

参考内容：

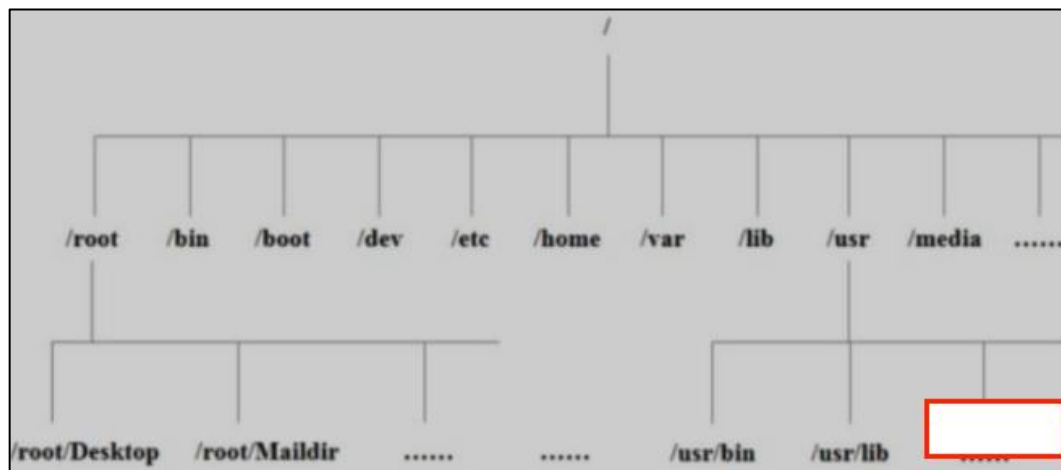
- https://mp.weixin.qq.com/s/RT93qJdTagtKjWKx_A_6Nw
- <https://mp.weixin.qq.com/s/4dZ3mbu287pC6mdGOczGAA>
- <https://mp.weixin.qq.com/s/AndIwanhZ3Tlg0iQe8a4lQ>
- <https://mp.weixin.qq.com/s/aMJeIQW5z5FzeEdsRn44vA>
- <https://github.com/Snailclimb/JavaGuide/blob/main/docs/cs-basics/operating-system/linux-intro.md>

一、了解 Linux

1、系统操作

sync	将系统数据由内存同步到硬盘中
shutdown -h 10	10 分钟后关机
shutdown -h +10	10 分钟后关机
shutdown -h now	立马关机
halt	立马关机
poweroff	立马关机
reboot	系统重启
shutdown -r +10	10 分钟后重启

2、系统目录结构



- /bin: bin 是 Binary 的缩写，这个目录存放着最经常使用的命令；
- /boot: 这里存放的是启动 Linux 时使用的一些核心文件，包括一些连接文件以及镜像文件；
- /dev: dev 是 Device(设备)的缩写，存放的是 Linux 的外部设备，在 Linux 中访问设备的方式和访问文件的方式是相同的；
- /etc: 这个目录用来存放所有的系统管理所需要的配置文件和子目录；

- **/home:** 用户的主目录，在 Linux 中，每个用户都有一个自己的目录，一般该目录名是以用户的账号命名的；
- **/lib:** 这个目录里存放着系统最基本的动态连接共享库，起作用类似于 Windows 里的 dll 文件；
- **/lost+found:** 这个目录一般情况下是空的，当系统非法关机后，这里就存放了一些文件；
- **/media:** linux 系统会自动识别一些设备，例如 U 盘、光驱等，当识别后，linux 会把识别的设备挂载到这个目录下；
- **/mnt:** 系统提供该目录是为了让用户临时挂载别的文件系统的，我们可以将光驱挂载在 /mnt/ 上，然后进入该目录就可以查看光驱里的内容了；
- **/opt:** 这是给主机额外安装软件所摆放的目录。比如你安装一个 ORACLE 数据库就可以放到这个目录下。默认是空的；
- **/proc:** 这个目录是一个虚拟的目录，它是系统内存的映射，我们可以通过直接访问这个目录来获取系统信息；
- **/root:** 该目录为系统管理员，也称作超级权限者的用户主目录；
- **/sbin:** s 就是 Super User 的意思，这里存放的是系统管理员使用的系统管理程序；
- **/srv:** 该目录存放一些服务启动之后需要提取的数据；
- **/sys:** 这是 linux2.6 内核的一个很大的变化。该目录下安装了 2.6 内核中新出现的一个文件系统 sysfs；
- **/tmp:** 这个目录是用来存放一些临时文件的；
- **/usr:** 这是一个非常重要的目录，用户的很多应用程序和文件都放在这个目录下，类似于 windows 下的 program files 目录；
- **/usr/bin:** 系统用户使用的应用程序；
- **/usr/sbin:** 超级用户使用的比较高级的管理程序和系统守护程序；
- **/usr/src:** 内核源代码默认的放置目录；
- **/var:** 这个目录中存放着在不断扩充着的东西，我们习惯将那些经常被修改的目录放在这个目录下。包括各种日志文件；
- **/run:** 是一个临时文件系统，存储系统启动以来的信息。当系统重启时，这个目录下的文件应该被删掉或清楚；

