linux基础

用户权限

```
lamp    linux + apache2 + mysql + php
java python  ssti
lnmp
```

- root

  最高权限，拥有系统几乎一切的读写执行功能 uid:0

  区分普通用户：1-499， 1-999 (centos7)

  对系统守护进程获取资源进行权限分配

登录用户：500+ ， 1000+(Centos7)

交互式登录

/root

- 组group

  linux组：groupname/GID

  管理员组：root, 0

  系统组：1-499， 1-999(centos)

  普通组：500+， 1000+(centos7)

  区分win中，用户名和组名不可以同名

  区分gid, uid win sid

linux组

用户的主要组：用户必须属于一个且只有一个主组，组名同用户名，且仅包含一个用户，私有组。用户的附加组，一个用户可以属于0个或者多个辅助组

```
id -u  查看uid号
id root 查看具体用户信息
id postfix 查看用户信息
```

配置文件

linux用户和组的主要配置文件：
`/etc/passwd` 用户及其属性信息(名称，uid, 主组id等)
`/etc/group`组及其属性信息
`/etc/shadow`用户密码及其相关属性
`/etc/gshadow`组密码及其相关属性,组信息文件

passwd:

```
login name: 登录用名
passwd: 密码(x)
UID: 用户身份编号(1000)
GID: 登录默认所在组编号(1000)
GECOS: 用户全名或注释
home directory: 用户主目录(/home/)
shell:用户默认使用shell(/bin/bash)


root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
```

```
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
```

shadow:

```
登录用户名
用户名密码：一般用sha512加密
从1970年1月1日起到密码最近一次被更改时间
密码再过几天可以被更改(0表示随时可被更改)
密码再过几天必须被更改(99999)表示永不过期
密码过期前几天系统提醒用户(默认为一周)
密码过期几天后账号会被锁定
从1970年1月1日算起，多少天后账号失效

root:$1$nJcOgVxG$gh1OO28cblVaqb7v28jIQ1:18881:0:99999:7:::
daemon:*:17737:0:99999:7:::
bin:*:17737:0:99999:7:::
sys:*:17737:0:99999:7:::
sync:*:17737:0:99999:7:::
games:*:17737:0:99999:7:::
man:*:17737:0:99999:7:::
```

- 用户管理命令

  `useradd`

  `usermod`

  `userdel`

- 组账号维护命令

  `groupadd`

  `groupmod`

  `groupdel`

  ```
  usermod -U user 解锁账号
  usermod -L user 锁定账号

  useradd -m ctf
  ```

useradd

默认值设定：`/etc/default/useradd`

```
useradd -D
useradd -D -s SHELL
useradd -D -b BASE_DIR
useradd -D -g GROUP

useradd [options] LOGIN

`-u`uid
`-o` 配合-u，不检查uid的唯一性
`-g gid`指明用户所属基本组，可为组名，也可以为gid
`-c 'comment'`用户的注释信息
`-d home_dir`指定的路径(不存在)为家目录
`-s shell`指明用户的默认shell程序，可用列表在/etc/shells文件中
`-G group1,[group2,...]`为用户指明附加组，组必须事先存在
`-N `不创建私用组做主组，使用users组做主组
`-r`创建系统用户centos6id<500  centos7 id<1000
`-m`强制创建家目录，用于系统用户
`-M`不创建家目录，用于系统用户
```

usermod

usermod [option] login

`-u uid` 新uid

`-g gid` 新主组

`-G group [..]` 新附加组，原来的附加组会被覆盖，若要保留原来的，则要同时使用 `-a` 选项

清空附加组：

```
usermod -G "" user 或
usermod -G user user

usermod  ctf -s /bin/zsh
```

`-s shell`新的默认shell
`-c comment`新的注释
`-d home`新加目录不会自动创建，若要创建新家目录并移动原家目录数据，同时使用-m选项
`-l login_name`新的名字
`-L`lock栏 指定用户，在/etc/shadow密码栏的增加!
`-U`unlock指定用户 将/etc/shadow密码栏的！拿掉
`-e yyyy-mm-dd`指明用户账号过期日期
`-f inactive`设定非活动期限

userdel

删除用户：userdel [option] login

-r 删除用户家目录

id

查看相关用户的id信息

id [option] [user]

