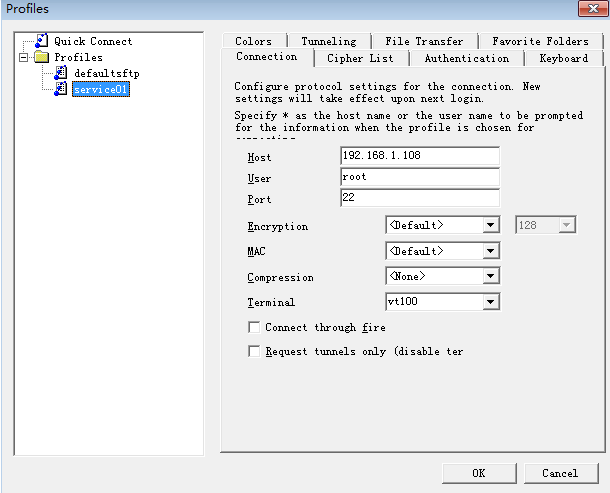
linux 学习

第一天 linux远程连接

远程连接使用SSH Secure Shell

选择profiles --> Edit profiles

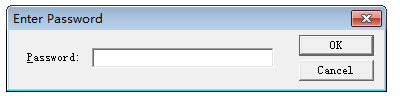


输入远程的 host 192.168.1.108 这个是linux 远端IP

user 用户 root 登录用户

port 22 端口号

输入完后OK 提示输入密码 远程机器的登录密码



如果错误提示连接不上的情况 错误排查如下三种问题

1.服务器的防火墙

iptabes./etc/init.d/iptables stop #关闭防火墙

或者去配置 防火墙ssh端口 一般为22

vi /etc/sysconfig/iptables #编辑防火墙配置文件

文件配置如下连接，第二步骤

http://www.cnblogs.com/Jerry1104/p/7653608.html

2.ssh 服务有问题

3.客户端到服务器端的线路有问题

ping www.baidu.com #ping一下看看网络是否有问题

linux 基础命令学习记录

进入界面

[root@bogon /]# #号表示root 用户

命令 [参数] [路径文件] 注意之间有至少一个空格

rm -f /tmp/oldboy.log

1）创建一个/ 根下面的data目录命令 make directory =mkdir 缩写

[root@bogon /]# make /data

[root@bogon /]# ls # 查看data创建成功没有

bin data download home lib64 mnt proc run srv tmp var

boot dev etc lib media opt root sbin sys usr

2)在/data 下创建一个文本 oldboy.txt

[root@bogon /]# cd data/

[root@bogon data]# ls

[root@bogon data]# pwd

/data

[root@bogon data]# touch oldboy.txt

[root@bogon data]# ls

oldboy.txt

3)增加文件内容 vi/vim 命令

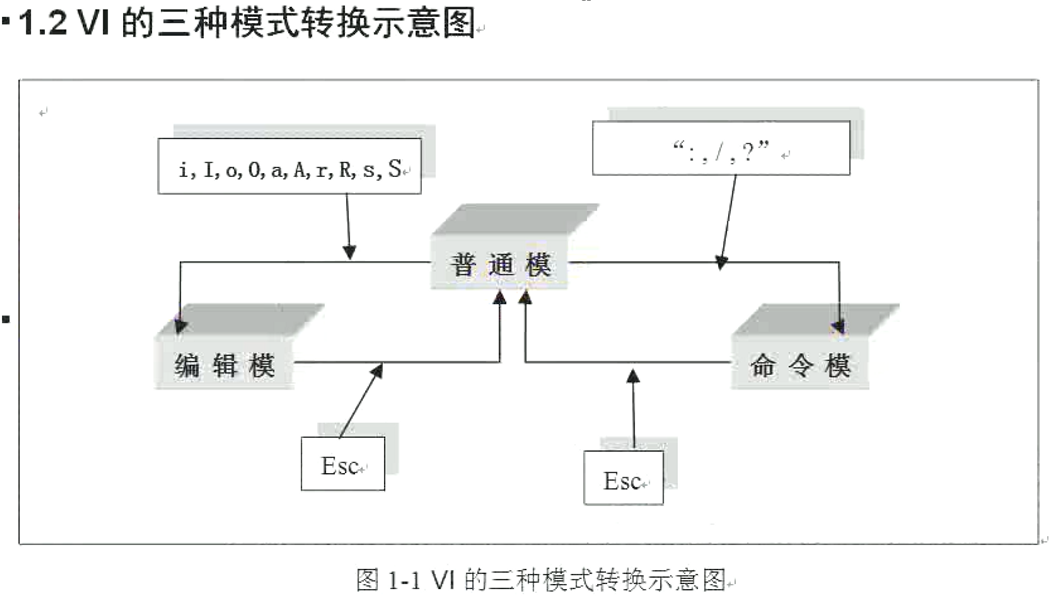
[root@bogon data]# vi oldboy.txt

用vi 命令 进入文本后安i 或者 a进入编辑模式 ——INSERT——

输入完后 按 esc 退出编辑模式 ，进入命令模式

保存退出输入 :wq!

q 是退出 w是写入 !强制退出



4) cat查看文本内容 用 cat 命令

[root@bogon data]# cat oldboy.txt

5) echo 写入输出

[root@bogon data]# echo "echo写入输"

echo写入输

把内容写入oldboy.txt文本 用 > 会把以前文本的内容清空然后加入新的内容

[root@bogon data]# echo "echo写入输" >oldboy.txt

[root@bogon data]# cat oldboy.txt

echo写入输

如果想追加在原来的内容后面，用双>>

[root@bogon data]# echo "echo写入输" >>oldboy.txt

[root@bogon data]# cat oldboy.txt

echo写入输

echo写入输

[root@bogon data]# cat >> oldboy.txt << oldboy

> 1

> 1

> 1

> 1

> 1

> 1

> 1

> 23124

>oldboy

[root@bogon data]# cat oldboy.txt

把文件拷贝到目录下( 默认情况只能拷贝文件 要拷贝目录要加 -a 或者 -r)

cp命令copy cp 文件 路径

拷贝文件

[root@bogon data]# cp oldboy.txt /emp/

拷贝目录 -a

[root@bogon /]# cp -a /data/ /emp/

移动目录

[root@bogon /]# mv /data /tmp/

rm 删除命令 remove （删除文件用-f 删除目录用 -rf）

[root@bogon /]# rm -rf /data

rmdir删除目录

[root@bogon data]# rmdir data

ll 查看命令

[root@bogon data]# ll

total 4

-rw-r--r-- 1 root root 42 Oct 14 00:22 oldboy.txt

查看命令三剑客（过滤功能）

grep命令（-v 加内容 排除 -f50 显示50行）

[root@bogon data]# cat oldboy.txt

echo写入输

echo写入输

1

1

23124

[root@bogon data]# grep -v echo oldboy.txt # -v 内容 排除

1

1

23124

head 命令 显示文件头部 -n行数 -行数

[root@bogon data]# head -1 oldboy.txt

echo写入输

tail 尾部 显示文件尾部-n -行数

[root@bogon data]# tail -10 oldboy.txt

sed 取各种内容 /过滤的内容/p p表示打印的意思 -n 表示取消默认输出 ^ 把内容开头的显示出来

[root@bogon data]# sed -n /1/p oldboy.txt

[root@bogon data]# sed -n /^1/p oldboy.txt

tree 树结构查询 需要安装 。 yum install tree

[root@bogon data]# tree

.

忖?oldboy.txt

0 directories, 1 file

出现乱码调整字符集 临时调整

[root@bogon data]# LANG=en

[root@bogon data]# tree

.

`-- oldboy.txt

man 命令 查看帮助文档

man mkdir

help 帮助文档

cp --help

tabel 命令 不全功能

find 命令 查询 遍历所有目录查询不是建数据库查询

find / -name "cat"

[root@bogon data]# find / -name "cat"

/usr/bin/cat

[root@bogon data]# find / -type f -name "cat"

/usr/bin/cat

[root@bogon data]# find / -type f -name "\*txt"

管道加 | 多个命令同时执行 显示后命令

[root@bogon data]# find / -type f -name "\*txt" | grep "1" -v

查看data.txt 文件 的10条中的头2条数据 用管道输出

[root@bogon data]# tail -10 data.txt| head -2

取消前10和后30的数据 查看中间的

[root@bogon data]# sed -n 10,30p data.txt

查询 用判断语句 shell 常用命令

[root@bogon data]# awk '{if(NR<8&&NR>2) print $0"\n"}'

tail 查询输出命令

sed查询输出命令

awk查询输出命令 重要命令

sed修改文件替换

[root@bogon data]# sed -i s#1#update#g oldboy.txt

[root@bogon data]# cat oldboy.txt

echo写入输

update

update

update

23update24

[root@bogon data]# find / -type f -name "oldboy.txt"|xargs cat

echo写入输

echo写入输

update

update

23update24

[root@bogon data]# find / -type f -name "oldboy.txt"|xargs ls

/data/oldboy.txt

[root@bogon data]# find / -type f -name "oldboy.txt"|xargs sed -i s#update#1234#g

[root@bogon data]# cat oldboy.txt

echo写入输

echo写入输

1234

1234

23123424

history 命令 查看历史记录使用过的命令

1062 history

[root@bogon data]# history

[root@bogon ~]# history -c # 清空

查看系统版本（基本都是用64位系统 x86\_64 ）

[root@bogon data]# cat /etc/redhat-release

CentOS Linux release 7.4.1708 (Core)

查看内核版本

[root@bogon data]# uname -r

3.10.0-693.2.2.el7.x86\_64

查看系统 32还是64位

[root@bogon data]# uname -m

x86\_64

查看系统全部信息

[root@bogon data]# uname -a

Linux bogon 3.10.0-693.2.2.el7.x86\_64 #1 SMP Tue Sep 12 22:26:13 UTC 2017 x86\_64 x86\_64 x86\_64 GNU/Linux

总结

mkdir 创建目录命令 make directorys

创建多个 mkdir -p /oldboy/test

ls -l (long) d (directory) 显示目录或文件 全程 list

cd 切换目录层次 change directory

pwd 查看当前目录路径

touch 新建一个文本 文件不存在就建新文件 ，如果文件存在就改变文件访问时间修改 atime 时间戳信息

vi/vim 类似于记事本 编辑器有命令模式（：wq！ 强制保存退出）和插入模式相互转换（i 编辑） esc切换 :set nu 表示显示行号

cat 查看文本内容

echo写入输

多内容插入

cp 拷贝文件 cp -apr 拷贝目录

mv 移动

rm 删除文件或目录 -f 强制删除 -r 递归删除 -rf 删除指定的目录和文件

ll 查看命令

rmdir 删除空目录 已经不用了

查看三剑客

grep 查看或者排除

egrep 相当于 grep -E

head 头部 显示文件头部 -n行数 -行数

tail尾部 显示文件尾部-n -行数

sed 取各种内容 /过滤的内容/p p表示打印的意思 -n 表示取消默认输出

tree 树结构查询 需要安装 。 yum install tree

man 命令 查看帮助文档

help 帮助文档

tabel 命令 不全功能

alias 显示设置别名

unalias 取消别名

which 查看命令的路径 whereis， locate ，find

find \* 查找命令 -type （f d） -name ' dd' 按名字查找 - mtime 按照修改时间查找 如 +4 就是4天以前 4 表示第4天的 -4 表示最近4天

tail 查询输出命令

sed查询输出命令

awk查询输出命令 重要命令

history 查看使用过的命令

文件比较工具diff 和vimdiff

diff a.txt b.txt 比较2个文本的内容

vimdiff a.txt b.txt

su 切换用户

useradd 添加用户 adduser 都可以

passwd 设置密码

env 查看用户所有的信息和配置

whoami 查看当前用户命令

yum 安装软件工具 yum install + 要安装的软件

rpm -ivh 安装类似exe程序的安装

http://mirrors.163.com # 163镜像 http://mirrors.163.com/.help/centos.html

clear：这个命令将会刷新屏幕，本质上只是让终端显示页向后翻了一页，如果向上滚动屏幕还可以看到之前的操作信息。一般都会用这个命令。  
reset：这个命令将完全刷新终端屏幕，之前的终端输入操作信息将都会被清空，这样虽然比较清爽，但整个命令过程速度有点慢，使用较少

chkconfig 是查看开机自动软件的命令

chattr 改变文件属性的

lsattr 查看文件属性的

符号命令

；多个命令的分隔符

/ 根或者路径的分隔符

>或1> 重定向 （数据流朝箭头方向流动）覆盖原来的文件 文本内容用 "" 括起来

>>或1>> 追加重定向 （数据流朝箭头方向流动）在原来的结尾追加内容

<或<0 输入重定向： 输入重定向用于改变命令的输入，指定输入内容 后跟文件名

<<或<<0 输入重定向： 后跟字符串 用来表示输入结束 也可用用ctrl + c 来结束输入 要换行

.. 上一级目录

. 当前目录 (. 在正则表达式中可以做任意字符)