二、常用的基本命令

1、目录管理

- Linux 的目录结构为树状结构，最顶级的目录为根目录 / ；
- 其它目录通过挂载可以将它们添加到树中，通过解除挂载可以移除它们；

ls -al

- 将目录下的所有文件列出来(含属性与隐藏档)
- **-a:** 全部的文件，连同隐藏文件(开头为.的文件)一起列出来；
- **-l:** 长数据串列出，包含文件的属性与权限等等数据；

cd

- Change Directory 的缩写，变换工作目录；

pwd

- Print Working Directory 的缩写，显示目前所在目录；
- -P: 显示出确实的路径，而非使用连接(link)路径；

mkdir -mp 711 可多级目录

- -m: 配置目录的权限；
- -p: 递归创建(包含上级目录)；

rmdir -p 可多级空目录

- 删除空目录；
- -p: 连同上级空目录一起删除；

cp -ai 来源档(文件或目录) 目标档(文件或目录)

cp -pdri source1 source2 ... directory

- -a: 相当于-pdr 的意思，参考以下；
- -p: 连同文件的属性一起复制过去，而非使用默认属性(备份常用)；
- -d: 若来源档为连接档的属性(link file)，则复制连接档属性而非文件本身；
- -r: 递归持续复制，用于目录的复制行为；
- -f: 强制，若目标文件已经存在且无法开启，则移除后再尝试一次；
- -i: 若目标档已经存在时，在覆盖时会先询问动作的进行；
- -l: 进行硬式连接的连接档创建，而非复制文件本身；
- -s: 复制称为符号连接档，亦即捷径文件；
- -u: 若目标档比来源档旧才升级目标档；

rm -fir 文件或目录

- -f: 强制，忽略不存在的文件，不会出现警告信息；
- -i: 互动模式，在删除前会询问使用者是否动作；
- -r: 递归删除，最常用在目录的删除，**非常危险的选项**；

mv -fiu 来源档(文件或目录) 目标档(文件或目录)

- 或修改名称;
- -f: 强制, 如果目标文件已经存在, 不会询问而直接覆盖;
- -i: 若目标文件已经存在时, 询问是否覆盖;
- -u: 若目标文件已经存在, 且 source 比较新, 才会 update;

2、基本属性

2.1、文件属性介绍

```
[[root@RyuAlvin ~]# ls -al
总用量 52
dr-xr-x---.  4 root root 4096 2月 27 18:55 .
dr-xr-xr-x. 18 root root 4096 2月 24 07:42 ..
-rw-----  1 root root 3154 2月 27 23:20 .bash_history
-rw-r--r--  1 root root  18 12月 29 2013 .bash_logout
-rw-r--r--  1 root root 176 12月 29 2013 .bash_profile
-rw-r--r--  1 root root 176 12月 29 2013 .bashrc
-rw-r--r--  1 root root 100 12月 29 2013 .cshrc
-rw-----  1 root root  35 2月 27 09:18 .lessht
drwxr-xr-x  2 root root 4096 11月 30 07:26 .pip
-rw-r--r--  1 root root 206 2月 18 08:29 .pydistutils.cfg
drwx-----  2 root root 4096 4月 26 2020 .ssh
-rw-r--r--  1 root root 129 12月 29 2013 .tcshrc
-rw-----  1 root root 1422 2月 27 18:55 .viminfo
```

- 在 Linux 中第一个字符代表这个文件是目录、文件或链接文件等等;
 - 当为[d]则是目录;
 - 当为[-]则是文件;
 - 若是[l]则表示为链接文档 (link file) ;
 - 若是[b]则表示为装置文件里面的可供储存的接口设备 (可随机存取装置);
 - 若是[c]则表示为装置文件里面的串行端口设备, 例如键盘、鼠标 (一次性读取装置);
- 接下来的字符中, 以三个为一组, 且均为【rwx】的三个参数的组合;
 - [r]代表可读;
 - [w]代表可写;
 - [x]代表可执行;
 - 如果没有权限, 就会出现减号[-];

文件 类型	属主 权限			属组 权限			其他用户 权限		
0	1	2	3	4	5	6	7	8	9
d	rwX			r-X			r-X		
目录 文件	读	写	执行	读	写	执行	读	写	执行

2.2、修改文件属性

chgrp [-R] 属组名 文件名

- 更改文件属组；
- -R: 递归更改文件属组，就是在更改某个目录文件的属组时，如果加上-R 的参数，那么该目录下的所有文件的属组都会更改；

chown [-R] 属主名 文件名

chown [-R] 属主名: 属组名 文件名

- 更改文件属主，也可以同时更改文件属组；

chmod [-R] 770 filename

2.3、文件操作

man [命令]

命令 -help

- 查看 各个命令的使用文档；

touch info.txt 创建空文件

ls -l > info.txt 输出，覆盖

ls -l >> info.txt 输出，追加

echo "Hello World" > info.txt

echo "Hello World" 在控制台输出 Hello World

ln -s /root/info.txt sLink 创建/root/info.txt 的软链接，inode 节点不同

ln -s /root/info.txt Link 创建/root/info.txt 的硬链接，inode 节点相同

```
cat /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
cat /etc/profile | grep -n profile 查找单词 profile 并且显示行号
```

```
cat /etc/profile | grep -ni profile 查找单词 profile, 不区分大小写并显示行号
```

- 查看网络配置：文件地址/etc/sysconfig/network-scripts/ifcfg-eth0
- cat 由第一行开始显示文件内容
- -A: 相当于-vET 的整合选项，可列出一些特殊字符而不是空白而已；
- -b: 列出行号，仅针对非空白行做行号显示，空白行不标行号；
- -E: 将结尾的断行字节¥显示出来；
- -n: 列印出行号，连同空白行也会有行号，与-b 的选项不同；
- -T: 将 tab 按键以^I显示出来；
- -v: 列出一些看不出来的特殊字符；

```
tac /etc/sysconfig/network-scripts/ifcfg-eth0
```