`-u` 显示uid
`-g` 显示GID
`-G` 显示用户所属组的ID
`-n` 显示名称，需配合ugG使用

```
id -gn user 主组名称
```

su

切换用户或以其他用户身份执行命令

`su [options...] [-] [users [args...]]`
切换用户的方式：
`su UserName` 非登录式切换，即不会读取目标用户的配置文件，不改变当前工作目录
`su - UserName` 登录式切换，会读取目标用户的配置文件，切换至家目录，完全切换
root su 至其他用户无需密码，非root用户切换时需要密码
换个身份执行命令：
`su [-] UserName -c "command"`
选项：-l --login
`su -l UserName` 相当于su - UserName

sudo

临时借用root权限

```
sudo -s

当前用户的密码
```

passwd

passwd [options] UserName:修改指定用户的密码
常用选项：
`-d` 删除指定用户密码
`-l` 锁定指定用户
`-u` 解锁指定用户

`-e` 强制用户下次登录修改密码
`-f` 强制操作
`-n mindays` 指定最短使用期限
`-x maxdays` 最大使用期限
`-w warndays` 提前多少天开始警告
`-i inactivedays` 非活动期限
`--stdin` 从标准输入接受用户密码

```
echo "password" | passwd --stdin Username
```

groupmod

组属性修改

```
groupmod [option] .. group
-n group_name:新名字
-g gid :新id
```

groupdel

```
groupdel groupname
```

gpasswd

```
gpasswd [option] group

-a user 将user添加至指定组
-d user 从指定组中移除用户user
-A user1,user2,...  设置有管理权限的用户列表
```

newgrp

临时切换主组，如果用户本不属于此组，则需要组密码

groupmems

```
usermod -aG root www-data
```

groupmems [option] [action]

options:

`-g --group groupname` 更改指定组(root)

actions:

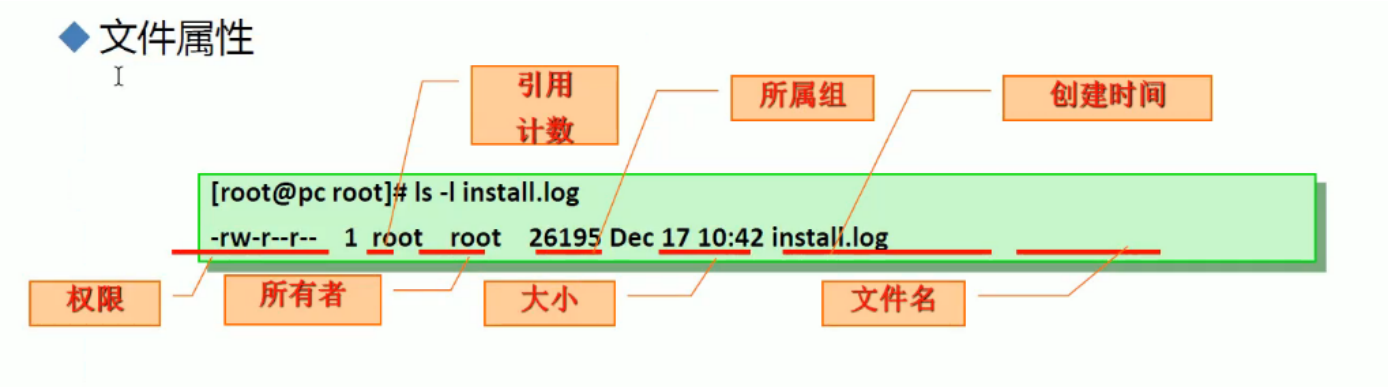
```
-a, -add username 指定用户加入组，一次只能加一个用户
-d, --delete username 从组中删除用户
-p, --purge 从组中清楚所有成员
-l, --list 显示组成员列表
```

uid相同，权限相同

文件权限

文件属性

```
┌──(root💀kali)-[/home/paddy]
└─# ls -al
                                                  130 ×
总用量 48
d  rwx r-x r-x     2 root root  4096  1月 21 18:01 .
drwxr-xr-x 5 root  root   4096  2月 16 11:41 ..
-rw-r--r-- 1 paddy paddy   220 11月  5  2020 .bash_logout
-rw-r--r-- 1 paddy paddy  4503 12月 22 15:27 .bashrc
-rw-r--r-- 1 paddy paddy  3526 11月  5  2020 .bashrc.original
-rw-r--r-- 1 paddy paddy 11759 11月 11  2020 .face
lrwxrwxrwx 1 paddy paddy     5 11月 11  2020 .face.icon -> .face
-rw-r--r-- 1 paddy paddy   807 11月  5  2020 .profile
-rw-r--r-- 1 paddy paddy  8063 11月 13  2020 .zshrc
```