~ 用户目录

| 管道 通过管道吧前一根命令的输出交给后一个命令执行继续处理

! 表示! 加行号 执行行号执行过的命令 要先有 history 执行后可以查看行号

!! 表示执行最近的命令

linux 快捷键

tab 补全

Ctrl+c 终止任务命令程序

Ctrl+d 退出当前应用环境

Ctrl+a 跳到最前面

Ctrl+e

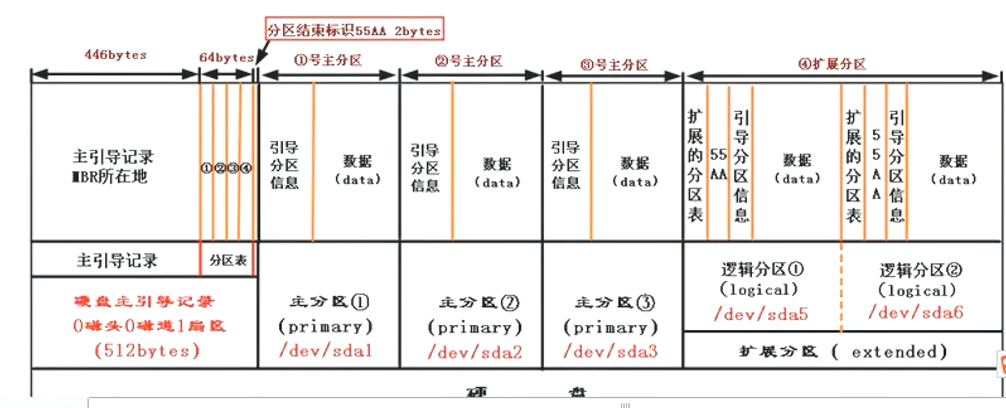
Ctrl+r 查找历史记录查找用过的命令记录

Ctrl+shift+c 复制 Ctrl+insert

Ctrl+shift+v 粘贴 shift+insert

第二课

系统磁盘的原理



主分区加扩展分区

3p+1e 或者 2p+1e 或 1p+1e 只能有一个扩展分区

3个主分区一个扩展分区 ，扩展分区不能用 ，还必须在拓展分区上划分多个逻辑分区，然后格式化，然后才能存数据或装系统

查看系统创建了几个分区

[root@bogon /]# ll /dev/sd\*

brw-rw---- 1 root disk 8, 0 Oct 13 06:41 /dev/sda

brw-rw---- 1 root disk 8, 1 Oct 13 06:41 /dev/sda1

brw-rw---- 1 root disk 8, 2 Oct 13 06:41 /dev/sda2

有3个主分区



安装系统后， 对系统做优化

[root@bogon /]# whoami # 查看当前用户

root

1.设置主机名

添加普通用户设置密码 用户名/密码 ：admin/admin

[root@bogon ~]# adduser admin

[root@bogon ~]# passwd admin

Changing password for user admin.

New password: admin

BAD PASSWORD: The password is shorter than 8 characters

Retype new password: admin

passwd: all authentication tokens updated successfully.

[root@bogon ~]# echo admin|passwd --stdin admin # 明文设置密码

[admin@bogon ~]$ # 普通用户是$ 符合

[admin@bogon ~]$ su root #su 切换到root 用户下

Password:

[root@bogon ~]#

[root@bogon ~]# env 查看当前用户所有的配置和信息

[root@bogon ~]# cat /etc/yum.repos.d/CentOS-Base.repo #查看镜像

参考下安装文档http://mirrors.163.com # 163镜像 http://mirrors.163.com/.help/centos.html

163镜像网站

http://mirrors.163.com/

修改镜像前做备份

[root@bogon yum.repos.d]# cp CentOS-Base.repo CentOS-Base.repo.ori

[root@bogon yum.repos.d]# ls

CentOS-Base.repo CentOS-Debuginfo.repo CentOS-Vault.repo

CentOS-Base.repo.ori CentOS-Media.repo CentOS-fasttrack.repo

CentOS-CR.repo CentOS-Sources.repo

然后下载163

[root@bogon yum.repos.d]# wget http://mirrors.163.com/.help/CentOS7-Base-163.repo

[root@bogon yum.repos.d]# ls

CentOS-Base.repo CentOS-Debuginfo.repo CentOS-Vault.repo

CentOS-Base.repo.ori CentOS-Media.repo CentOS-fasttrack.repo

CentOS-CR.repo CentOS-Sources.repo CentOS7-Base-163.repo

替换

[root@bogon yum.repos.d]# cp CentOS7-Base-163.repo CentOS-Base.repo

cp: overwrite 'CentOS-Base.repo'? y

[root@bogon yum.repos.d]# ls

CentOS-Base.repo CentOS-Debuginfo.repo CentOS-Vault.repo

CentOS-Base.repo.ori CentOS-Media.repo CentOS-fasttrack.repo

CentOS-CR.repo CentOS-Sources.repo CentOS7-Base-163.repo

更新

[root@bogon yum.repos.d]# yum -y update

#yum check-update

#yum remove 软件包名

#yum install 软件包名

#yum update 更新

#yum upgrade 更新

[root@bogon ~]# yum grouplist 查看安装了哪些包 安装完后可以查看到安装

自己配置本地yum

必要的软件包安装

[root@bogon ~]# yum -y install tree telnet dos2unix sysstat lrzsz

简单调优

关闭SELinux 是一个安全的东西

[root@bogon ~]# vi /etc/selinux/config 修改 SELINUX= enforcing 为SELINUX= disabled 就是关闭了

[root@bogon ~]# cd /etc/selinux

[root@bogon selinux]# sed -i s# SELINUX= enforcing #SELINUX= disabled#g config

[root@bogon selinux]# grep " SELINUX= disabled " config

改了配置文件需要重启

[root@bogon ~]# getenforce # 查看当前selinux状态

Enforcing

[root@bogon ~]# setenforce 0

[root@bogon ~]# getenforce #修改后的状态

Permissive

linux 有7种运行模式 0,1,2,3,4,5,6

查看启动模式

[root@bogon ~]# runlevel 查看当前系统的运行级别

[root@bogon ~]# init3 切换当前系统的运行级别 0为关机

N3 为命令模式

N5 桌面模式

[root@bogon ~]# vi /etc/inittab 编辑inittab id:3 为命令模式

[root@bogon ~]# init 0 关机

wind7 查看 系统开机启动 msconfig

linux 开机需要启动的4个服务

crond （定时任务），network（网络） ，rsyslog（系统日志功能） ，sshd（远程连接）

[root@bogon ~]# chkconfig --list |grep "3:on"

Note: This output shows SysV services only and does not include native

systemd services. SysV configuration data might be overridden by native

systemd configuration.

If you want to list systemd services use 'systemctl list-unit-files'.

To see services enabled on particular target use

'systemctl list-dependencies [target]'.

netconsole 0:off 1:off 2:off 3:off 4:off 5:off 6:off

network 0:off 1:off 2:on 3:on 4:on 5:on 6:off

[root@bogon ~]# chkconfig netconsole off # netconsole关闭

[root@bogon ~]# chkconfig netconsole on # netconsole启动

最小原则

安装系统最小

开机程序服务最小

操作命令最小原则

登录最小原则 ，用普通用户

权限最小原则， 尽量不用root权限账号操作

配置参数合理 最小化

ssh服务远程配置

windows默认端口

管理user ：administrator guest 远程端口 port 3389

linux 默认端口 远程 超级用户端口

user：root 普通用户 远程连接端口 port 22

修改 ssh 服务端口

[root@bogon /]# alias # 查看

alias cp='cp -i'

alias egrep='egrep --color=auto'

alias fgrep='fgrep --color=auto'

alias grep='grep --color=auto'

alias l.='ls -d .\* --color=auto'

alias ll='ls -l --color=auto'

alias ls='ls --color=auto'

alias mv='mv -i'

alias rm='rm -i'

alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'

[root@bogon /]# ls -l /etc/ssh/sshd\_config

-rw-------. 1 root root 3906 Sep 15 11:31 /etc/ssh/sshd\_config

修改前备份

[root@bogon /]# cd /etc/ssh

[root@bogon ssh]# cp sshd\_config sshd\_config.ori

[root@bogon ssh]# vi sshd\_config

修改Port 22 为 52113 端口范围为 0~65535 最大

#Protocol 2 表示 ssh2

#PermitRootLogin yes 改为NO 表示root 用户不能ssh远程登录可以在本地登录 为安全级别比较高

#UseDNS yes 改为NO

#PermitEmptyPasswords no 容许密码登录

修改以上四个点

[root@bogon ssh]# /etc/init.d/sshd reload # 重启

重新连接ssh 用新端口就OK 用root 不能连接了

关闭防火墙 iptables

[root@bogon ~]# /etc/init.d/iptables stop

查看连接的端口

[root@bogon ~]# netstat -an 查看网络状态

Active Internet connections (servers and established)

Proto Recv-Q Send-Q Local Address Foreign Address State

tcp 0 0 0.0.0.0:22 0.0.0.0:\* LISTEN

tcp 0 0 127.0.0.1:25 0.0.0.0:\* LISTEN

tcp 0 0 0.0.0.0:3306 0.0.0.0:\* LISTEN

tcp 0 52 192.168.1.108:22 192.168.1.101:58969 ESTABLISHED

tcp 1 0 192.168.1.108:39098 120.119.118.1:80 CLOSE\_WAIT

tcp 1 0 192.168.1.108:35486 140.113.168.126:80 CLOSE\_WAIT

tcp 0 0 192.168.1.108:3306 192.168.1.101:52007 ESTABLISHED

tcp 1 0 192.168.1.108:38384 67.219.148.138:80 CLOSE\_WAIT

tcp6 0 0 :::22 :::\* LISTEN

tcp6 0 0 ::1:25 :::\* LISTEN

udp 0 0 127.0.0.1:323 0.0.0.0:\*

udp6 0 0 ::1:323 :::\*

raw6 0 0 :::58 :::\* 7

[root@bogon ~]# netstat -an |grep 192.168.1.101 # 远程连接的2个端口 22 和3306

tcp 0 52 192.168.1.108:22 192.168.1.101:58969 ESTABLISHED

tcp 0 0 192.168.1.108:3306 192.168.1.101:52007 ESTABLISHED

vi编辑器

98gg 定位行到98行

yy 拷贝当前行

p 粘贴

权限 : 给普通用户给root的部分权限

su

sudo

[root@bogon ~]# visudo #增加一个用权限 相当于 vi /etc/sudoers

## Allow root to run any commands anywhere

root ALL=(ALL) ALL

增加 admin ALL=(ALL) /usr/sbin/useradd #说明只授权了添加用户的权限给这个用户

增加 admin ALL=(ALL) ALL # 表示和root 用户一样的权限

在 root 下 加用户信息和root的一样

就是设置权限的意思

http://blog.csdn.net/zwlq1314521/article/details/40508943

[admin@bogon /]$ sudo su -root

[sudo] password for admin: # 输入密码

查看当前的host 配置

[root@bogon ~]# cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

#zookeeper

192.168.1.102 bogon

环境变量设置

[root@bogon ~]# PATH=/zookeeper/:$PATH #增加环境变量临时生效 重新启动失效

[root@bogon ~]# echo $PATH # 查看环境变量

/download/zookeeper-3.4.9/bin:#ZOOPEEPER\_HOME/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/download/apache-tomcat-9.0.0.M26/bin:/root/bin