- tac 与 cat 命令刚好相反，文件内容从最后一行开始显示，可以看出 tac 是 cat 倒写；

```
nl /etc/sysconfig/network-scripts/ifcfg-eth0
```

- 显示行号；
- -b: 指定行号指定的方式，主要有两种：
 - -b a: 表示不论是否为空行，也同样列出行号（类似 cat -n）；
 - -b t: 如果有空行，空的那一行不要列出行号（默认值）；
- -n: 列出行号表示的方法，主要有三种：
 - -n ln: 行号在荧幕的最左方显示；
 - -n rn: 行号在自己栏位的最右方显示，且不加 0；
 - -n rz: 行号在自己栏位的最右方显示，且加 0；
- -w: 行号栏位的占用的位数；

```
more /etc/csh.login
```

- more 是一个基于 VI 编辑器的文本过滤器，以全屏的方式按页显示文本文件的内容；
- 一页一页翻动；
- 空白键：向下翻一页；
- Enter: 向下翻一行；
- /字符串：向下搜寻这个字符串；
- :f: 立刻显示出档名以及目前显示的行数；
- q: 代表立刻离开 more，不再显示该文件内容；
- b 或[ctrl]-b: 往回翻页，不过这动作只对文件有用，对管线无用；

```
less /etc/man.config
```

- 用来分屏查看文件内容，功能比 more 强大，支持各种显示终端。在显示文件内容时，并不是一次加载，而是根据需求时才被加载，显示大型文件具有比较高的效率；
- 空白键：向下翻动一页；
- pagedown: 向下翻动一页；

- `pageup`: 向上翻动一页;
- `/`字符串: 向下搜寻这个字符串;
- `?`字符串: 向上搜寻这个字符串;
- `n`: 重复前一个搜寻 (与/或?有关);
- `N`: 反向的重复前一个搜寻 (与/或?有关);
- `q`: 离开 `less` 这个程序;

```
head -n 20 /etc/csh.login
```

- 取出文件前面几行, 默认显示前面 10 行;
- `-n` 数字: 显示几行;

```
tail -n 20 /etc/csh.login
```

- 取出文件后面几行, 默认后面 10 行;
- `-n` 数字: 显示几行;
- `-f` 文件名: 实时监控文件内容变化;

```
find -user ryu 查找用户 ryu 所属文件
```

```
find /home -name info.txt 在/home 下查找 info.txt 文件
```

```
find /home -name *.txt 正则匹配查找
```

```
find /home -size -20M 在/home 下查找小于 20M 的文件
```

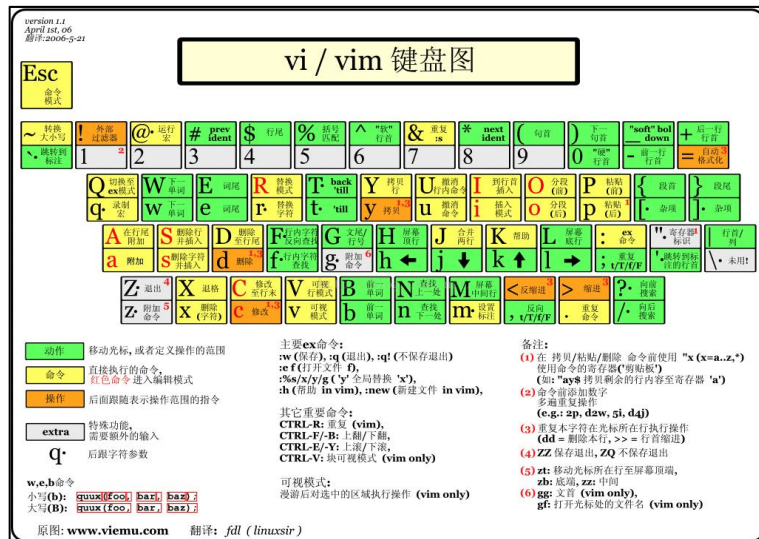
```
find /home -size -2048k
```

3、Vim 使用

3.1、什么是 Vim 编辑器

```
vim /etc/profile
```

- Vim 是从 vi 发展出来的一个文本编辑器, 方便编辑的功能特别丰富;
- 简单来说, vi 是老式的字处理器, 不过功能已经很其齐全了, 但是还是有可以进步的地方。vim 则可以说是程序开发者的一项很好用的工具;



3.2、三种使用模式

- 基本上 vi/vim 共分为三种模式，分别是命令模式（Command mode），输入模式（Insert mode）和底线命令模式（Last line mode）；