属性操作：

- `chown` 设置文件的所有者
- `chgrp` 设置文件属组信息 `chgrp bin f2` 改变f2文件属组

  操作技巧：改变文件属组和所有者 `chown user.group file` 或者全改成属组一样的user `chown .group file` 只改组 `: .` 都可以
- 文件权限

```
r   文本文件    读   查看文件内容
w   文本文件    写     修改文件内容
x   excute    执行     程序二进制脚本

- rwx r-x r-x 1 root    root       1500  6月 12 15:17 46978.sh      755
drwxr-xr-x 2 root    root       4096  6月  9 17:16 公共
drwxr-xr-x 2 root    root       4096  6月  9 17:16 模板
drwxr-xr-x 2 root    root       4096  6月  9 17:16 视频
drwxr-xr-x 2 root    root       4096  6月  9 17:16 图片
drwxr-xr-x 2 root    root       4096  6月  9 17:16 文档
drwxr-xr-x 2 root    root       4096  6月  9 17:16 下载
drwxr-xr-x 2 root    root       4096  6月  9 17:16 音乐
```

chmod(模式法)

```
chmod who opt per file
who : u g o a
opt : + - +
per : r w x
chmod u+w file
chmod u=, g=r, o=w file
-R 递归
chmod -R a-x /file 去掉file目录下的所有执行权限，本身目录也执行相应权限

chmod +x filename

改变权限的命令
chmod -R +x dir
chmod 改变文件或目录的权限

chmod -R 755 /var/www/html/

chmod -x /var/www/html/index.php

chmod 755 abc：赋予abc权限rwxr-xr-x

chmod u=rwx，g=rx，o=rx abc：同上u=用户权限，g=组权限，o=不同组其他用户权限

chmod u-x，g+w abc：给abc去除用户执行的权限，增加组写的权限

chmod a+r abc：给所有用户添加读的权限
chmod -x abc



列如 /var/www/html/  添加执行权限
chmod 777 /var/www/html
chmod +x /var/www/html
```

数字法

```
r   4
w   2
x   1

所有用户都有执行权限
chmod 111 elf
```

chown

更改文件所有者或者所有组

```
chown hazel file 更改文件所有者
chown 所有者可以改权限，不可以改所有者
chown -R /data/ 递归设置权限，以后该目录下的所有新文件或者文件夹都将继承该权限
```

注：root如果没有执行权限也是没有办法执行的，但是root为权限最高的用户可以任意修改相应权限

常用命令

- ls

  列出当前目录下的文件

  ```
  ls -a
  ls -al
  ls -alh



  ls -a 包含隐藏文件
  ls -l 显示额外信息
  ls -R 目录递归通过
  ls -ld 目录和符号链接信息
  ls -1 文件分行显示
  ls -S 按从大到小排序文件大小      (字母优先级数字小写字母大写字母)
  ls -r 倒序排序
  ls -t 按mtime排序
  ls -u 配合-t选项，显示并按atime从新到旧排序
  ls -U 按目录存放顺序显示
  ls -X 按文件后缀排序

  mtime   修改时间
  atime    访问时间 ls --time=atime  访问时间排序
  ctime     元数据的记录时间  ll --time=ctime
  stat 文件     直接查看三个时间
  ```