[root@bogon ~]# echo ' PATH=/zookeeper/:$PATH' >> /etc/profile #增加环境变量 要用单引号

[root@bogon ~]# source /etc/profile #增加环境变量后 设置永久生效

[root@bogon ~]# yum install lsof # 安装lsof 命令软件

[root@bogon ~]# lsof -i

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

chronyd 650 chrony 1u IPv4 15787 0t0 UDP localhost:323

chronyd 650 chrony 2u IPv6 15788 0t0 UDP localhost:323

sshd 987 root 3u IPv4 19176 0t0 TCP \*:ssh (LISTEN)

sshd 987 root 4u IPv6 19178 0t0 TCP \*:ssh (LISTEN)

mysqld 1222 mysql 14u IPv4 20679 0t0 TCP \*:mysql (LISTEN)

mysqld 1222 mysql 34u IPv4 28030 0t0 TCP bogon:mysql->bogon:52007 (ESTABLISHED)

master 1267 root 13u IPv4 20343 0t0 TCP localhost:smtp (LISTEN)

master 1267 root 14u IPv6 20344 0t0 TCP localhost:smtp (LISTEN)

sshd 14644 root 3u IPv4 186581 0t0 TCP bogon:ssh->bogon:54205 (ESTABLISHED)

字符集  
[root@bogon ~]# echo $LANG # 查看当前字符集

en\_US.UTF-8

[root@bogon ~]# LANG=EN # 临时调整字符集为英文 重启失效

[root@bogon ~]# locale

LANG=en\_US.UTF-8

LC\_CTYPE="en\_US.UTF-8"

LC\_NUMERIC="en\_US.UTF-8"

LC\_TIME="en\_US.UTF-8"

LC\_COLLATE="en\_US.UTF-8"

LC\_MONETARY="en\_US.UTF-8"

LC\_MESSAGES="en\_US.UTF-8"

LC\_PAPER="en\_US.UTF-8"

LC\_NAME="en\_US.UTF-8"

LC\_ADDRESS="en\_US.UTF-8"

LC\_TELEPHONE="en\_US.UTF-8"

LC\_MEASUREMENT="en\_US.UTF-8"

LC\_IDENTIFICATION="en\_US.UTF-8"

LC\_ALL=

服务器时间同步

linux 下的时间同步 ntp服务时间同步

[root@bogon ~]# which ntpdate # 查看是否有这个服务

/usr/sbin/ntpdate

[root@bogon ~]# /usr/sbin/ntpdate time.nist.gov # 同步互联网服务器的时间

14 Oct 12:37:39 ntpdate[14792]: adjust time server 216.229.0.179 offset -0.044196 sec

加大文件描述符

[root@bogon ~]# ulimit -n

1024

[root@bogon ~]# ulimit -HSn 65535 # 这样执行后退出就没有了

[root@bogon ~]# ulimit -n

65535

[root@bogon ~]# logout

bash: logout: not login shell: use `exit'

[root@bogon ~]# exit

exit

[root@bogon /]# ulimit -n

1024

[root@bogon /]# cat /etc/security/limits.conf

[root@bogon /]# vi /etc/security/limits.conf

\* - nofile 65535 # 在文本中加这行到最后

内核优化（未详学）

[root@bogon /]# cat /etc/sysctl.conf

[root@bogon /]# vi /etc/sysctl.conf

[root@bogon /]# df -h # 查看磁盘空间

Filesystem Size Used Avail Use% Mounted on

/dev/mapper/centos-root 18G 4.2G 14G 24% /

devtmpfs 226M 0 226M 0% /dev

tmpfs 237M 0 237M 0% /dev/shm

tmpfs 237M 8.7M 228M 4% /run

tmpfs 237M 0 237M 0% /sys/fs/cgroup

/dev/sda1 1014M 154M 861M 16% /boot

tmpfs 48M 0 48M 0% /run/user/0

[root@bogon /]# df -hi

Filesystem Inodes IUsed IFree IUse% Mounted on

/dev/mapper/centos-root 9.0M 108K 8.9M 2% /

devtmpfs 57K 396 56K 1% /dev

tmpfs 60K 1 60K 1% /dev/shm

tmpfs 60K 524 59K 1% /run

tmpfs 60K 16 60K 1% /sys/fs/cgroup

/dev/sda1 512K 333 512K 1% /boot

tmpfs 60K 1 60K 1% /run/user/0

防止黑客进入

隐藏系统版本：不让黑客看到系统版本信息

[root@bogon /]# cat /etc/issue

\S

Kernel \r on an \m

[root@bogon /]# cat /dev/null >/etc/issue # 隐藏 系统版本

锁定关键系统文件 ：不让黑客系统提权

[root@bogon /]# chattr +i /etc/passwd /etc/shadow /etc/group /etc/gshadow /etc/inittab

解锁关键系统文件

[root@bogon /]# chattr -i /etc/passwd /etc/shadow /etc/group /etc/gshadow /etc/inittab

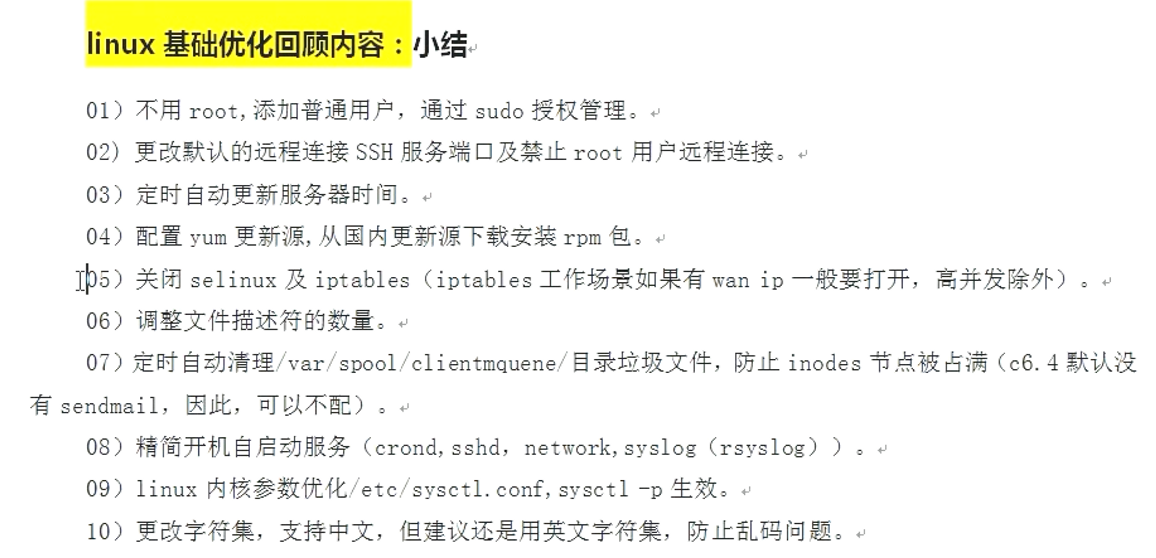
查看文件是否加锁

[root@bogon /]# lsattr /etc/passwd

----i----------e- /etc/passwd # 有一个 i为加锁了 ，如果------------/etc/passwd 这样是没有加锁

[root@bogon /]# lsattr /etc/passwd

---------------- /etc/passwd





linux 系统的目录结构

应用程序 /usr/bin

数据文件,帮助 usr/share

配置文件 /etc/

启动命令 /etc/init.d

查看挂载

[root@bogon /]# cat /etc/fstab

#

# /etc/fstab

# Created by anaconda on Fri Sep 15 17:18:44 2017

#

# Accessible filesystems, by reference, are maintained under '/dev/disk'

# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info

#

/dev/mapper/centos-root / xfs defaults 0 0

UUID=4fc4bd0e-0d4f-4a0f-88ea-8a11b4e84153 /boot xfs defaults 0 0

/dev/mapper/centos-swap swap swap defaults 0 0

目录结构介绍

[root@bogon ~]# which cp

alias cp='cp -i'

/usr/bin/cp

[root@bogon ~]# whereis -b rmdir

rmdir: /usr/bin/rmdir

[root@bogon ~]# whereis -b mkdir

mkdir: /usr/bin/mkdir

[root@bogon ~]# whereis -b cat

cat: /usr/bin/cat

[root@bogon ~]# ll /bin/ # 查看所有命令

[root@bogon /]# tree -L 1

.

系统目录 centos7

[root@bogon /]# tree -L 1 # 查询第一层结构

|-- bin -> usr/bin #常用二进制命令所在的目录

|-- boot # linux 的内核及引导系统程序所需的文件目录安装系统分区的时候一包要分一个boot分区

|-- dev # 设备文件目录 比如声卡 磁盘 光驱

|-- etc # 二进制安装包（yum, rmp）的配置文件默认路径 ，服务启动命令存放的目录

/etc/init.d(yum, rpm)

|-- home #普通用户的家目录默认数据存放目录

|-- lib -> usr/lib #库文件存放目录

|-- lib64 -> usr/lib64 #64位系统就有这个目录

|-- media #

|-- mnt #临时挂载目录 （文件系统的设备入口） u盘挂载这里

|-- opt # 可选的意思 有些软件包被安装在这里，

也是自定义软件包可以通过 ./configgure --prefix=/opt/目录

|-- proc #操作系统运行时，进程信息及内核信息 比如 cpu 硬盘分区 内存信息等

存放在这里

|-- root #超级权限用户 root 的家目录

|-- run #

|-- sbin -> usr/sbin #大多是设计系统管理的命令的存放 ，是超级管理权限用户root 的可执行命令存放地

|-- srv

|-- sys

|-- tmp #临时文件目录 有时用户运行程序的时候会产生临时文件 所有用户都有权限进入 这个目录

|-- usr # 系统存放程序的目录 比如 命令 帮助文档 等 /usr/bin 可以把自己的命令放在 这里

|-- var # 这个目录的内容是经常变动的 /var/log/messages 放系统日志的文件

查看系统挂载情况

[root@bogon media]# fdisk -l

Disk /dev/sda: 21.5 GB, 21474836480 bytes, 41943040 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk label type: dos

