满足，查找条件就满足。例如在/home/user 目录下查找名字为 main.cc 或名字为 main.c 的文件：

```
$ find  /home/user  -name main.cc  -o  -name main.c
```

3. 在命令中用！表示逻辑非，表示查找不满足所给条件的文件。例如在/home/user 下查找名字不是 main.c 的文件：

```
$ find  /home/user  !  –name main.c
```

查找的权限

必须要拥有某个目录的执行权限，才能够进入该目录进行查找

组合操作

在 find 命令后面添加-exec 新命令 \;就可以对每个查找出来的文件执行新命令

```
$ find  /usr/include –name  stdio.h  -exec ls –l  {}  \;
```

工作中一般使用 xargs，通常可以使用|将 find 命令的执行结果输出到一个缓冲区（而不是显示屏）里面，这个缓冲区称为管道。然后 xargs 可以从管道中按行读取内容，并且循环执行命令

```
$ find  /usr/include –name  stdio.h  main.c|xargs ls –l
```

如果对于管道里面的内容，不需要按行循环执行命令，那么就不需要使用 xargs，直接使用管道即可

```
$ ls|wc -l
#  列出所有文件然后统计行数
```

1.2.5 磁盘相关的命令

文件系统的整体磁盘空间使用情况

使用 df 命令可以列出文件系统的整体磁盘空间使用情况

```
$ df [选项] [文件名]
$ df –h
#  显示整个磁盘使用情况
```

显示每个文件和目录的磁盘使用空间

使用 du 命令可以显示每个文件和目录的磁盘使用空间 显示目录的每一级

```
$ du [选项] [文件名]
```

只显示当前目录

```
$ du -h --max-depth=0 /home/liao
```

1.2.6 文件查看和处理

查看文件内容

使用 cat 命令可以查看文件内容

```
$ cat [选项] [文件]
```

一些选项参数

- -b 对非空输出行编号
- -E 在每行结束处显示$
- -n 对输出的所有行编号
- -s 不输出多行空行 查看密码文件

```
$ cat etc/shadow
# 如果出现拒绝访问，是什么原因？
```

shadow 文件里面存放了盐值以及通过 sha512 算法得到的密文

标准的输入输出与重定向

文件描述符是一个整数，它代表一个打开的文件 标准的三个描述符号

1. 标准输入：一般指键盘，描述符为：0

2. 标准输出：一般指屏幕输出，描述符为：1

3. 错误输出：也是屏幕，描述符为：2 重定向符号包括重定向输入<，重定向输出>，添加输出>>，错误重定向 2>，错误和信息重定向&>

- 可以尝试一下同时将信息打印到 stdout 和 stderr，然后使用重定向符号进行调试
- 注意程序的信息要么只能重定向到 stdout，要么只能重定向到 stderr，不能分开重定向

cat 常常与重定向一起使用。其中>表示创建，>>表示追加，<<表示以什么结束，如果 cat 的命令行中没有参数，它就会从标准输入中读取数据，并将其送到标准输出

创建文件

Linux 中创建空文件的四种方式：

```
# 方式 1，使用 echo：
$ echo > a.txt（会有一个字节）
$ echo -n > a.txt
# 方式 2，使用 touch：
$ touch b.txt
# 方式 3，使用 cat：
$ cat > c.txt
$ 按 ctrl+c 或 ctrl+d 退出
# 方式 4： vi d.txt
$ 进入之后按：wq 退出
```

部分显示

使用 head 命令可以显示文件的前面几行

```
$ head [–n  行数值] [文件名]
```

使用 tail 命令可以显示文件的后面几行

```
$ tail [–n  行数值] [文件名]
```

提问：怎么样将历史记录的后 10 行存入文件 latestHistory.txt 里面？ 使用 C 语言怎么做？打开~/.bash_history 文件，然后记录最后 10 行，然后再输出 使用 shell 怎么做？history|tail -n 10> latestHistory.txt 由此可见，shell 程序的编写效率要远远高于 C 语言，这也是 shell 出现的目的