  文件类型：

```
- 普通文件
d 目录文件
b 块设备
c 字符设备
l 符号链接文件
p 管道文件pipe
s 套接字文件socket
```

绝对路径和相对路径

- 绝对路径

  正斜杠开始
  完整的路径位置
  可用于任何想要指定一个文件名的时候

- 相对路径
  不以斜杠开头
  指定相对当前工作目录或者目录的位置
  可以作为一个简短的形式指定一个文件名

- 帮助文档的使用

  man

  -h or -help or --h

  搜索引擎

- cd

  切换目录

```
cd 路径名
cd
cd -
cd ..
cd ../
cd /
cd /路径
cd ~    家目录
```

- pwd

  输出当前路径

- mkdir

  建立文件夹

```
mkdir ~/hello
mkdir -p
```

- touch

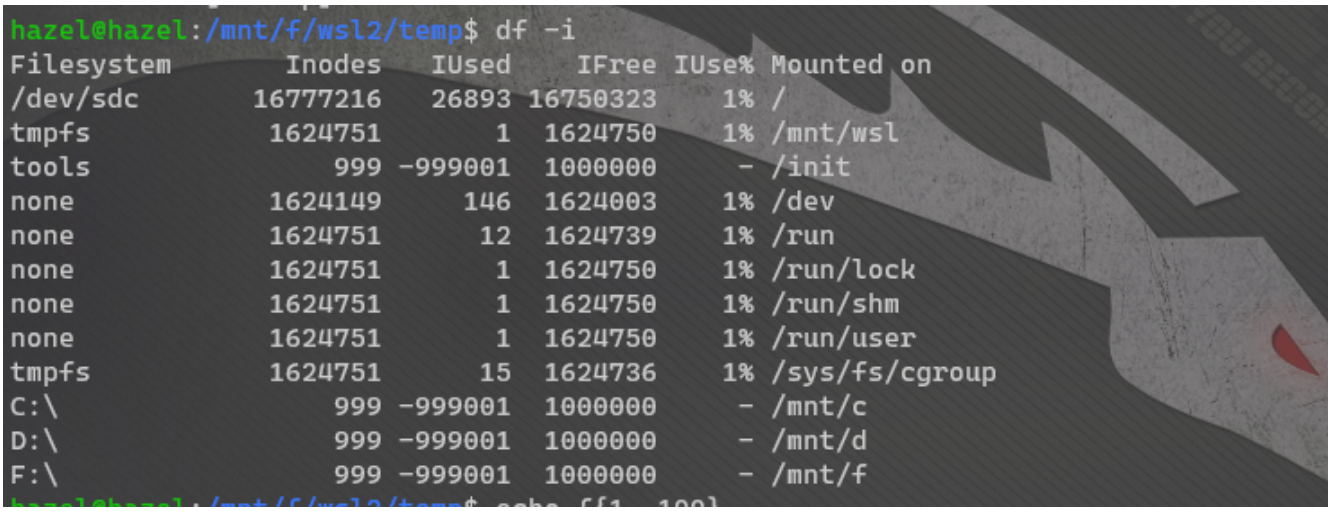
创建文件

ctf搅屎棍操作：占用节点号或者循环写入垃圾数据，撑爆服务器硬盘
或者 `dd if=/dev/zero of=/boot/bigfile bs=1M count=硬盘空间大小`

```
touch f{1..最大节点数} 会触发参数错误，无法创建
echo f{1..节点最大数} | xargs touch 创建超过节点数量的文件，来达到提示硬盘空间不足
no space left on device
while true; do touch {1.1000000000} ;done
```

文件太多`ls *`和`rm *`删不了，提示文件太多了，可以通过直接来删除文件夹来删除文件

`rm file`删除只是目录项，节点编号，数据区没有删除。同一目录下移动文件之所以比不同分区下移动文件块，同一分区下只改变了目录项，而不同分区下都要改变，重新分配节点号。。建文件的速度慢于删文件，是因为删文件没有删除数据区。



- cp

  复制文件，文件夹，重命名

  ```
  cp 1.txt 2.txt
  cp -r
  ```

- rename(文件的重命名，**注意：有些系统时没有的**)

  rename conf conf.bak *.conf

- mv

  移动文件，文件夹，重命名

  ```
  mv 1.txt 2.txt
  mv a b
  ```

- rmdir

  只删除空目录，不能删除非空目录
  `-p` 递归删除父空目录
  `-v` 显示详细信息

```
rmdir d1/d2/d3/d4   直接删除d4空目录，加上-p之后如果前面的都是空目录，直接全部删除
```