Disk identifier: 0x000dad6e

Device Boot Start End Blocks Id System

/dev/sda1 \* 2048 2099199 1048576 83 Linux

/dev/sda2 2099200 41943039 19921920 8e Linux LVM

常见的proc下的内容

[root@bogon proc]# cat /proc/meminfo

[root@bogon proc]# cat /proc/cpuinfo

[root@bogon proc]# cat /proc/loadavg

[root@bogon /]# halt 关机（init 0）

[root@bogon /]# reboot 重启（init 6）

[root@bogon /]# shutdown -r now（重启） -h now （关机）

系统日志

[root@bogon /]# cat /var/log/messages

安全日志 （哪些连接了的session 都在这里可以看到）

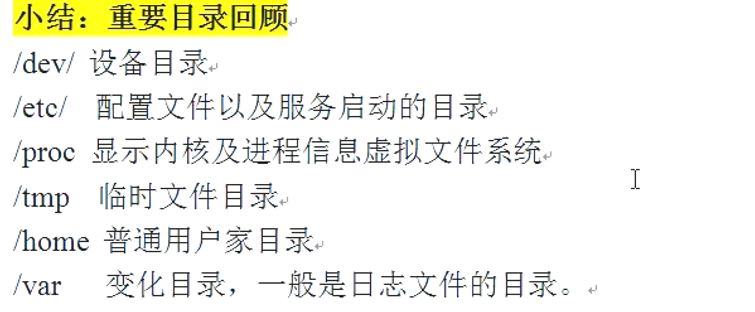
[root@bogon /]# cat /var/log/secure

Oct 14 20:42:12 bogon sshd[14644]: Disconnected from 192.168.1.101 port 54205

Oct 14 20:42:12 bogon sshd[14644]: pam\_unix(sshd:session): session closed for user root

Oct 14 20:42:12 bogon su: pam\_unix(su:session): session closed for user root

Oct 14 20:42:12 bogon su: pam\_unix(su:session): session closed for user admin



重要子目录

[root@bogon ~]# cat /etc/sysconfig/network-scripts/ifcfg-ens33 #查看网卡设置网络配置信息

|  |  |
| --- | --- |
| 目录 | 描述 |
| / | 第一层次结构的根、整个文件系统层次结构的[根目录](http://zh.wikipedia.org/wiki/%E6%A0%B9%E7%9B%AE%E5%BD%95)。 |
| /bin/ | 需要在[单用户模式](http://zh.wikipedia.org/w/index.php?title=%E5%8D%95%E7%94%A8%E6%88%B7%E6%A8%A1%E5%BC%8F&action=edit&redlink=1)可用的必要命令（[可执行文件](http://zh.wikipedia.org/wiki/%E5%8F%AF%E6%89%A7%E8%A1%8C%E6%96%87%E4%BB%B6)）；面向所有用户，例如：[cat](http://zh.wikipedia.org/wiki/Cat_(Unix))、[ls](http://zh.wikipedia.org/wiki/Ls)、[cp](http://zh.wikipedia.org/wiki/Cp_(Unix))，和/usr/bin类似。 |
| [/boot/](http://zh.wikipedia.org/w/index.php?title=/boot/&action=edit&redlink=1) | [引导程序](http://zh.wikipedia.org/wiki/%E5%BC%95%E5%AF%BC%E7%A8%8B%E5%BA%8F)文件，例如：[kernel](http://zh.wikipedia.org/wiki/%E5%86%85%E6%A0%B8)、[initrd](http://zh.wikipedia.org/wiki/Initrd)；时常是一个单独的分区[[6]](http://zh.wikipedia.org/wiki/%E6%96%87%E4%BB%B6%E7%B3%BB%E7%BB%9F%E5%B1%82%E6%AC%A1%E7%BB%93%E6%9E%84%E6%A0%87%E5%87%86#cite_note-6) |
| **/dev/** | 必要[设备](http://zh.wikipedia.org/wiki/%E8%AE%BE%E5%A4%87%E6%96%87%E4%BB%B6%E7%B3%BB%E7%BB%9F), 例如：, [/dev/null](http://zh.wikipedia.org/wiki/dev/null). |
| **/etc/** | 特定主机，系统范围内的[配置文件](http://zh.wikipedia.org/w/index.php?title=%E9%85%8D%E7%BD%AE%E6%96%87%E4%BB%B6&action=edit&redlink=1)。  关于这个名称目前有争议。在贝尔实验室关于UNIX实现文档的早期版本中，/etc 被称为[/etcetra 目录](http://zh.wikipedia.org/w/index.php?title=Et_cetera&action=edit&redlink=1)，[[7]](http://zh.wikipedia.org/wiki/%E6%96%87%E4%BB%B6%E7%B3%BB%E7%BB%9F%E5%B1%82%E6%AC%A1%E7%BB%93%E6%9E%84%E6%A0%87%E5%87%86#cite_note-7)这是由于过去此目录中存放所有不属于别处的所有东西（然而，FHS限制/etc存放静态配置文件，不能包含二进制文件）。[[8]](http://zh.wikipedia.org/wiki/%E6%96%87%E4%BB%B6%E7%B3%BB%E7%BB%9F%E5%B1%82%E6%AC%A1%E7%BB%93%E6%9E%84%E6%A0%87%E5%87%86#cite_note-8)自从早期文档出版以来，目录名称已被以各种方式重新称呼。最近的解释包括[反向缩略语](http://zh.wikipedia.org/w/index.php?title=%E5%8F%8D%E5%90%91%E7%BC%A9%E7%95%A5%E8%AF%AD&action=edit&redlink=1)如："可编辑的文本配置"（英文 "Editable Text Configuration"）或"扩展工具箱"（英文 "Extended Tool Chest")。[[9]](http://zh.wikipedia.org/wiki/%E6%96%87%E4%BB%B6%E7%B3%BB%E7%BB%9F%E5%B1%82%E6%AC%A1%E7%BB%93%E6%9E%84%E6%A0%87%E5%87%86#cite_note-9) |
| /etc/opt/ | /opt/的配置文件 |
| /etc/X11/ | [X\_Window系统](http://zh.wikipedia.org/wiki/X_Window%E7%B3%BB%E7%BB%9F)(版本11)的配置文件 |
| /etc/sgml/ | [SGML](http://zh.wikipedia.org/wiki/SGML)的配置文件 |
| /etc/xml/ | [XML](http://zh.wikipedia.org/wiki/XML)的配置文件 |
| **/home/** | 用户的[家目录](http://zh.wikipedia.org/wiki/%E5%AE%B6%E7%9B%AE%E5%BD%95)，包含保存的文件、个人设置等，一般为单独的分区。 |
| /lib/ | /bin/ and /sbin/中二进制文件必要的[库](http://zh.wikipedia.org/wiki/%E5%BA%93)文件。 |
| /media/ | 可移除媒体(如[CD-ROM](http://zh.wikipedia.org/wiki/CD-ROM))的挂载点 (在FHS-2.3中出现)。 |
| /lost+found | 在ext3文件系统中，当系统意外崩溃或机器意外关机，会产生一些文件碎片在这里。当系统在开机启动的过程中fsck工具会检查这里，并修复已经损坏的文件系统。当系统发生问题。可能会有文件被移动到这个目录中，可能需要用手工的方式来修复，或移到文件到原来的位置上。 |
| /mnt/ | 临时[挂载](http://zh.wikipedia.org/w/index.php?title=%E6%8C%82%E8%BD%BD&action=edit&redlink=1)的文件系统。比如cdrom,u盘等，直接插入光驱无法使用，要先挂载后使用 |
| /opt/ | 可选[应用软件](http://zh.wikipedia.org/wiki/%E5%BA%94%E7%94%A8%E8%BD%AF%E4%BB%B6)[包](http://zh.wikipedia.org/wiki/%E8%BD%AF%E4%BB%B6%E5%8C%85)。 |
| **/proc/** | 虚拟[文件系统](http://zh.wikipedia.org/wiki/%E6%96%87%E4%BB%B6%E7%B3%BB%E7%BB%9F)，将[内核](http://zh.wikipedia.org/wiki/%E5%86%85%E6%A0%B8)与[进程](http://zh.wikipedia.org/wiki/%E8%BF%9B%E7%A8%8B)状态归档为文本文件（系统信息都存放这目录下）。例如：uptime、 network。在Linux中，对应[Procfs](http://zh.wikipedia.org/wiki/Procfs)格式挂载。该目录下文件只能看不能改（包括root） |
| /root/ | [超级用户](http://zh.wikipedia.org/wiki/%E8%B6%85%E7%BA%A7%E7%94%A8%E6%88%B7)的[家目录](http://zh.wikipedia.org/wiki/%E5%AE%B6%E7%9B%AE%E5%BD%95) |
| **/sbin/** | 必要的系统二进制文件，例如： init、 ip、 mount。sbin目录下的命令，普通用户都执行不了。 |
| /srv/ | 站点的具体[数据](http://zh.wikipedia.org/wiki/%E6%95%B0%E6%8D%AE)，由系统提供。 |
| /tmp/ | 临时文件(参见 /var/tmp)，在系统重启时目录中文件不会被保留。 |
| /usr/ | 默认软件都会存于该目录下。用于存储只读用户数据的第二层次；包含绝大多数的([多](http://zh.wikipedia.org/wiki/%E5%A4%9A%E7%94%A8%E6%88%B7))用户工具和应用程序。 |
| **/var/** | 变量文件——在正常运行的系统中其内容不断变化的文件，如日志，脱机文件和临时电子邮件文件。有时是一个单独的分区。如果不单独分区，有可能会把整个分区充满。如果单独分区，给大给小都不合适。 |

**3、/****etc/目录**

特定主机系统范围内的[配置文件](http://zh.wikipedia.org/w/index.php?title=%E9%85%8D%E7%BD%AE%E6%96%87%E4%BB%B6&action=edit&redlink=1)。

|  |  |
| --- | --- |
| 目录 | 描述 |
| **/etc/rc /etc/rc.d**  **/etc/rc\*.d** | 启动、或改变运行级时运行的scripts或scripts的目录. |
| /etc/hosts | 本地域名解析文件 |
| **/etc/sysconfig/network** | IP、掩码、网关、主机名配置 |
| **/etc/resolv.conf** | DNS服务器配置 |
| **/etc/fstab** | 开机自动挂载系统，所有分区开机都会自动挂载 |
| **/etc/inittab** | 设定系统启动时Init进程将把系统设置成什么样的runlevel及加载相关的启动文件配置 |
| **/etc/exports** | 设置NFS系统用的配置文件路径 |
| **/etc/init.d** | 这个目录来存放系统启动脚本 |
| **/etc/profile,**/etc/csh.login,  /etc/csh.cshrc | **全局系统环境配置变量** |
| **/etc/issue** | 认证前的输出信息，默认输出版本内核信息 |
| /etc/motd | 设置认证后的输出信息， |
| /etc/mtab | 当前安装的文件系统列表.由scripts初始化，并由mount 命令自动更新.需要一个当前安装的文件系统的列表时使用，例如df 命令 |
| **/etc/group** | 类似/etc/passwd ，但说明的不是用户而是组. |
| **/etc/passwd** | 用户数据库，其中的域给出了用户名、真实姓名、家目录、加密的口令和用户的其他信息. |
| **/etc/shadow** | 在安装了影子口令软件的系统上的影子口令文件.影子口令文件将/etc/passwd 文件中的加密口令移动到/etc/shadow 中，而后者只对root可读.这使破译口令更困难. |
| **/etc/sudoers** | 可以sudo命令的配置文件 |
| **/etc/syslog.conf** | 系统日志参数配置 |
| /etc/login.defs | 设置用户帐号限制的文件 |
| /etc/securetty | 确认安全终端，即哪个终端允许root登录.一般只列出虚拟控制台，这样就不可能(至少很困难)通过modem或网络闯入系统并得到超级用户特权. |
| /etc/printcap | 类似/etc/termcap ，但针对打印机.语法不同. |
| /etc/shells | 列出可信任的shell.chsh 命令允许用户在本文件指定范围内改变登录shell.提供一台机器FTP服务的服务进程ftpd 检查用户shell是否列在 /etc/shells 文件中，如果不是将不允许该用户登录. |
| /etc/xinetd.d | 如果服务器是通过xinetd模式运行的，它的脚本要放在这个目录下。有些系统没有这个目录，比如Slackware，有些老的版本也没有。在Redhat Fedora中比较新的版本中存在。 |
| /etc/opt/ | /opt/的配置文件 |
| /etc/X11/ | [X\_Window系统](http://zh.wikipedia.org/wiki/X_Window%E7%B3%BB%E7%BB%9F)(版本11)的配置文件 |
| /etc/sgml/ | [SGML](http://zh.wikipedia.org/wiki/SGML)的配置文件 |
| /etc/xml/ | [XML](http://zh.wikipedia.org/wiki/XML)的配置文件 |
| **/etc/skel/** | 默认创建用户时，把该目录拷贝到家目录下 |