命令模式:

i 输入模式

x 删除当前光标所在字符

：切换到底线命令模式，可输入命令

输入模式:

a、常用

- 字符按键以及 Shift 组合：输入字符
- ENTER：换行
- BACK SPACE：删除字符，退格
- DEL：删除
- 方向键：在文本中移动光标
- HOME/END：移动光标到行首/行尾
- Page Up/Page Down：上/下翻页
- Insert：切换光标为输入/替换模式，光标变成竖线/下划线
- ESC：退出输入模式，切换至命令模式

b、移动光标的方法

- h 或者左箭头：向左移动一个字符
- j 或者下箭头：向下移动一个字符
- k 或者上箭头：向上移动一个字符

- **l** 或者右箭头：向右移动一个字符
- **Ctrl+f**：向下翻页，相当于 **Page Down**
- **Ctrl+b**：向上翻页，相当于 **Page Up**
- **Ctrl+d**：向下移动半页
- **Ctrl+u**：向上移动半页
- **+**：光标移动到非空格符的下一行
- **-**：光标移动到非空格符的上一行
- **n+SPACE**：n 表示数字，向右移动 n 个字符
- **0** 或者 **Home**：移动到行首
- **\$** 或者 **End**：移动到行尾
- **H**：移动到屏幕的最上方那一行的第一个字符
- **M**：移动到屏幕的中央那一行的第一个字符
- **L**：移动到屏幕的最下方那一行的第一个字符
- **G**：移动到这个档案的最后一行
- **nG**：移动到这个档案的第 n 行
- **gg**：移动到这个档案的第一行
- **n+Enter**：光标向下移动 n 行

c、搜索替换

- **/word**：向光标之下寻找一个 **word** 的字符串
- **?word**：向光标之上寻找一个 **word** 的字符串
- **n**：代表重复前一个搜寻的动作，向下继续寻找
- **N**：向上继续寻找

d、删除、复制与粘贴

- **x**：向后删除一个字符
- **X**：向前删除一个字符
- **nx**：连续向后删除 n 个字符
- **dd**：删除游标所在那一整行
- **ndd**：删除光标所在的向下 n 行
- **20dd**：删除 20 行
- **d1G**：删除所在到最后一行的所有数据
- **d\$**：删除游标所在处到该行的最后一个字符
- **d0**：删除游标所在处到该行的第一个字符
- **yy**：复制游标所在的那一行
- **nyy**：复制光标所在的向下 n 行
- **y1G**：复制游标所在行到第一行的所有数据
- **yG**：复制游标所在行到最后一行的所有数据
- **y0**：复制光标所在的那个字符到该行行首的所有数据
- **y\$**：复制光标所在的那个字符到该行行尾的所有数据
- **p**：将已复制的数据在光标下一行贴上
- **P**：则为贴在游标上一行
- **J**：将光标所在行与下一行的数据结合成同一行
- **c**：重复删除多个数据，例如向下删除 10 行，**[10cj]**
- **u**：复原前一个动作

- Ctrl+r: 重做上一个动作

底线命令模式:

`:w` 保存文件, 不退出

`:w filename` 将修改另存为 file, 不退出

`:w!` 强制保存, 不退出

`:wq` 保存文件, 并退出

`:wq!` 强制保存文件, 并退出

`:q` 不保存文件, 并推出

`:q!` 不保存文件, 并强制退出

`:e!` 放弃所有修改, 从上次保存文件开始再编辑

`:set nu` 显示行号

`:set nonu` 取消行号

4、账号管理

- 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;
- 实现用户账号的管理, 要完成的工作主要有如下几个方面:
 - 用户账号的添加、删除与修改;
 - 用户口令的管理;
 - 用户组的管理;

4.1、用户账号的管理

`useradd -m ryushihin` 添加用户并添加对应用户文件夹

- 增加用户账号就是在 `/etc/passwd` 文件中为新用户增加一条记录, 同时更新其它系统文件 `/etc/shadow`, `/etc/group` 等;
- `-c`: comment, 指定一段注释性描述;
- `-d`: directory, 指定用户主目录。如果此目录不存在, 则同时使用 `-m` 选项, 可以创建主目录;
- `-g`: group, 指定用户所属的用户组;
- `-G`: Group, 指定用户所属的附加组;
- `-m`: 使用者目录如不存在则自动建立;
- `-s`: Shell, 指定用户的登录 Shell;

- `-u`: 指定用户的用户号, 如果同时有`-o`选项, 则可以重复使用其它用户的标识号;

su test 切换至 test 用户

sudo su 从普通用户切换到 root 用户

exit/logout/ctrl+d 退回到原来用户

- `ctrl+d` 也是执行的 `exit` 命令;

su - root / su - 其它用户名 切换用户之后使用新用户的工作环境

- `$`表示普通用户;
- `#`表示超级用户, 也就是 `root` 用户;

userdel -r ryushihin 删除用户&配置&主目录

- 主要`/etc/passwd`, `/etc/shadow`, `/etc/group` 等的配置记录;
- `-r`: 作用是把用户的主目录一起删除;

usermod -s /bin/ksh -d /home/z -g developer liusibin 修改账号

- 此命令将用户 `liusibin` 的登录 Shell 修改 `ksh`, 主目录改为`/home/z`, 用户组改为 `developer`;
- 修改用户账号就是在`/etc/passwd`文件中为用户修改记录, 同时更新其它系统文件`/etc/shadow`, `/etc/group`等;
- `-c`: comment, 指定一段注释性描述;
- `-d`: directory, 指定用户主目录。如果此目录不存在, 则同时使用`-m`选项, 可以创建主目录;
- `-g`: group, 指定用户所属的用户组;
- `-G`: Group, 指定用户所属的附加组;
- `-m`: 使用者目录如不存在则自动建立;
- `-s`: Shell, 指定用户的登录 Shell;
- `-u`: 指定用户的用户号, 如果同时有`-o`选项, 则可以重复使用其它用户的标识号;

whoami 查看当前是哪个用户

4.2、用户口令的管理

- 用户管理的一项重要内容是用户口令的管理。用户账号刚创建时没有口令, 但是被系统锁定无法使用, 必须为其指定口令后才可以使使用, 即使是指定空口令;
- 超级用户可以为自己和其它用户指定口令, 普通用户只能用它修改自己的口令;