- rm **注意：慎用**

  删除文件，文件夹

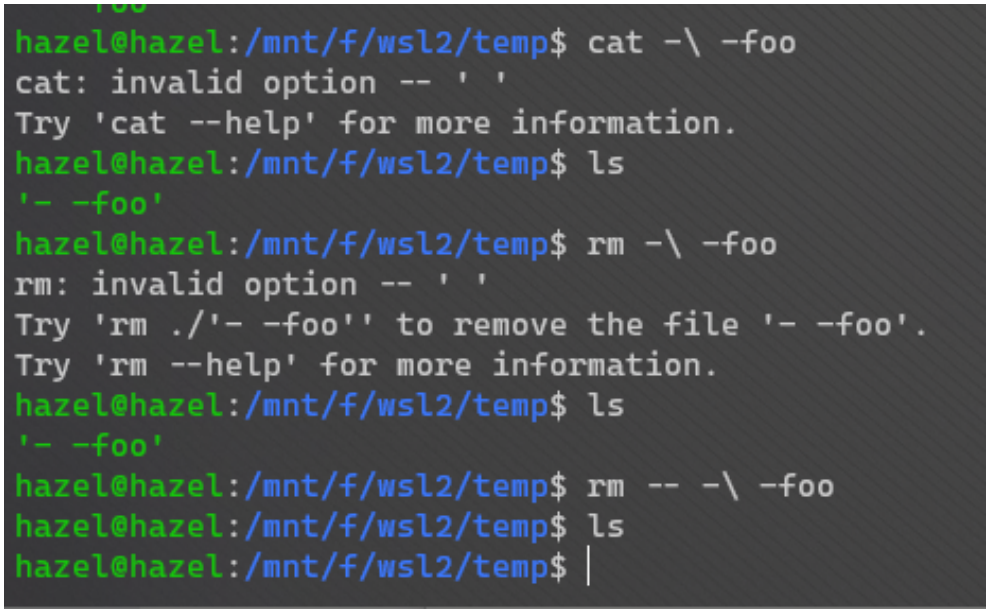
  ```
  rm -rf ~/hello
  ```

  注：ctf骚操作 创建删不掉的文件:

  ```
  创建 -foo 文件


  touch -- -\ -1234
  rm -rf -- -\ -1234

  touch -- --123
  rm -rf -- --123
  ```



- 权限修改相关的

  chown
  chmod

  ```
  /var/www/html
  chmod -R 777
  chown
  ```

- echo

  输出命令

  echo $PATH　**注: 一定要大写**

- cat / tac / head / tail / more / less

  查看文件内容

  `cat /etc/passwd`

- wget 网络上的东西下载下来

- curl 发送相关请求

  `curl http://www.baidu.com`

  其他工具httpie
- clear

  清空终端，或者快捷键 `Ctrl + l`
- su

  切换用户
- sudo

  以最高权限执行

  `sudo apt-get update && apt-get upgrade`
- exit

  退出，exit 或者快捷键 `Ctrl + D`
- nc ~~ncat~~

  nc 127.0.0.1 9000

  ~~ncat 127.0.0.1 9000~~
- ifconfig

  输出网卡信息
- **ssh**

  ssh 远程连接
- ping

  测试网络联通性
- 命令分隔符

  ```
  |
  ||    或者
  &&    与
  ;     分割
  ${IFS} 空格
  ```

- cmd1 | cmd2

  管道符号 将cmd1的stdout pipe 到cmd2的stdin

  echo -e hi | md5sum
- `cmd`

  反引号执行命令

  export PATH=$PATH:`pwd`
- cmd > file

将file的内容reditect到cmd的stdin
特殊操作：

```
ls  > /dev/pts/1   重定向另一个窗口显示
```

- cmd << file

  将file的内容追加到cmd的stdin，而不是覆盖

  多行重定向

```
mail -s "Please Call"  admin@magedu.com <<END
>hi, wnag,
>
>pajdlshfskhjahsjhf
>
>jdlkajf
>END成对出现
```

- tar

  压缩命令

```
tar -czvf /var/www/html/ * .htaccess
scp -r -P 990 ctf@39.99.198.177:/var/www/html/123.tar.gz ./

tar -zxvf 123.tar.gz


tar -czvf
tar -zxvf 压缩包的名字 -C 指定解压目录
```

- diff

  源码对比工具beycond compare

```
diff -u foo.conf foo2.conf > foo.patch
patch -b foo.conf foo.patch

beycond compare
```

  比较两个文件之间的区别

```
diff foo.conf foo2.conf
```

  `-u` 详细信息
  通过两个文件之间的不同来恢复其中一个文件

  注：后期绝望的时候可以使用diff来找找cms之间的不同，找码(前提是会删码，会搞事情。如果对手套路太骚，酌情考虑是否申请重置) 其实文件太多了这个也没什么用

- whoami
- who 当前系统所有登录会话
- w系统当前所有的登录会话及所做的操作

常见编辑文件命令

nano(不建议使用)

vim

vim模式

命令(normal)
插入(insert)
拓展命令(extended command)