**4、/****usr/目录**

默认软件都会存于该目录下。用于存储只读用户数据的第二层次；包含绝大多数的用户工具和应用程序。

|  |  |
| --- | --- |
| 目录 | 描述 |
| /usr/X11R6 | 存放X-Windows的目录； |
| /usr/games | 存放着XteamLinux自带的小游戏； |
| /usr/doc | Linux技术文档； |
| /usr/include | 用来存放Linux下开发和编译应用程序所需要的头文件； |
| /usr/lib | 存放一些常用的动态链接共享库和静态档案库； |
| /usr/man | 帮助文档所在的目录； |
| /usr/src | Linux开放的源代码，就存在这个目录，爱好者们别放过哦； |
| /usr/bin/ | 非必要[可执行文件](http://zh.wikipedia.org/wiki/%E5%8F%AF%E6%89%A7%E8%A1%8C%E6%96%87%E4%BB%B6) (在[单用户模式](http://zh.wikipedia.org/w/index.php?title=%E5%8D%95%E7%94%A8%E6%88%B7%E6%A8%A1%E5%BC%8F&action=edit&redlink=1)中不需要)；面向所有用户。 |
| /usr/lib/ | /usr/bin/和/usr/sbin/中二进制文件的[库](http://zh.wikipedia.org/wiki/%E5%BA%93)。 |
| /usr/sbin/ | 非必要的系统二进制文件，例如：大量[网络服务](http://zh.wikipedia.org/wiki/%E7%BD%91%E7%BB%9C%E6%9C%8D%E5%8A%A1)的[守护进程](http://zh.wikipedia.org/wiki/%E5%AE%88%E6%8A%A4%E8%BF%9B%E7%A8%8B)。 |
| /usr/share/ | 体系结构无关（共享）数据。 |
| /usr/X11R6/ | [X Window系统](http://zh.wikipedia.org/wiki/X_Window%E7%B3%BB%E7%BB%9F)版本 11, Release 6. |
| /usr/local/ | 本地数据的第三层次，具体到本台主机。通常而言有进一步的子目录，例如：bin/、lib/、share/.这是提供给一般用户的/usr目录，在这里安装一般的应用软件； |

**5、/****var/目录**

/var 包括系统一般运行时要改变的数据.每个系统是特定的，即不通过网络与其他计算机共享.

|  |  |
| --- | --- |
| 目录 | 描述 |
| /var/log/message | 日志信息，按周自动轮询 |
| /var/spool/cron/root | 定时器配置文件目录，默认按用户命名 |
| /var/log/secure | 记录登陆系统存取信息的文件，不管认证成功还是认证失败都会记录 |
| /var/log/wtmp | 记录登陆者信息的文件，last,who,w命令信息来源于此 |
| /var/spool/clientmqueue/ | 当邮件服务未开启时，所有应发给系统管理员的邮件都将堆放在此 |
| /var/spool/mail/ | 邮件目录 |
| /var/tmp | 比/tmp 允许的大或需要存在较长时间的临时文件. (虽然系统管理员可能不允许/var/tmp 有很旧的文件.) |
| /var/lib | 系统正常运行时要改变的文件. |
| /var/local | /usr/local 中安装的程序的可变数据(即系统管理员安装的程序).注意，如果必要，即使本地安装的程序也会使用其他/var 目录，例如/var/lock . |
| /var/lock | 锁定文件.许多程序遵循在/var/lock 中产生一个锁定文件的约定，以支持他们正在使用某个特定的设备或文件.其他程序注意到这个锁定文件，将不试图使用这个设备或文件. |
| /var/log/ | 各种程序的Log文件，特别是login   (/var/log/wtmp log所有到系统的登录和注销) 和syslog (/var/log/messages 里存储所有核心和系统程序信息. /var/log 里的文件经常不确定地增长，应该定期清除. |
| /var/run | 保存到下次引导前有效的关于系统的信息文件.例如， /var/run/utmp 包含当前登录的用户的信息. |
| /var/cache/ | 应用程序缓存数据。这些数据是在本地生成的一个耗时的I/O或计算结果。应用程序必须能够再生或恢复数据。缓存的文件可以被删除而不导致数据丢失。 |

**6、/****proc/目录**

虚拟[文件系统](http://zh.wikipedia.org/wiki/%E6%96%87%E4%BB%B6%E7%B3%BB%E7%BB%9F)，将[内核](http://zh.wikipedia.org/wiki/%E5%86%85%E6%A0%B8)与[进程](http://zh.wikipedia.org/wiki/%E8%BF%9B%E7%A8%8B)状态归档为文本文件（系统信息都存放这目录下）。

例如：uptime、 network。在Linux中，对应[Procfs](http://zh.wikipedia.org/wiki/Procfs)格式挂载。该目录下文件只能看不能改（包括root）

|  |  |
| --- | --- |
| 目录 | 描述 |
| **/proc/meminfo** | 查看内存信息 |
| **/proc/loadavg** | 还记得 top 以及 uptime 吧？没错！上头的三个平均数值就是记录在此！ |
| **/proc/uptime** | 就是用 uptime 的时候，会出现的资讯啦！ |
| **/proc/cpuinfo** | 关于处理器的信息，如类型、厂家、型号和性能等。 |
| /proc/cmdline | 加载 kernel 时所下达的相关参数！查阅此文件，可了解系统是如何启动的！ |
| /proc/filesystems | 目前系统已经加载的文件系统罗！ |
| /proc/interrupts | 目前系统上面的 IRQ 分配状态。 |
| /proc/ioports | 目前系统上面各个装置所配置的 I/O 位址。 |
| /proc/kcore | 这个就是内存的大小啦！好大对吧！但是不要读他啦！ |
| /proc/modules | 目前我们的 Linux 已经加载的模块列表，也可以想成是驱动程序啦！ |
| /proc/mounts | 系统已经挂载的数据，就是用 mount 这个命令呼叫出来的数据啦！ |
| /proc/swaps | 到底系统挂加载的内存在哪里？呵呵！使用掉的 partition 就记录在此啦！ |
| /proc/partitions | 使用 fdisk -l 会出现目前所有的 partition 吧？在这个文件当中也有纪录喔！ |
| /proc/pci | 在 PCI 汇流排上面，每个装置的详细情况！可用 lspci 来查阅！ |
| /proc/version | 核心的版本，就是用 uname -a 显示的内容啦！ |
| /proc/bus/\* | 一些汇流排的装置，还有 U盘的装置也记录在此喔！ |

**7、/****dev/目录**

设备文件分为两种：块设备文件(b)和字符设备文件(c)  
设备文件一般存放在/dev目录下，

对常见设备文件作如下说明：

|  |  |
| --- | --- |
| 目录 | 描述 |
| /dev/hd[a-t] | IDE设备 |
| /dev/sd[a-z] | SCSI设备 |
| /dev/fd[0-7] | 标准软驱 |
| /dev/md[0-31] | 软raid设备 |
| /dev/loop[0-7] | 本地回环设备 |
| /dev/ram[0-15] | 内存 |
| /dev/null | 无限数据接收设备,相当于黑洞 |
| /dev/zero | 无限零资源 |
| /dev/tty[0-63] | 虚拟终端 |
| /dev/ttyS[0-3] | 串口 |
| /dev/lp[0-3] | 并口 |
| /dev/console | 控制台 |
| /dev/fb[0-31] | framebuffer |
| /dev/cdrom | => /dev/hdc |
| /dev/modem | => /dev/ttyS[0-9] |
| /dev/pilot | => /dev/ttyS[0-9] |
| /dev/random | 随机数设备 |
| /dev/urandom | 随机数设备 |