使用 more 或者 less 命令来单页浏览文件

```
$ more  文件名
#  进入以后使用 f 和 b 向前向后翻页，使用 q 退出浏览
```

其他操作

使用 sort 命令可以对文件内容进行排序

```
$ sort  文件名
```

- 提问：怎么将结果进行保存？

使用 file 命令查看文件内容类型

```
$ file  文件名
```

```
# 根据文件内容，判别文件类型：比如可执行文件、文本等等
```

使用 uniq 命令报告或删除文件中重复的行（只能去除相邻相同的）

```
$ uniq 文件名
```

- -c 在输出行前面加上每行在输入文件中出现的次数。
- -d 仅显示重复行。
- -u 仅显示不重复的行。

使用 wc 命令统计指定文件中的字节数、字数、行数

```
$ wc 文件名
```

- -c 统计字节数
- -l 统计行数
- 提问：统计文件数

```
# 统计当前目录下的文件
$ ls|wc -l
# 统计当前目录以及所有的子目录下的文件
$ find . -name *|wc -l
# -m 统计字符数。这个标志不能与 -c 标志一起使用。
# -w 统计字数。一个字被定义为由空白、跳格或换行字符分隔的字符串
```

使用 iconv 命令汉字编码转换

```
$ iconv [-f 原编码方式] [-t 新的编码方式] 文件命令
```

Windows 默认是采用 GBK（gb2312）编码，在 Windows 里面保存一个内容为中文“烫”的文件（它的二进制形式是 0xcccc），再将其放入 Linux 当中，如果直接使用 cat 打印是不会显示内容的

使用下列的 iconv 命令转换语句以后，将结果重定向到新文件里面，再使用 cat 打印新文件就能显示正确的文本信息

```
$ iconv -f gb2312 -t utf-8 hanzi.txt>hanzi1
```

utf-8 编码使用最多 3 个字节来表示符号，它可以表示世界上所有语言的所有符号，而 GB2312 是中国的标准，它只能使用最多 2 个字节来表示汉字 在 vim 编辑器里面，输入:%!xxd 可以令文本按 16 进制显示

1.2.7 文件内容查找

搜索文件内容

使用 grep 命令可以查找文件内容

```
$ grep [选项][查找模式][文件名 1，文件名 2，...]
```

- grep 过滤器查找指定字符模式的文件，并显示含有此模式的所有行，符合规则的所有模式会显示为红色。被寻找的模式称为正则表达式
- 选项中使用-E，可以使用拓展正则表达式规则，否则默认采用基本规则
- 其他参数：

```
-F 每个模式作为固定的字符串对待

-c 只显示匹配行的数量

-i 比较式不区分大小写
```

```
-n 在输出前加上匹配串所在的行号
```

正则表达式

和 find 命令使用通配符有所区别，grep 使用正则表达式的方式来进行模式匹配。正则表达式是利用通配符（特殊字符）来定义了规则，从而和文本进行匹配。grep 命令的目标就是找到符合规则的字符串集合

正则表达式语法:

1. 单个字符是匹配的基本单位，比如 a 就能够匹配含有字母的字符串集合。几乎所有的字母和数字都能够匹配它们本身
2. 反斜杠+某些字符构成转义字符，它们作为整体可以看成是一个特殊的单个字符
3. 大部分的通配符规则:

```
a|b                 匹配 a 或 b 使用-E
gr(a|e)y 匹配 gray 或 grey
.                   匹配任一字符
[abc]       匹配任一字符:a 或 b 或 c
[^abc]      匹配任一字符，但不包括 a,b,c
[a-z]       匹配从 a 到 z 之间的任一字符
[a-zA-Z] 匹配从 a 到 z，及从 A 到 Z 之间的任一字符
^表达式     匹配字符串的头部 使用-E
表达式$     匹配字符串的尾部 使用-E
( )                 匹配标记的子表达式 使用-E
\b                  匹配字词边界
\n                  匹配第 n 个标记的子表达式，n 代表 1 到 9
*                   匹配前一项内容 0 或多次
```

?	匹配前一项内容 0 或 1 次 使用-E
+	匹配前一项内容 1 或多次 使用-E
{x}	匹配前一项内容 x 次
{x,}	匹配前一项内容最少 x 次
{,x}	匹配前一项内容最多 x 次
{x,y}	匹配前一项内容次数介于 x 和 y 之间 使用-E
\	转义字符

grep 应用

grep 的运行原理是打开文件，然后按行来读取文件内容，如果本行里面拥有匹配模板的字符串，那么就染色这个字符串，并打印输出

- 在 C 程序文件里面找到 int 开始的语句

```
$ grep -n ^int main.c
```

- 在 C 程序文件里面找到;结尾的语句

```
$ grep -n \;$ main.c

$ grep -n ";$" main.c
```

- 提问：为什么不能使用 grep -n ;$ main.c
- 提问：怎么样匹配任意多个字符？
- 怎么样查找特殊字符？

```
$ grep -nF ^ main.c
```