\$ passwd 普通用户修改自己的密码

- 普通用户修改自己的口令时, `passwd`命令会先询问原口令, 验证后再要求用户输入两遍新口令。如果两次输入的口令一致, 则将这个口令指定给用户;

passwd test1 超级用户指定修改 test1 用户的密码

- 超级用户为用户指定口令时, 就不需要知道原口令;

passwd -d test1 删除用户 test1

passwd -l test1 锁定用户 test1，使其不能登录

- -l: 锁定口令，即禁用账号；
- -u: 解锁账号；
- -d: 使账号无口令，即删除；
- -f: 强迫用户下次登录时修改口令；

4.3、用户组的管理

- 每个用户都有一个用户组，系统可以对一个用户组中的所有用户进行集中管理。不同 Linux 系统对用户组的规定有所不同，如 Linux 下的用户属于与它同名的用户组，这个用户组在创建用户时同时创建；
- 用户组的管理涉及用户组的添加、删除和修改。组的增加、删除和修改实际上就是对 `/etc/group` 文件的更新；

groupadd -g 101 group1 增加新组 group1，指定标识号为 101

- -g: GID，指定新用户组的组标识号。没有指定的话是在当前已有的最大组标识号的基础上加 1；
- -o: 一般与 -g 选项同时使用，表示新用户组的 GID 可以与系统已有用户组的 GID 相同；

groupdel group1 从系统中删除组 group1

groupmod -g 10000 -n group3(新) group2(旧) 将组 group2 的标识号改为 10000，组名修改为 group3

- g: GID，为用户组指定新的组标识号；
- o: 与 -g 选项同时使用，用户组的新 GID 可以与系统已有用户组的 GID 相同；
- n: 新用户组，将用户组的名字改为新名字；

\$ newgrp root 将当前用户切换到 root 用户组

- 前提条件是 root 用户组确实是该用户的主组或附加组；
- 如果一个用户同时属于多个用户组，那么用户可以在用户组之间切换，以便具有其它用户组的权限；

5、配置文件介绍

- 完成用户管理的工作有许多种方法，但是每一种方法实际上都是对有关的系统文件进行修改；
- 与用户和用户组相关的信息都存放在一些系统文件中，这些文件包括 `/etc/passwd`，`/etc/shadow`，`/etc/group` 等；

5.1、`/etc/passwd`

- Linux 系统中的每个用户都在 `/etc/passwd` 文件中有一个对应的记录行，它记录了这个用

户的一些基本属性;

- 这个文件对所有用户都是可读的;

```

[ryuu ~ root@RyuAlvin:~ - ssh root@172.107.70.231 - 80x24]

[root@RyuAlvin ~]# cat /etc/passwd
root:x:0:root:/root:/bin/bash
sbin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:10:games:/usr/games:/sbin/nologin
ftp:x:14:14:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:0:Nobody:/:/sbin/nologin
systemd-networkd:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:998:User for polkitd:/sbin/nologin
sshdx:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
postfix:x:89:89:/var/spool/postfix:/sbin/nologin
chrony:x:998:996:/var/lib/chrony:/sbin/nologin
nscd:x:28:28:NSSD daemon:/var/lib/nscd:/sbin/nologin
tcpdump:x:72:72:/:/sbin/nologin

[root@RyuAlvin ~]#

```

- 用户名:口令:用户标识号:组标识号:注释性描述:主目录:登录 shell

5.2、 /etc/shadow

- 由于 `/etc/passwd` 文件是所有用户都可读的，如果用户的密码太简单或规律比较明显的话，一台普通的计算机就能够很容易地将它破解，因此对安全性要求较高的 Linux 系统都把加密后的口令字分离出来，单独存放在一个文件中，这个文件是 `/etc/shadow` 文件。有超级用户才拥有该文件读权限，这就保证了用户密码的安全性；
- `/etc/shadow` 中的记录行与 `/etc/passwd` 中的一一对应，它由 `pwconv` 命令根据 `/etc/passwd` 中的数据自动产生；
- 登录名:加密口令:最后一次修改时间:最小时间间隔:最大时间间隔:警告时间:不活动时间:失效时间:标志

```
[root@RyuAlvin ~]# cat /etc/shadow  
root:$6$zDuk/Ackrs$lChJeeZ9wKqk/xEisFeG0L0oWfMuJ.uNhkJmS8Si1XfbQPTR3kt.qsmn  
APns/Q7rcUtlr/Ld6..slOjM3/1P9k4e:0:99999:7:::  
bin!:17834:0:99999:7::  
daemon!.:17834:0:99999:7::  
adm!:17834:0:99999:7::  
lp:!17834:0:99999:7::  
sync!:17834:0:99999:7::  
shutdown!:17834:0:99999:7::  
halt!:17834:0:99999:7::  
mail!:17834:0:99999:7::  
operator!:17834:0:99999:7::  
games!:17834:0:99999:7::  
ftp!:17834:0:99999:7::  
nobody!:17834:0:99999:7::  
system-network!!!:18378::::  
dbus!!!:18378::::  
polkitcd!!:18378!!!!  
ssh!!:18378!!!!  
postfix!!:18378!!!!  
chrony!!:18378!!!!  
ncsd!!:18378!!!!  
tcpdump!!:18378!!!!
```

```
[root@RyuAlvin ~]# █
```