[root@bogon ~]# last 记录登录信息

[root@bogon log]# lastlog 查看最近登录过的用户

文件类型

如 file.txt file.tar.gz

第一个属性为 ‘—’的都是普通文件

-rw-r--r-- 1 root root 66 Oct 14 07:16 oldboy.txt 普通文件删除用 rm 命令

drwxr-xr-x. 4 root root 34 Sep 15 17:19 src

lrwxrwxrwx. 1 root root 10 Sep 15 17:19 tmp -> ../var/tmp

[root@bogon data]# file oldboy.txt # 普通文本文件

oldboy.txt: ISO-8859 text

[root@bogon xml]# file /var/log/wtmp #数据文件

/var/log/wtmp: data

[root@bogon xml]# ls -ld /etc

drwxr-xr-x. 99 root root 8192 Oct 16 06:38 /etc

[root@bogon /]# ls -p

bin data/ download/ home/ lib64 mnt/ proc/ run/ srv/ tmp/ var/

boot/ dev/ etc/ lib media/ opt/ root/ sbin sys/ usr/

[root@bogon /]# ls -F

bin@ data/ download/ home/ lib64@ mnt/ proc/ run/ srv/ tmp/ var/

boot/ dev/ etc/ lib@ media/ opt/ root/ sbin@ sys/ usr/

[root@bogon /]#

[root@bogon /]# ls -l /dev/tty

crw-rw-rw- 1 root tty 5, 0 Oct 14 11:54 /dev/tty

[root@bogon /]# ls -l /dev/sda\*

brw-rw---- 1 root disk 8, 0 Oct 13 06:41 /dev/sda

brw-rw---- 1 root disk 8, 1 Oct 13 06:41 /dev/sda1

brw-rw---- 1 root disk 8, 2 Oct 13 06:41 /dev/sda2

创建一个字符设备

console 是设备的名字

c 指字符设备

5是该设备在major.h中的定义的标记

1是第一个子设备

[root@bogon data]# mknod oldboy c 5 1

[root@bogon data]# ls -l

total 4

crw-r--r-- 1 root root 5, 1 Oct 17 06:37 oldboy

符号链接文件

[root@bogon data]# ls -ld /etc/init.d

lrwxrwxrwx. 1 root root 11 Sep 15 17:19 /etc/init.d -> rc.d/init.d

查询管道文件

[root@bogon /]# find / -type p|xargs ls -l

prw------- 1 root root 0 Oct 13 06:41 /run/dmeventd-client

prw------- 1 root root 0 Oct 13 06:41 /run/dmeventd-server

prw------- 1 root root 0 Oct 14 11:07 /run/systemd/ask-password-block/136:2

prw------- 1 root root 0 Oct 13 07:32 /run/systemd/ask-password-block/4:1

prw------- 1 root root 0 Oct 13 06:43 /run/systemd/inhibit/1.ref

prw------- 1 root root 0 Oct 13 06:41 /run/systemd/initctl/fifo

prw------- 1 root root 0 Oct 13 06:58 /run/systemd/sessions/1.ref

prw------- 1 root root 0 Oct 16 06:38 /run/systemd/sessions/226.ref

[root@bogon /]# find / -type p

/run/dmeventd-client

/run/dmeventd-server

/run/systemd/ask-password-block/136:2

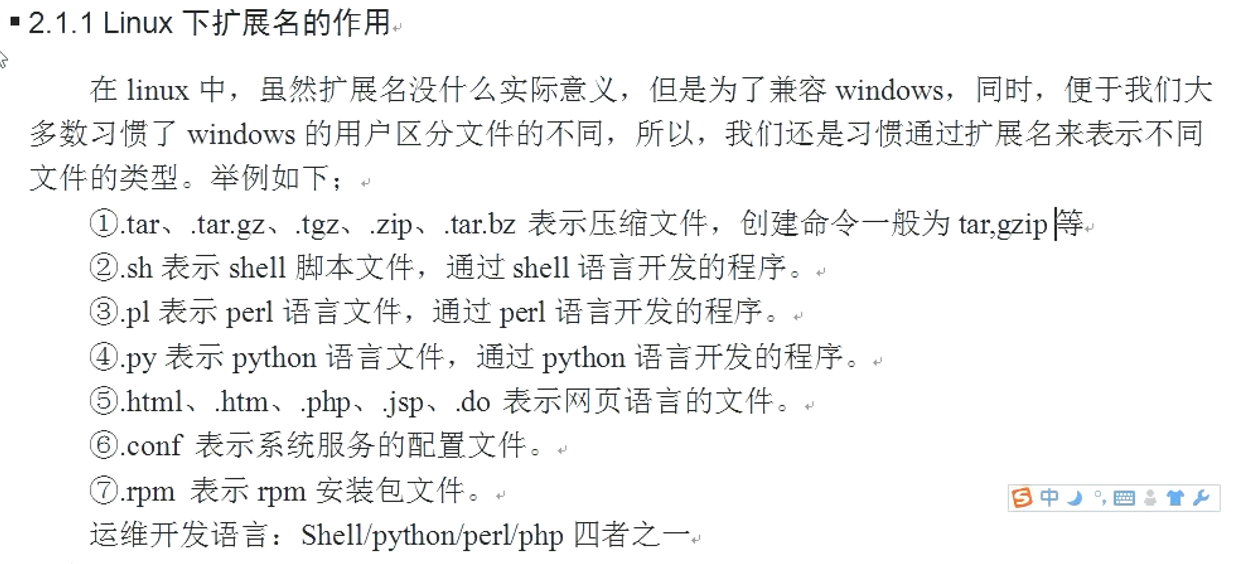
/run/systemd/ask-password-block/4:1

/run/systemd/inhibit/1.ref

/run/systemd/sessions/226.ref

/run/systemd/sessions/1.ref

/run/systemd/initctl/fifo



[root@bogon /]# ls -lhi -i 在前面显示的数字 -h 格式化K 大小

total 24K

112 lrwxrwxrwx. 1 root root 7 Sep 15 17:18 bin -> usr/bin

64 dr-xr-xr-x. 5 root root 4.0K Sep 15 10:43 boot

25874 drwxr-xr-x 2 root root 38 Oct 17 06:37 data

3 drwxr-xr-x 20 root root 3.3K Oct 13 06:43 dev

5890 drwxr-xr-x. 6 root root 4.0K Sep 27 09:06 download

16777281 drwxr-xr-x. 99 root root 8.0K Oct 16 06:38 etc

33649685 drwxr-xr-x. 3 root root 19 Oct 14 08:30 home

116 lrwxrwxrwx. 1 root root 7 Sep 15 17:18 lib -> usr/lib

82 lrwxrwxrwx. 1 root root 9 Sep 15 17:18 lib64 -> usr/lib64

50332066 drwxr-xr-x. 2 root root 6 Nov 5 2016 media

83 drwxr-xr-x. 2 root root 6 Nov 5 2016 mnt

16778357 drwxr-xr-x. 2 root root 6 Nov 5 2016 opt

1 dr-xr-xr-x 115 root root 0 Oct 13 06:41 proc

33574977 dr-xr-x---. 2 root root 198 Sep 27 09:27 root

8164 drwxr-xr-x 26 root root 780 Oct 16 06:38 run

117 lrwxrwxrwx. 1 root root 8 Sep 15 17:18 sbin -> usr/sbin

33649686 drwxr-xr-x. 2 root root 6 Nov 5 2016 srv

1 dr-xr-xr-x 13 root root 0 Oct 13 06:41 sys

16777288 drwxrwxrwt. 13 root root 4.0K Oct 17 03:13 tmp

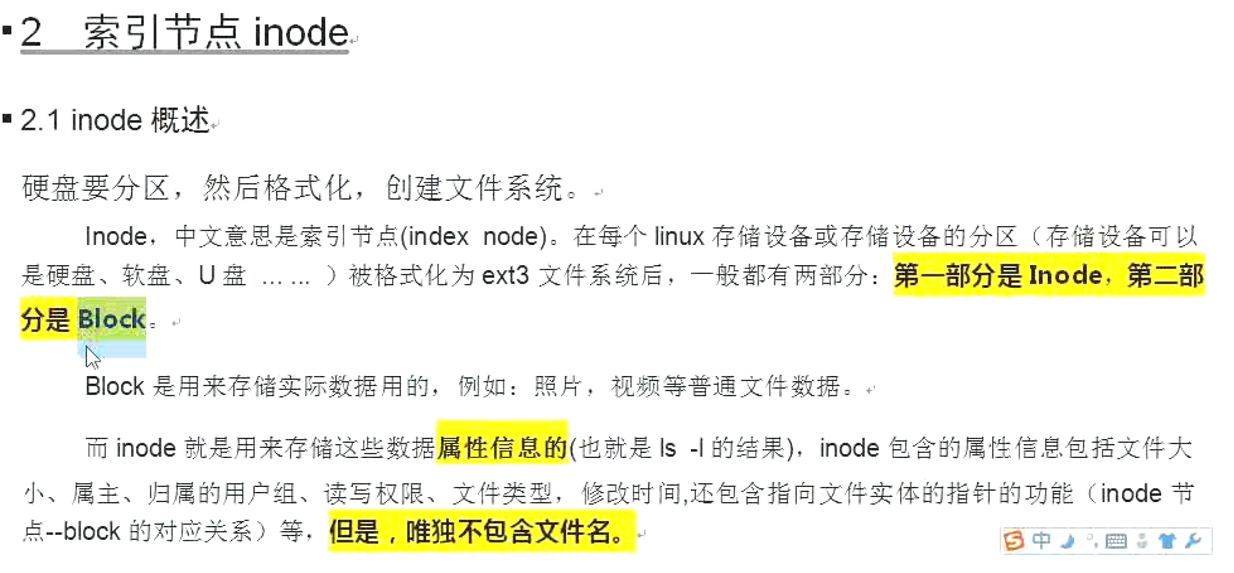
33649634 drwxr-xr-x. 13 root root 155 Sep 15 17:19 usr

50331713 drwxr-xr-x. 20 root root 278 Sep 15 09:54 var

文件属性说明







[root@bogon data]# stat oldboy.txt

File: 'oldboy.txt'

Size: 66 Blocks: 8 IO Block: 4096 regular file

Device: fd00h/64768d Inode: 25876 Links: 1

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2017-10-16 06:43:56.993818298 -0400

Modify: 2017-10-14 07:16:25.485826228 -0400

Change: 2017-10-14 07:16:25.485826228 -0400

Birth: -

查看Inode 容量

[root@bogon data]# df -i

Filesystem Inodes IUsed IFree IUse% Mounted on

/dev/mapper/centos-root 9433088 110572 9322516 2% /

devtmpfs 57739 396 57343 1% /dev

tmpfs 60452 1 60451 1% /dev/shm

tmpfs 60452 525 59927 1% /run

tmpfs 60452 16 60436 1% /sys/fs/cgroup

/dev/sda1 524288 333 523955 1% /boot

tmpfs 60452 1 60451 1% /run/user/0

查看磁盘容量

[root@bogon data]# df -h

Filesystem Size Used Avail Use% Mounted on

/dev/mapper/centos-root 18G 4.2G 14G 24% /

devtmpfs 226M 0 226M 0% /dev

tmpfs 237M 0 237M 0% /dev/shm

tmpfs 237M 13M 224M 6% /run

tmpfs 237M 0 237M 0% /sys/fs/cgroup

/dev/sda1 1014M 154M 861M 16% /boot

tmpfs 48M 0 48M 0% /run/user/0

[root@bogon /]# man find # 然后用 /type 再用小写的n 查找 到如下

-type c

File is of type c:

b block (buffered) special

c character (unbuffered) special

d directory

p named pipe (FIFO)

f regular file

l symbolic link; this is never true if the -L option or the -fol鈁0m

low option is in effect, unless the symbolic link is broken.

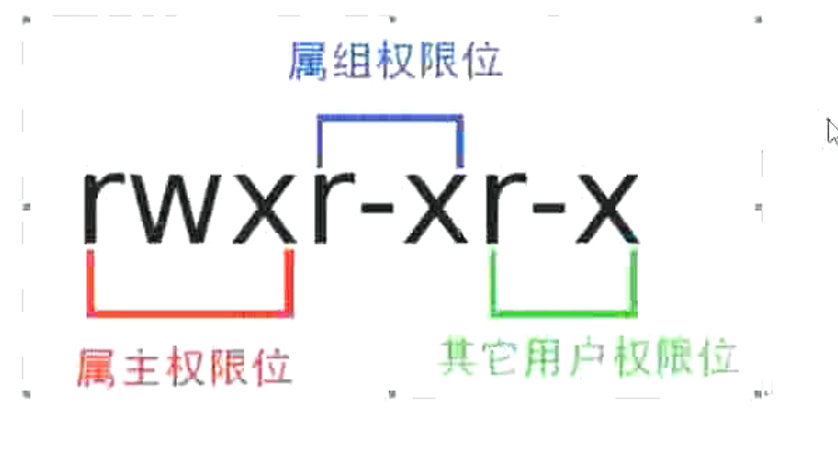
If you want to search for symbolic links when -L is in effect,

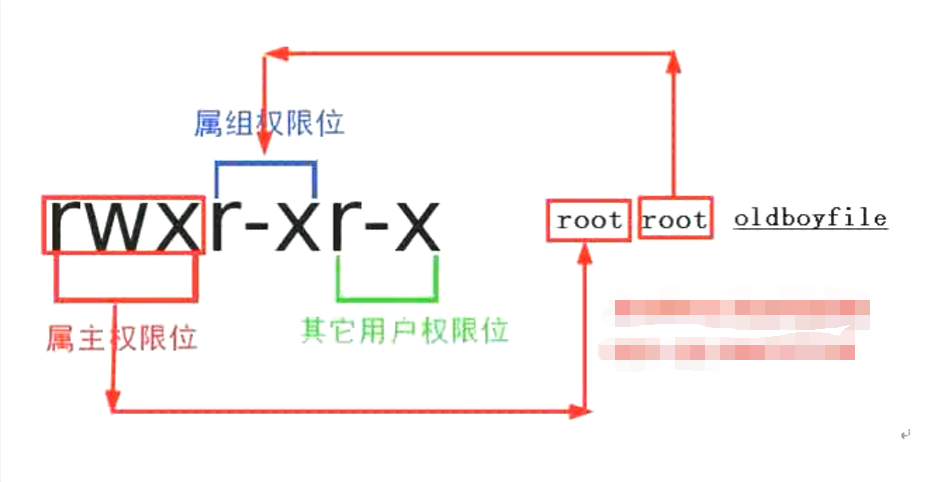
use -xtype.