命令模式操作

字符编辑：

```
x:      删除光标处的字符
#x      删除光标处起始的#个字符            x时剪切p时粘贴
xp     交换光标处所在的字符及其后面的字符的位置
~          转换大小写
J      删除当前行后的换行符
r      替换光标所在处的字符
R       切换成replace模式
```

删除命令：

```
d    删除命令，可结合光标跳转字符，实现范围删除
d$       删除到行尾
d$      删除到行首
d0 删除到行首
dw
de
db
#command
dd       删除光标所在行
#dd   多行删除
D    从当前光标位置一直删除到行尾，等同于d$
```

复制命令；

```
y
y$
y^
y0
ye
yw
yb
#command
yy    复制行
#yy   复制多行
Y复制整行
```

粘贴命令

```
p  缓冲区存的如果为整行，  则粘贴在当前光标所在行的下方，否则粘贴至当前光标所在处的后面
P       上方，  或光标前
```

改变命令

```
c修改后切换成插入模式
c$
c^
c0
cb
ce
cw
#command
cc       删除当前行并输入新内容相当于S
#cc
C:  删除当前光标到行尾，并切换成插入模式
```

```
100iwang [esc]粘贴wang 100次
<start postion> <command> <end position>
command:
y 复制，d删除   gU 变大写，  gu变小写
列如：0y$
0    先到行头
y   从这里开始拷贝
$    拷贝到本行最后一个字符
 ye    从当前位置拷贝到本单词的最后一个字符
```

撤销

```
u 撤销最近的更改
#u    撤销之前多次更改
U  撤销光标落在这行后 的所有此行的更改
按ctro -r 重做最后的撤销更改
. 重复前一次操作
n.重复前一个操作n次
```

```
di"        光标在”“之间 则删除""之间的内容
yi(    光标在()之间，则复制()之间的内容
vi[    光标在[]之间则选中[]之间的内容
dtx   删除字符直到遇见光标之后的第一个x字符
ytx    复制
```

拓展命令模式

地址定界

```
: start_pos ,end_pos
# 具体第#行， 例如2表示第二行
#，#从左侧#表示起始行，到右侧#表示结尾行
#，+# 从左侧#表示的起始行，加上右侧#表示的行数
         : 2，+3 表示2到5行
.当前行
$最后一行
   .,$-1 当前行到倒数第二行
 % 全文，相当于1，$

 /pat1/,/pat2/ 从第一次被pat1匹配到的行开始，一直到第一次被pat2匹配到的行结束
 #,/pat/
 /pat/,$
 使用方式：后跟一个编辑命令
 d
 y
 w file 将范围内的行另存至指定文件中
 r   file 在指定位置插入指定文件中的所有内容
```

查找

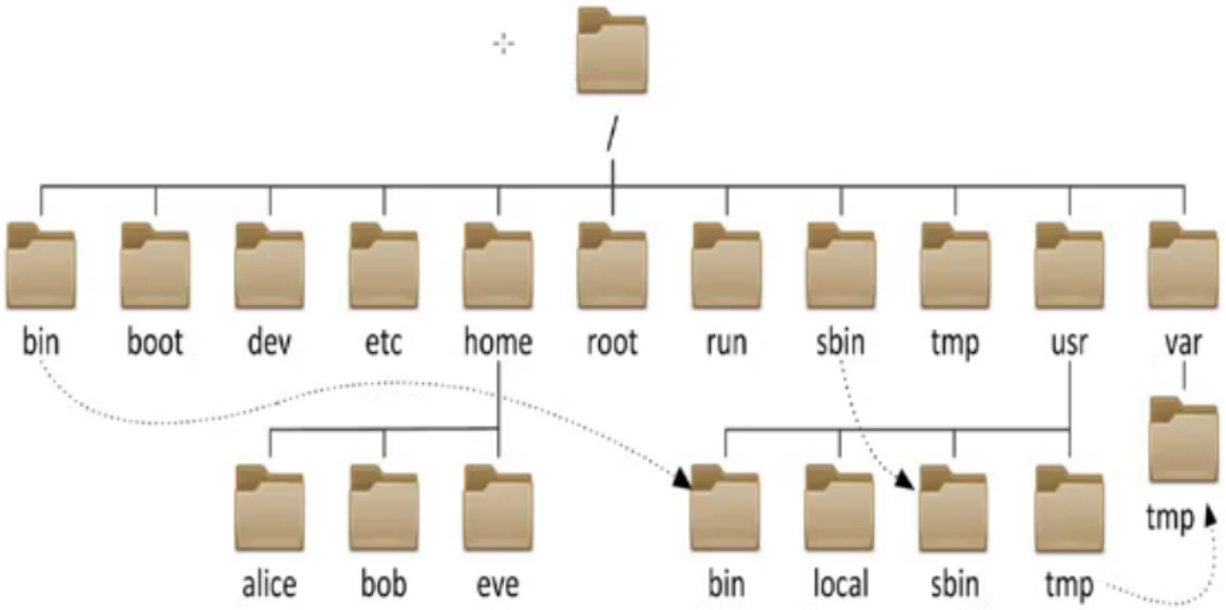
```
/pattern 从当前光标所在处向文件尾部查找
?pattern 从当前光标所在处向文件头部查找
n    与命令同方向
N     与命令反方向
```