在 C 程序文件里面找到 main 函数

```
$ find . -name "*.c"|xargs grep -n "main"
```

1.2.8 其他命令相关

查看命令的别名

使用 alias 命令可以打印当前所使用的别名情况。一旦给命令起了别名，那么输入命令的别名就可以起到和原来命令一样的效果。

```
$ alias
```

下面的例子是使用 alias 命令以后，终端打印的部分信息：

```
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
```

- 根据别名信息可以得知，当我们键入 l 的时候，等价于输入 ls -CF（目录名后增加/符号显示）；当我们键入 ll，等价于键入 ls -alF

输入命令的历史记录

使用 history 命令可以获取输入命令的历史记录，在 history 命令后面再添加>文本文件，就可以将历史记录保存到文本文件（本例子中是 myHistory.txt）里面

```
$ history > myHistory.txt
```

查看命令的帮助

Linux 系统中会内置帮助手册，帮助手册有 8 个部分，包括用户手册、程序员手册和系统管理员手册等等。通常通过关键字+数字的形式来说明是第几个手册的帮助信息。例如，ls(1)就说明是第一部分（用户手册，即 shell 命令手册）的 ls 命令的信息

使用 man 命令可以查看帮助信息

```
$ man [手册编号]  命令名字
```

- 进入帮助界面以后，使用 q 可以退出界面，使用方向键可以向前向后查看，使用 b 和 f 可以前后翻页，使用/+字符串可以进行查找

一次输入以后执行多条命令

输入完命令以后不键入回车键，而是键入分号，然后再输入下一条命令，就可以实现按顺序依次执行多条命令的效果。例如先创建目录，再展示所有文件

```
$ mkdir dir1;ls -l

#  第二条的 ls 命令的结果当中就会出现 dir1 的信息
```

管道和命令替换

管道是重定向的一种，就像一个导管一样，将一个程序或命令的输出作为另一个程序或命令的输入

```
$ ls -l /etc|wc -w
```

- 提问： 这个命令的结果和 ls /etc|wc -w 的结果有什么区别？

命令替换和重定向有点相似，但区别在于命令替换是将一个命令的输出作为另一个命令的参数

常用的格式为

```
$ command1 `command2`
# command1  是任意命令
#  这里不是单引号，是键盘 1 左边的按键
$ command1 $(command2)
```

首先列出当前的所有信息，并重定向到 aa 文件中：

```
$ ls|cat>aa
#或
$ ls>aa
```

然后，通过命令替换，列出 aa 文件中所有的文件信息

```
$ ls –l `cat aa`
#  或者用
$ ls –l $(cat aa)
```

文件或目录的创建掩码

umask 指文件（默认是 0666）或目录（默认 0777）创建时在全部权限中要去掉的一些权限，普通用户缺省时 umask 的值为 002，root 用户为 022

- 提问：请根据实际创建的文件的权限回答，掩码是怎么运算的？

  以通过 umask 查看默认的缺省的掩码值，通过 umask 001 来修改掩码值

```
$ umask
$ umask 001
```

- 这种操作是临时的，如果要永久修改，需要修改配置文件（例如~/.bashrc）

打包和压缩

Linux 当中是可以把一大堆的文件和目录全部打包成一个文件，这对于备份文件或将几个文件组合成为一个文件以便于网络传输是非常有用的。使用 tar 命令大小可以打包，可以将文件加入到某个包文件或者是将文件移除出某个包文件

```
$ tar [主选项+辅选项] 目标文档 源文件或目录
```

- 参数说明

```
# c：创建新的包

$ tar cfv packet.tar file*
```

包文件通常是比较庞大的，具体原因是包文件创建默认分配了较大的磁盘空间，所以需要压缩来解决磁盘空间

```
# r：要把存档的文件追加到包文件的末尾。

$ tar rfv packet.tar test

# x：从包文件中释放文件

$ tar xfv packet.tar

# f：使用包文件或设备。

# v：在归档过程中显示处理的文件。

# z：用 gzip 来压缩/解压缩文件，后缀名为.gz，加上该选项后可以将档案文件进行压缩。

$ tar  czvf packet.tar.gz  file* *.txt
```