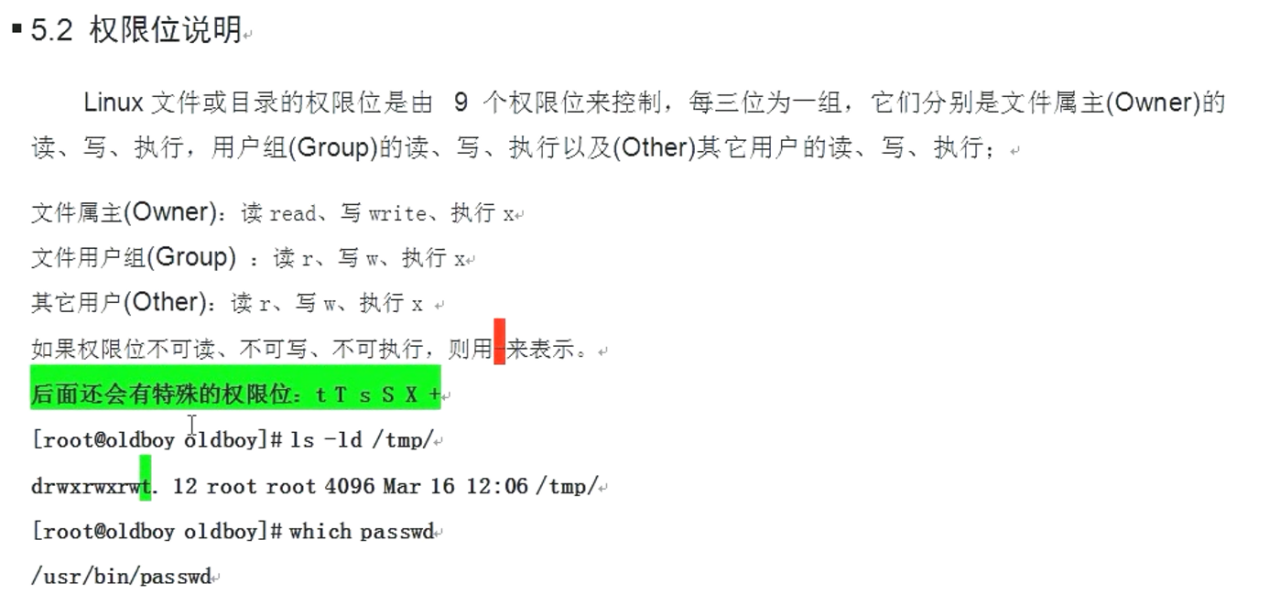
s socket

D door (Solaris)

文件权限 linux 看到的是9位权限 ， 实际是12位权限







linux 系统中链接分为两种，一种是硬链接（ln 是创建硬链接）和软链接（ln -s 创建软链接）

软链接就相当于快捷方式

硬链接总结

1.硬链接文件是具有相同inode 节点的不同文件

2 删除硬链接文件或者删除源文件之一， 文件实体并未被删除

3只有删除了源文件及所有对应的硬链接文件，文件实体才会被删除

4 当所有的硬链接文件及源文件被删除后，在存放新的数据会占用这个文件的空间或者磁盘会做fsck检查的时候 ，数据会被回收 。如果关机后没有再写入数据是可以恢复的，再写入数据后就不能恢复了

5硬链接文件是文件的油一个入口

6可以通过给文件设置硬链接文件来防止中要文件被误删

7通过执行命令 “ ln 源文件 硬链接文件 ” 既可创建硬链接

8 rm -f 硬链接文件 删除

软链接

1软链接类似windows里的快捷方式 软连接里面存放的是源文件的路径，并且执行源文件

2 删除源文件，软链接文件依然存在，但无法访问指向的路径内容

3执行命令 ln -s 源文件 软链接文件 即可完成创建软链接

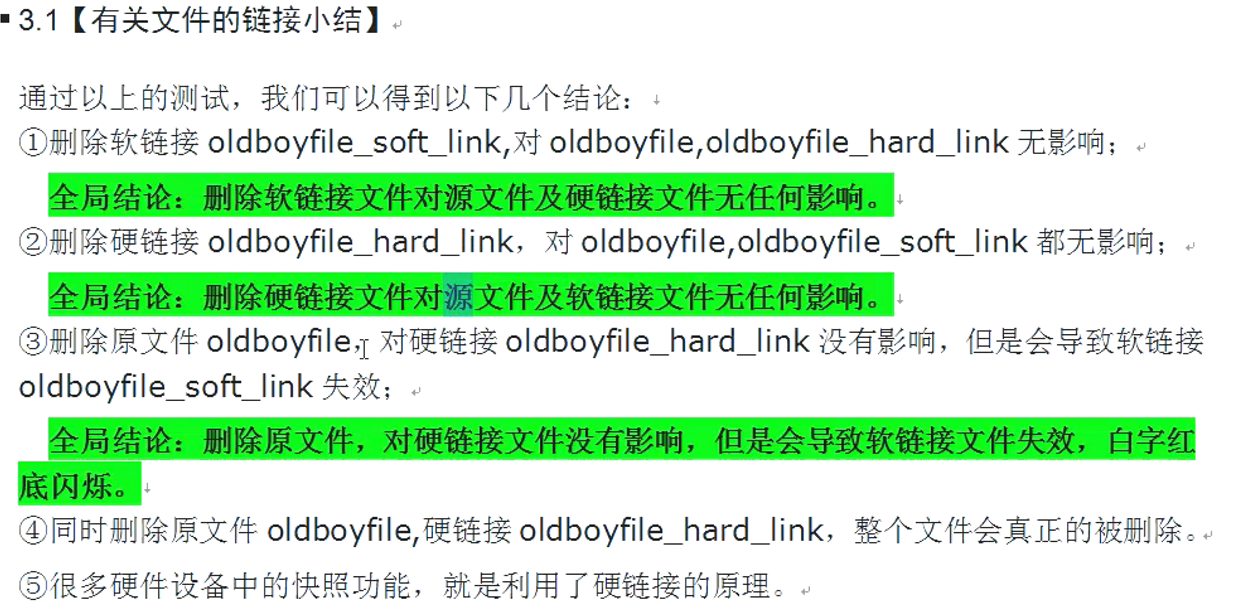
4软链接和源文件是不同类型的文件 也是不同的文件 inode号不同

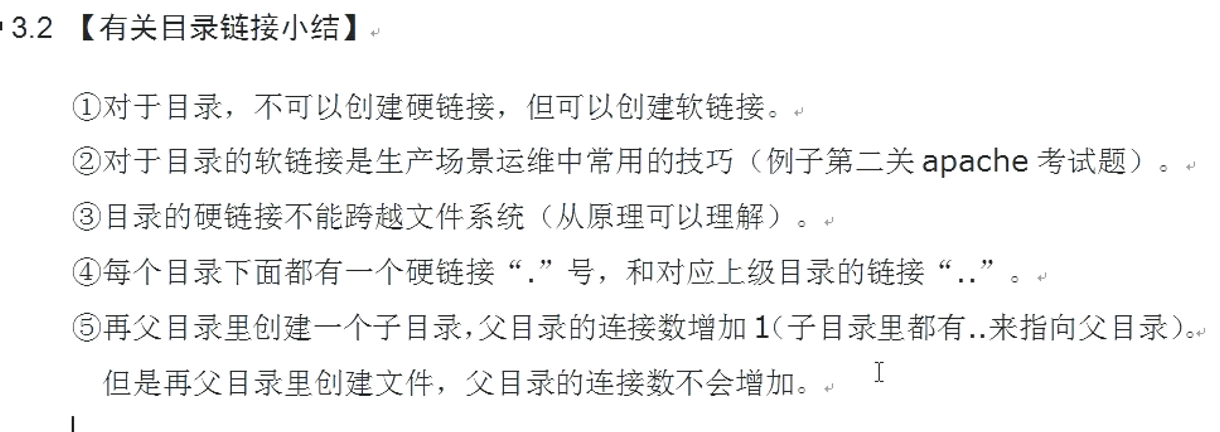
5 rm -f 软链接文件 删除

查看软链接

[root@bogon /]# readlink lib

usr/lib





空间不释放有3中情况

1有硬链接

2有进程文件在使用

3磁盘空余空间维护出现问题

可以看bash 的内置命令

[root@bogon /]# man bash

[root@bogon /]# help cd

检查是不是内置命令用: type+命令名

linux用户和用户组

用户角色的划分,用户的角色是通过UID和GID识别的

UID相当于身份证一样

[root@bogon /]# id admin

uid=1000(admin) gid=1000(admin) groups=1000(admin)

[root@bogon /]# id root # root均为0

uid=0(root) gid=0(root) groups=0(root)

sudo提权

[root@bogon /]# cat /etc/passwd

root:x:0:0:root:/root:/bin/bash # bash 是可以登陆

bin:x:1:1:bin:/bin:/sbin/nologin

daemon:x:2:2:daemon:/sbin:/sbin/nologin ## nologin 是无法登陆的

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:12:100:games:/usr/games:/sbin/nologin

ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin

nobody:x:99:99:Nobody:/:/sbin/nologin

systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin

dbus:x:81:81:System message bus:/:/sbin/nologin

polkitd:x:999:997:User for polkitd:/:/sbin/nologin

postfix:x:89:89::/var/spool/postfix:/sbin/nologin

sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin

chrony:x:998:996::/var/lib/chrony:/sbin/nologin

mysql:x:27:27:MariaDB Server:/var/lib/mysql:/sbin/nologin

dockerroot:x:997:994:Docker User:/var/lib/docker:/sbin/nologin

dhcpd:x:177:177:DHCP server:/:/sbin/nologin

admin:x:1000:1000::/home/admin:/bin/bash

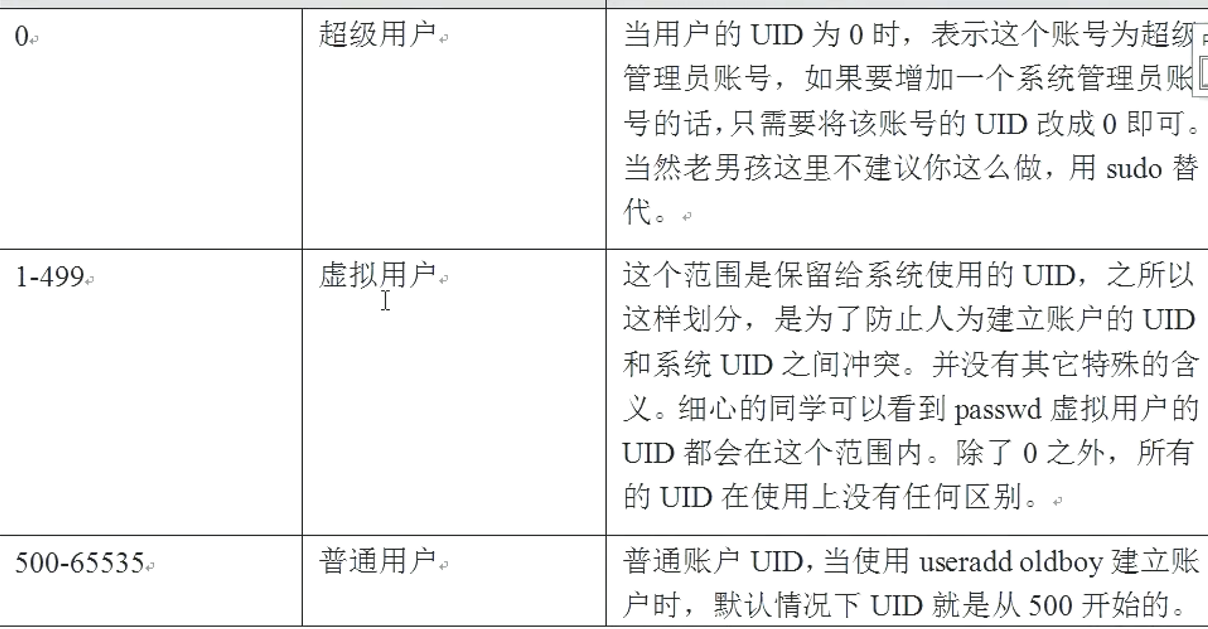
saslauth:x:996:76:Saslauthd user:/run/saslauthd:/sbin/nologin