5.3、 /etc/group

- 用户组的所有信息都存放在 `/etc/group` 文件中；
- 将用户分组是 Linux 系统中对用户进行管理及控制访问权限的一种手段；
- 每个用户都属于某个用户组；一个组中可以有多个用户，一个用户也可以属于不同的组；
- 当一个用户同时是多个组中的成员时，在 `/etc/passwd` 文件中记录的是用户所属的主组。

也就是登录时所属的默认组，而其它组称为附加组：

- 用户要访问属于附加组的文件时，必须首先使用 `newgrp` 命令使自己成为所要访问的组中的成员；
- 用户组的所有信息都存放在 `/etc/group` 文件中。此文件的格式也类似与 `/etc/passwd` 文件；
- 组名:口令:组标识号:组内用户列表

```
ryuu ~ root@RyuAlvin:~ -- ssh root@47.107.70.231 -- 80x24
[root@RyuAlvin ~]# cat /etc/shadow
root:$6$5zDuk/ich8:$1chJee2B9xJwKik/eXisEfG3L0dWFMuJ.uihkjm8S011Xfb0PTR3kT.q5nm
Apm5/DJTeUtlNz/m6.mLQIMJ/!19046:0:99999:7:::
bin:*:17834:0:99999:7:::
daemon:*:17834:0:99999:7:::
adm:*:17834:0:99999:7:::
lp:*:17834:0:99999:7:::
sync:*:17834:0:99999:7:::
shutdown:*:17834:0:99999:7:::
halt:*:17834:0:99999:7:::
mail:*:17834:0:99999:7:::
operator:*:17834:0:99999:7:::
games:*:17834:0:99999:7:::
ftp:*:17834:0:99999:7:::
nobody:*:17834:0:99999:7:::
systemd-network:!:18378:!!!!:
dbus:!:18378:!!!!:
polkitd:!:18378:!!!!:
sshd:!:18378:!!!!:
postfix:!:18378:!!!!:
chronyd:!:18378:!!!!:
nscd:!:18378:!!!!:
tcpdump:!:18378:!!!!:
[root@RyuAlvin ~]#
```

6、磁盘管理

- Linux 磁盘管理好坏直接关系到整个系统的性能问题：

6.1、磁盘空间查看

df 列出文件系统的整体磁盘使用量

```
[root@RyuAlvin ~]# df
文件系统      1K-块    已用    可用  已用% 挂载点
devtmpfs      930436      0   930436    0% /dev
tmpfs         940944      0   940944    0% /dev/shm
tmpfs         940944    452   940492    1% /run
tmpfs         940944      0   940944    0% /sys/fs/cgroup
/dev/vda1     41152812 2053872 37195232    6% /
tmpfs         188192      0   188192    0% /run/user/0
```

- 检查文件系统的磁盘空间占用情况。可以利用该命令来获取磁盘被占用了多少空间，目前还剩下多少空间等信息；
- `df` 可将系统内所有的文件系统列出来；
- 在 Linux 下如果 `df` 没有加任何选项，那么默认会将系统内所有的（不包含特殊内存内的文件系统与 `swap`）都以 1Kbytes 的容量来列出来；
- `-a`：列出所有的文件系统，包括系统特有的 `/proc` 等文件系统；
- `-k`：以 KBytes 的容量显示各文件系统；
- `-m`：以 MBytes 的容量显示各文件系统；
- `-h`：以人们较易阅读的 GBytes，MBytes，KBytes 等格式自行显示；
- `-H`：以 M=1000K 取代 M=1024K 的进位方式；
- `-T`：显示文件系统类型，连同该 `partition` 的 `filesystem` 名称（例如 `ext3`）也列出；
- `-i`：不用硬盘容量，而以 `inode` 的数量来显示；

df -h 将容量结果以易读的容量格式显示出来

```
[[root@RyuAlvin ~]# df -h
文件系统      容量  已用  可用  已用% 挂载点
devtmpfs      909M    0  909M    0% /dev
tmpfs         919M    0  919M    0% /dev/shm
tmpfs         919M  484K  919M    1% /run
tmpfs         919M    0  919M    0% /sys/fs/cgroup
/dev/vda1     40G   2.0G   36G    6% /
tmpfs        184M    0  184M    0% /run/user/0
```

df -aT 将系统内的所有特殊文件格式及名称都列出来

```
[[root@RyuAlvin ~]# df -aT
文件系统      类型      1K-块    已用    可用  已用% 挂载点
sysfs         sysfs          0        0        0     - /sys
proc          proc          0        0        0     - /proc
devtmpfs      devtmpfs    930436    0   930436    0% /dev
securityfs    securityfs    0        0        0     - /sys/kernel/security
tmpfs         tmpfs     940944    0   940944    0% /dev/shm
devpts        devpts        0        0        0     - /dev/pts
tmpfs         tmpfs     940944   484   940460    1% /run
tmpfs         tmpfs     940944    0   940944    0% /sys/fs/cgroup
cgroup        cgroup        0        0        0     - /sys/fs/cgroup/systemd
pstore        pstore        0        0        0     - /sys/fs/pstore
cgroup        cgroup        0        0        0     - /sys/fs/cgroup/cpu,cpu
acct
```

df -h /etc 将/etc 下的可用的磁盘容量以易读的容量格式显示

```
[[root@RyuAlvin ~]# df -h /etc
文件系统      容量  已用  可用  已用% 挂载点
/dev/vda1     40G   2.0G   36G    6% /
```