- 这时候检查压缩文件的大小，它的大小实际上比原文件还要小

  解压释放文件的命令

```
$ tar  xzvf  packet.tar.gz
```

- 一种典型的压缩方法是哈夫曼编码树的方法

使用 gzip/bzip2 命令来文件压缩与解压，gzip 用来将文件压缩成后缀为.gz 的压缩文件，或者将后缀为.gz 的文件进行解压。Bzip2 用来将文件压缩成后缀名为.bz2 的压缩文件，或者将后缀为.bz2 的压缩文件解压

```
$ gzip/bzip2  [选项] [压缩或解压缩的文件名]
```

常用参数：

- -d: 将压缩文件进行解压。
- -v： 在压缩或解压过程中显示解压或压缩的文件。 例如，将 main.c 进行压缩，则使用

```
$ gzip/bzip2 –v  main.c。
```

- 会将 main.c 压缩成 main.c.gz 或者 main.c.bz2 如果需要将刚才的压缩文件解压，则使用

```
$ gzip  –dv  main.c.gz,或者 bzip2 –dv  main.c.bz2
```

远程拷贝

远程拷贝之前，需要先 ping 通目标服务器

```
$ ping 112.124.31.45
```

使用 scp 命令从本机拷贝到远程服务器

```
$ scp /home/liao/pre/file1 liaozs@112.124.31.45:~/
```

如果是首次连接，还需要使用一个授权（输入 yes 即可）。然后输入密码

使用 scp 命令从远程服务器拷贝到本机

```
$ scp liaozs@112.124.31.45:~/file1 ~
```

使用脚本自动将文件在本地和服务器之间交换 上传文件：

```
scp -r $1 [用户名]@[目标 IP]:[路径]`date -d "0 week" +%Y%m%d`/code
```

$1 是命令行的参数，类似 C 语言的 agrv[1]

下载文件：

```
scp -r [用户名]@[目标 IP]:[路径]`date -d "0 week" +%Y%m%d`/code .
```

使用的时候将上述命令保存到文件，添加执行权限，使用 sudo 放在/bin 下面

无秘钥登录可以实现不使用密码登录

每个用户的秘钥存放在~/.ssh/里面，其中 id_rsa 是私钥，id_rsa.pub 是公钥

使用 ssh-keygen 来生成钥匙（一直回车即可），让后会生成一对钥匙（这里是采用了非对称加密的算法，有兴趣的同学可以课后自行查阅相关知识），然后将公钥的内容拷贝到目标机器的.ssh/authorized_keys 文件里面（如果没有必需要自己创建）

1. 使用 scp 命令拷贝目标主机
2. 登陆到目标主机（ssh 用户名@ip 地址）
3. 使用 cat 打开公钥，然后使用>>追加到授权文件里面

- 注意使用无秘钥登录是单向的，并且本机的私钥不能移动位置或者修改

使用 ssh 命令可以远程执行命令

```
$ ssh 用户名@ip 地址 “命令”
```

1.3 杂项

WinScp 软件的使用

工作当中经常需要在服务器和本地之间交换文件，使用 WinScp 软件可以高效地在不同的设备当中交换文件

打开 WinScp 软件，然后登录服务器。然后在本地和服务器之间的文件系统当中拖动文件就可以便捷地实现文件上传和下载

xshell 软件设置调整

修改 xshell 设置实现选中复制，右击粘贴

- 进入工具-->选项-->键盘和鼠标，勾选将选中的文本自动复制到剪切板，将鼠标中键的功能修改为打开弹出式菜单，将右键的功能设置为粘贴剪切板内容 修改 xshell 设置，实现退格键正常退格
- 进入文件-->打开，右键点击当前会话，进入属性-->终端-->键盘，将 delete 和 backspace 都修改为 ASCII，然后重启会话。

命令输入的常用快捷键

- **tab** 输入预测，根据前缀自动补全路径名
- **ctrl+a** 光标回到命令输入区开始位置
- **ctrl+e** 光标回到命令输入区结束位置
- **ctrl+r** 搜索之前出现过的命令
- **方向键上** 键入上一条命令

安装帮助手册

Ubuntu 系统中的帮助手册默认只有 1 部分（命令帮助），其他部分需要自行安装 安装 POSIX 帮助

```
$ sudo apt install manpages-posix-dev
```

安装完成以后，使用 man 命令就可以查看 POSIX 接口和 ISO C 接口的帮助了

网络相关的命令

当网络不通时，通过执行 route 命令查看路由，查看网关配置是否正常

```
$ route
```