查找并替换

```
s  在扩展模式下完成查找替换操作
格式 s/要查找的内容/替换为的内容/修饰符
要查找的内容:可使用模式
替换为的内容：不能使用模式，但可以使用\1,\2，...等后向引用符号，还可以使用&引用前面查找时查找到的整个内容
修饰符:
i  忽略大小写
g：全局替换，默认情况下，每一行只替换第一次出现
gc ：  全局替换，每次替换前询问

查找替换中的分隔符/可替换为其他符号，列入
s@/etc@/var/@g
s#/boot#/#i
```

目录结构



`/dev/zero` 字符设备 c开头的是字符设备，b开头的是块设备

`/etc/` 配置文件

文件系统:

- 文件和目录被组织成一个单根倒置树结构，文件系统从根目录下开始用 `/` 表示
- 根文件系统(rootfs): root filesystem, 文件名称区分大小写(特定文件系统) 以 `.` 开头隐藏文件
- 路径分隔 `/`
- 文件有两类数据:

  元数据：metadata(文件属性)

  数据: data

- 文件系统分层结构：LSB linux Standard Base
- FHS:(fileststem Hierarchy Standard)

  `proc` 内存数据存放 `sys` 系统硬件 `misc` 杂项， `cd /misc` 目录为空可以进入 cd文案金里面 `cd cd/`

  添加硬盘后硬盘生效设置`echo  '- - -' > /sys/class/scsi_host/host0/scan`识别硬件

  `mnt  media` 硬盘挂载点 `mount  /dev/sdb3 /mnt`挂在

大致目录存放文件

`/boot` : 引导文件存放目录，内核文件(vmlinuz), 引导加载器(bootloader, grub)都存放于此目录

`/bin` 所有用户使用的基本命令；不饿能关联至独立分区，OS启动即会用到的程序

`/sbin` 管理类的基本命令； 不能关联至独立分区，os启动即会用到的程序

`/lib` 启动时程序依赖的基本共享库文件以及内核模块文件(/lib/modules)

`/lib64` 专用于x86_64系统上的辅助共享库文件存放位置

`/etc` 配置文件目录

`/home/username` 普通用户家目录

`/root` 管理员的家目录

`/media` 便携式移动设备挂载点

`var` variable data files

```
cache：应用程序缓存数据目录
lib：应用程序状态信息数据
local：专用于为/usr/local下的应用程序存储可变数据
lock ：锁文件
log：日志目录及文件
opt：专用于为/opt 下的应用程序存储可变数据
run：运行中的进程相关数据，通常用于存储进程pid文件
spool：  应用程序数据池
tmp：保存系统两次重启之间产生的临时数据
```

`/proc` 用于输出内核于进与进程信息相关的虚拟文件系统
`sys` 用于输出当前系统上硬件设备相关信息虚拟文件系统
`selinux` secuity enhanced linux selinux 相关的安全策略等信息的存储位置

二进制程序：`/bin, /sbin, /usr/bin, /usr/sbin, /usr/local/bin, /usr/local/sbin`
库文件：`/lib, /lib64, /usr/lib, /usr/lib64, /usr/local/lib, /usr/local/lib64`
配置文件：`/etc, /etc/DIREXTORY, /usr/local/etc`
帮助文件：`/usr/share/man, /usr/share/doc, /usr/local/share/man, /usr/local/share/doc`