du 只列出当前目录下的所有文件夹容量（包括隐藏文件夹）

```
[[root@RyuAlvin sbin]# du -sm /*
0      /bin
146    /boot
0      /dev
34     /etc
1      /home
0      /lib
0      /lib64
1      /lost+found
1      /media
1      /mnt
1      /opt
du: 无法访问 "/proc/27432/task/27432/fd/4": 没有那个文件或目录
du: 无法访问 "/proc/27432/task/27432/fdinfo/4": 没有那个文件或目录
du: 无法访问 "/proc/27432/fd/4": 没有那个文件或目录
du: 无法访问 "/proc/27432/fdinfo/4": 没有那个文件或目录
0      /proc
1      /root
1      /run
0      /sbin
1      /srv
0      /sys
1      /tmp
```

- Linux du 命令也是查看使用空间的，但与 df 命令不同的是 Linux du 命令是对文件和目录磁盘使用空间的查看，还是和 df 命令有一些区别的；
- 直接输入 du 没有加任何选项时，则 du 会分析当前所在目录的文件与目录所占用的硬盘空间；
- -a: 列出所有的文件与目录容量，因为默认仅统计目录下的文件量而已；
- -h: 以人们较易读的容量格式（G/M）显示；
- -s: 列出总量而已，而不列出每个各别的目录占用容量；
- -S: 不包括子目录下的总计，与-s 有点差别；
- -k: 以 KBytes 列出容量显示；

- -m: 以 MBytes 列出容量显示;

`du -sm /*` 通配符*来代表每个目录

- 与 df 不一样的是, du 这个命令其实会直接到文件系统内去搜寻所有的文件数据;

6.2、磁盘挂载与卸载

- 根文件系统之外的其它文件要想能够被访问,都必须通过“关联”至根文件系统上的某个目录来实现,此关联操作即为“挂载”,此目录即为“挂载点”,解除此关联关系的过程称之为“卸载”;

`mount [-t 文件系统] [-L Label 名] [-o 额外选项] [-n] 装置文件名 挂载点`

`mkdir /mnt/hdc6`

`mount /dev/hdc6 /mnt/hdc6` 将/dev/hdc6(外部设备 U 盘)挂载到/mnt/hdc6 上

`umount [-fn] 装置文件名或挂载点`

`umount /dev/hdc6` 卸载/dev/hdc6(相当于退出 U 盘)

- -f: 强制卸载! 可用在类似网络文件系统(NFS)无法读取到的情况下;
- -n: 不升级/etc/mtab 情况下卸载;

7、进程管理

`ps -aux` 查看所有的进程

`ps -aux | grep mysql` 过滤进程, 只显示有 mysql 字样的进程信息

- -a: 显示当前终端运行的所有的进程信息;
- -u: 以用户的信息显示进程;
- -x: 显示后台运行进程的参数;

`ps -ef` 可以查看到父进程的信息

- <https://wenku.baidu.com/view/eb6334d6fa0f76c66137ee06eff9aef8941e488a.html>

`pstree -pu` 进程树


```

├── {AliYunDun}(6451)
├── {AliYunDun}(6452)
├── {AliYunDun}(6453)
├── {AliYunDun}(6454)
├── {AliYunDun}(6455)
├── {AliYunDun}(6456)
├── {AliYunDun}(6457)
├── {AliYunDun}(6458)
├── {AliYunDun}(6459)
├── {AliYunDun}(6460)
├── {AliYunDun}(6461)
├── {AliYunDun}(6466)
├── AliYunDunUpdate(6417) ─┬── {AliYunDunUpdate}(6418)
                        ├── {AliYunDunUpdate}(6419)
                        ├── {AliYunDunUpdate}(6420)
                        ├── {AliYunDunUpdate}(6421)
                        └── {AliYunDunUpdate}(6476)
├── agetty(1048)
├── agetty(1049)
└── aliyun-service(1040) ─┬── {aliyun-service}(1051)
                        └── {aliyun-service}(1052)

```

- -p: 显示父进程 ID;
- -u: 显示用户组;

lsof -i:8080 查看端口占用情况

kill -9 进程 ID 强制结束进程

8、服务管理

service 服务名 [start|stop|restart|reload|status]

service firewalld [start|stop|restart|reload|status]

systemctl [start|stop|restart|reload|status] 服务名

systemctl [start|stop|restart|reload|status] firewalld

systemctl [start|stop|restart|reload|status] firewalld.service

firewall-cmd --list-all 查看全部信息

firewall-cmd --list-ports 只看端口信息

```
firewall-cmd --zone=public --add-port=9090/tcp --permanent 开启 9090 端口(要重启防火墙)
```

```
firewall-cmd --zone=public --remove-port=9090/tcp --permanent 关闭 9090 端口(要重启防火墙)
```

- --zone: 作用域
- --add-port=9090/tcp: 添加端口, 格式为: 端口/通讯协议
- --permanent: 永久生效, 没有此参数重启后失效
- 确保 Linux 的防火墙端口是开启的, 如果是阿里云, 需要保证阿里云的安全组策略是开放的;



```
netstat -aptn 查看所有开启的端口号
```

```
netstat -nupl 查看所有使用 UDP 协议的端口号
```

```
netstat -ntpl 查看所有使用 TCP 协议的端口号
```