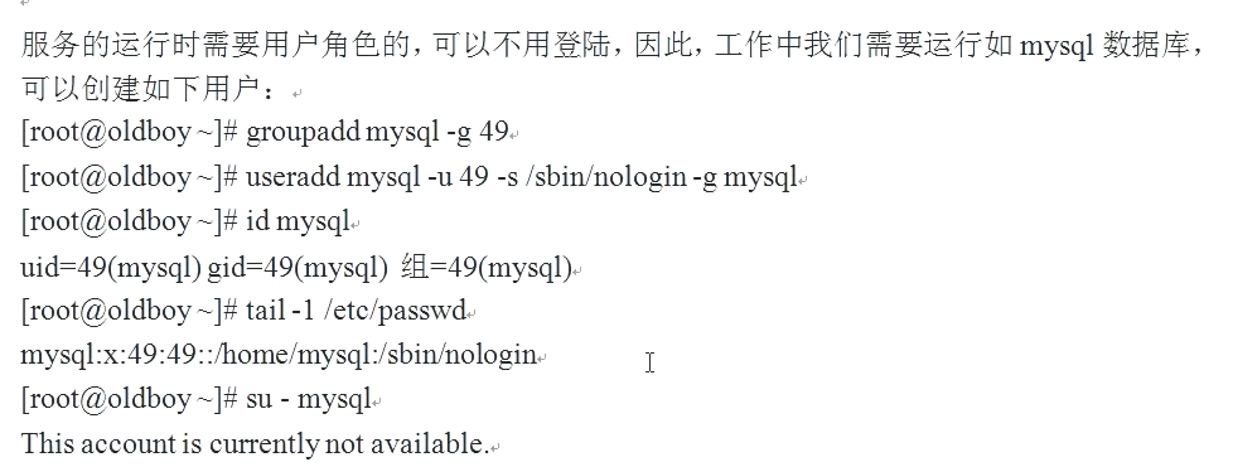
ntp:x:38:38::/etc/ntp:/sbin/nologin

tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin

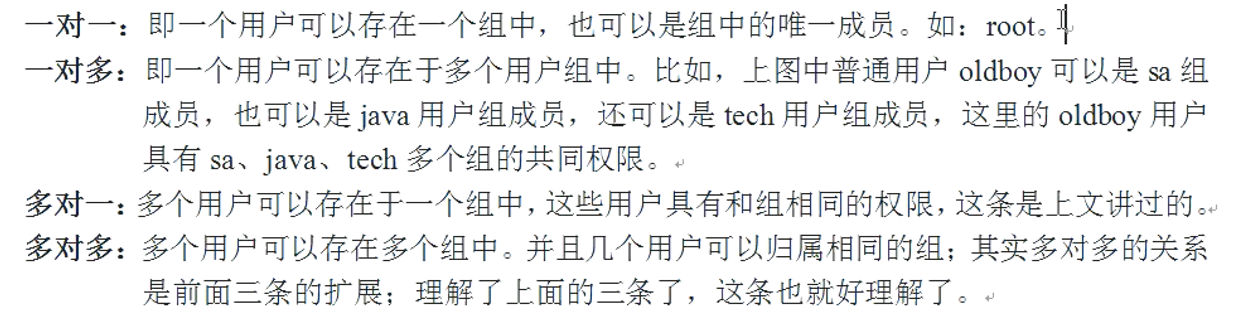
geoclue:x:995:993:User for geoclue:/var/lib/geoclue:/sbin/nologin

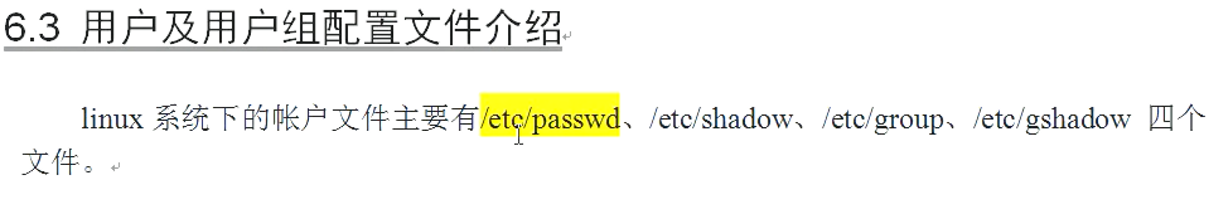
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin





用户可以用密码登陆也可以用秘钥验证登陆





加锁

[root@bogon /]# chattr +i /etc/passwd /etc/shadow /etc/group /etc/gshadow /etc/inittab

[root@bogon /]# lsattr /etc/passwd /etc/shadow /etc/group /etc/gshadow /etc/inittab

----i----------- /etc/passwd

----i----------- /etc/shadow

----i----------- /etc/group

----i----------- /etc/gshadow

----i----------- /etc/inittab

解锁

[root@bogon /]# chattr -i /etc/passwd /etc/shadow /etc/group /etc/gshadow /etc/inittab

[root@bogon /]# lsattr /etc/passwd /etc/shadow /etc/group /etc/gshadow /etc/inittab

---------------- /etc/passwd

---------------- /etc/shadow

---------------- /etc/group

---------------- /etc/gshadow

---------------- /etc/inittab

初始所有人都可以查看

[root@bogon /]# ls -l /etc/passwd

-rw-r--r-- 1 root root 1369 Oct 18 07:35 /etc/passwd

密码文件 只有root 能读

[root@bogon /]# ls -l /etc/shadow

---------- 1 root root 884 Oct 18 07:35 /etc/shadow

密码放在这里

[root@bogon /]# cat /etc/shadow

root:$6$warWy41MJo0qGxt5$J9lZdGShItUMhL0lZS5cicq/Q3oZpyVxFE5U5rQfOXDT3IdwkKjc5wnGZNduYKtv.tPMgLK7WFTja38Jdo4Vv.::0:99999:7:::

bin:\*:17110:0:99999:7:::

daemon:\*:17110:0:99999:7:::

adm:\*:17110:0:99999:7:::

lp:\*:17110:0:99999:7:::

sync:\*:17110:0:99999:7:::

shutdown:\*:17110:0:99999:7:::

halt:\*:17110:0:99999:7:::

mail:\*:17110:0:99999:7:::

operator:\*:17110:0:99999:7:::

games:\*:17110:0:99999:7:::

ftp:\*:17110:0:99999:7:::

nobody:\*:17110:0:99999:7:::

systemd-network:!!:17424::::::

dbus:!!:17424::::::

polkitd:!!:17424::::::

postfix:!!:17424::::::

sshd:!!:17424::::::

chrony:!!:17424::::::

mysql:!!:17425::::::

dockerroot:!!:17427::::::

dhcpd:!!:17427::::::

admin:$6$VLhJD.nV$UlUG/y78eCwLPYvPla43Qz55.KZvQ5tfOsiOk.oCSw3zmpCDyaLPvm261HKZ4fogiSv/EU6DgJQyumhq66AA70:17453:0:99999:7:::

saslauth:!!:17453::::::

ntp:!!:17453::::::

tss:!!:17453::::::

geoclue:!!:17453::::::

apache:!!:17457::::::

时间

[root@bogon data]# stat oldboy.txt

File: 鈕ldboy.txt?

Size: 66 Blocks: 8 IO Block: 4096 regular file

Device: fd00h/64768d Inode: 25876 Links: 1

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2017-10-16 06:43:56.993818298 -0400 #最近修改时间

Modify: 2017-10-14 07:16:25.485826228 -0400 #修改时间

Change: 2017-10-14 07:16:25.485826228 -0400#访问时间

[root@bogon data]# ls -lu oldboy.txt 访问后时间发生变化

-rw-r--r-- 1 root root 5 Oct 16 06:43 oldboy.txt

[root@bogon data]# stat oldboy.txt

File: 鈕ldboy.txt?

Size: 5 Blocks: 8 IO Block: 4096 regular file

Device: fd00h/64768d Inode: 25876 Links: 1

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2017-10-16 06:43:56.993818298 -0400

Modify: 2017-10-18 09:05:14.387269083 -0400

Change: 2017-10-18 09:05:14.387269083 -0400

Birth: -

查看完整时间

[root@bogon data]# ls -l --time-style=long-iso oldboy.txt

-rw-r--r-- 1 root root 5 2017-10-18 09:05 oldboy.txt

[root@bogon data]# ls -li --time-style=long-iso oldboy.txt

25876 -rw-r--r-- 1 root root 5 2017-10-18 09:05 oldboy.txt

[root@bogon data]# ls -li /etc # 这个点与 Selinux 有关

total 1268

16912228 -rw-r--r--. 1 root root 16 Sep 15 17:27 adjtime

16778328 -rw-r--r--. 1 root root 1518 Jun 7 2013 aliases

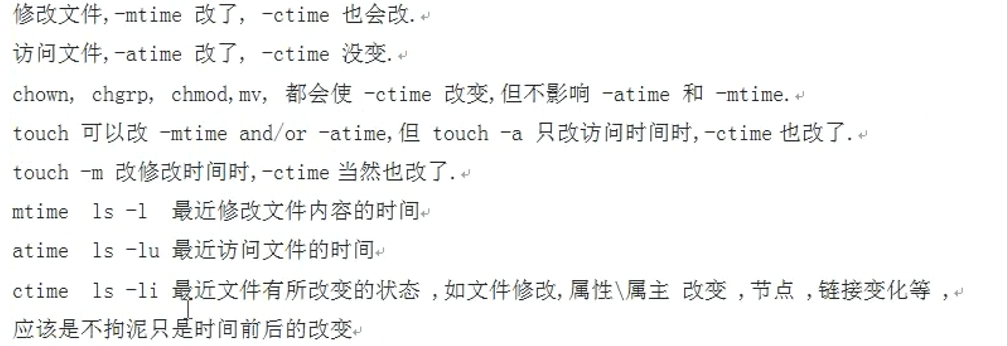
17490056 -rw-r--r--. 1 root root 12288 Sep 15 17:31 aliases.db

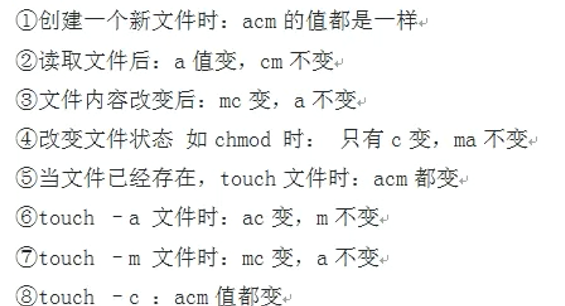
33683294 drwxr-xr-x. 2 root root 4096 Oct 14 09:24 alternatives

16833341 -rw-------. 1 root root 541 Aug 3 11:33 anacrontab

16854607 -rw-r--r--. 1 root root 55 Mar 1 2017 asound.conf

459049 drwxr-x---. 3 root root 43 Sep 15 17:21 audisp





**awk ,sed ,grep(egrep)三剑客的正则表达式**

**正则表达式： 正则表达是就是一些特殊字符，赋予了特殊的含义**

**1. ^word 搜索以word开头的**

**2. word$ 搜索以word结尾的**

[root@bogon /]# grep -i "^a" log.txt # -i 不区分大小写

[root@bogon /]# grep -i "a$" log.txt # -i 不区分大小写

[root@bogon /]# grep -n . log.txt # -n 是显示行号

[root@bogon /]# grep "\." log.txt

[root@bogon /]# grep "abc\*abc" log.txt

[root@bogon /]# grep ".\*" log.txt #匹配所有字符 全部查出来

[root@bogon /]# grep "b[a]c" log.txt #匹配有 bac 的

[root@bogon /]# grep "[^a]" log.txt # ^表示非的意思