9、上传和下载

```
scp file root@47.107.70.231:/home/RyuAlvin 上传文件
```

```
scp -r folder root@47.107.70.231:/home/RyuAlvin 上传整个目录
```

```
scp root@47.107.70.231:/home/RyuAlvin/file folder 下载文件
```

```
scp -r root@47.107.70.231:/home/RyuAlvin folder 下载整个目录
```

10、解压缩文件

gzip /home/info.txt 压缩文件

- ->info.txt.gz
- 操作完了，源文件不存在；

gunzip /home/info.txt.gz 解压文件

- ->info.txt
- 操作完了，压缩文件不存在；

zip -r info.zip /home 将目录打包成 info.zip

- 常在项目打包发布中使用；

zip info.zip /home/info.txt 将文件打包成 info.zip

unzip info.zip 解压文件

unzip -d /home info.zip 将 info.zip 解压到指定目录下

tar -zcvf /home/info.zip info1.txt info2.txt 打包多个文件到指定目录下

tar -zcvf /home/home.tar.gz /home 将目录打包到指定目录下

tar -zxvf /home/info.tar.gz 解压到当前目录下

三、软件安装方式（3 种）及服务器基本环境搭建

1、JDK 安装

1.1、rpm 安装

● rpm 下载: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

```
rpm -qa|grep jdk 检查 jdk
```

```
[[root@RyuAlvin installer]# rpm -qa|grep jdk  
jdk-11-11.0.14-ga.x86_64
```

```
rpm -e --nodeps jdk-11-11.0.14-ga.x86_64 强制删除
```

```
rpm -ivh jdk-11.0.14_linux-x64_bin.rpm 安装 jdk
```

```
java -version 检查是否安装成功
```

1.2、压缩包安装

```
mkdir /usr/local/java
```

```
tar -zxvf jdk-8u351-linux-x64.tar.gz
```

```
vim /etc/profile
```

```
JAVA_HOME=/usr/local/java/jdk1.8.0_351
```

```
JRE_HOME=/usr/local/java/jdk1.8.0_351/jre
```

```
CLASS_PATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:$JRE_  
HOME/lib
```

```
PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin
```

```
export JAVA_HOME JRE_HOME CLASS_PATH PATH
```

```
source /etc/profile
```

```
java -version
```

2、配置环境变量

```
vim /etc/profile
```

```
ryuu — root@RyuAlvin:/usr/tomcat/apache-tomcat-9.0.59/bin — ssh root@47.107.70.231 — 116x24
if [ $UID -gt 199 ] && [ "/usr/bin/id -gn" = "/usr/bin/id -un" ]; then
else
    umask 022
fi
for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done
unset i
unset -f pathmunge

JAVA_HOME=/usr/java/jdk-11.0.14
PATH=$JAVA_HOME/bin:$PATH
CLASSPATH=$JAVA_HOME/lib:$CLASSPATH
export JAVA_HOME PATH CLASSPATH
"/etc/profile" 81L, 1947C
81,1 底部
```

`JAVA_HOME=/usr/java/jdk-11.0.14`

`PATH=$JAVA_HOME/bin:$PATH`

`CLASSPATH=$JAVA_HOME/lib:$CLASSPATH`

`export JAVA_HOME PATH CLASSPATH`

3、Tomcat 安装（解压缩安装）

● Tomcat 下载: <https://tomcat.apache.org/download-90.cgi>

`scp apache-tomcat-9.0.59.zip root@47.107.70.231:/home/installer`

`mkdir /usr/tomcat`

`cp /home/installer/apache-tomcat-9.0.59.zip /usr/tomcat`

`yum install -y unzip zip`

`cd /usr/tomcat`

`unzip apache-tomcat-9.0.59.zip`

`cd /usr/tomcat/apache-tomcat-9.0.59/bin`

`chmod 744 startup.sh shutdown.sh catalina.sh`

```
./startup.sh
```

```
47.107.70.231:8080
```

```
nohup java -jar xxx.jar (可用其它端口)
```

```
./shutdown.sh
```

4、安装 Docker (yum 安装)

- 官方安装参考手册: <https://docs.docker.com/install/linux/docker-ce/centos/>

4.1、检查版本

- `cat /etc/redhat-release`

```
[[root@RyuAlvin ~]# cat /etc/redhat-release  
CentOS Linux release 7.7.1908 (Core)
```

4.2、安装 gcc 相关

- `yum -y install gcc`
- `yum -y install gcc-c++`
- `yum -y install gcc gcc-c++`
- 虚拟机的话, 需要确保可以上外网;
- yum 一般是连网下载, 也可搭建本地仓库;
- 以上三者功能一样, 都是下载 gcc, 用哪个都行;
- `-y`: 默认都是 yes;
- `gcc -v`: 检查安装是否成功;

4.3、卸载旧版本

- `yum -y remove docker docker-common docker-selinux docker-engine`

4.4、安装需要的软件包

- `yum install -y yum-utils device-mapper-persistent-data lvm2`

4.5、设置 stable 镜像仓库

- `yum-config-manager`
`--add-repo https://download.docker.com/linux/centos/docker-ce.repo`
- `yum-config-manager`
`--add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo`

4.6、更新 yum 软件包索引

- `yum makecache fast`

4.7、安装 Docker CE

- `yum -y install docker-ce docker-ce-cli containerd.io`

4.8、启动 Docker

- `systemctl start docker`

4.9、测试

- `docker version`
- `docker run hello-world`
- `docker images`