

```
#####
###在线班-随堂记录-第2周#####
#####
讲师：老男孩教育-李老师
QQ:918391635
微信：lidao996
site: http://lidao.blog.51cto.com
```

#####linux目录结构特点
一切从根开始
磁盘/设备/分区没有挂载 无法使用

举例-linux下面使用光盘
###1.把光盘放入到光驱中

```
###2.linux中使用光盘 /dev/cdrom
[root@oldboyedu-01 ~]# ll /dev/cdrom
lrwxrwxrwx. 1 root root 3 Sep 25 10:01 /dev/cdrom -> sr0
[root@oldboyedu-01 ~]# cd /dev/cdrom
-bash: cd: /dev/cdrom: Not a directory
[root@oldboyedu-01 ~]# cat /dev/cdrom
```

```
###3.把光盘挂载到系统里面
[root@oldboyedu-01 ~]# mount /dev/cdrom /mnt/
mount: block device /dev/sr0 is write-protected, mounting read-only
[root@oldboyedu-01 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3        8.8G  1.4G   7.0G  17% /
tmpfs            1.9G   0    1.9G   0% /dev/shm
/dev/sda1        190M   40M   141M  22% /boot
/dev/sr0         3.7G  3.7G    0 100% /mnt
```

小结：
1.一切从根开始，一切皆文件
2.linux设备（光盘/磁盘分区）不挂载无法使用
3.挂载相当于给磁盘分区/设备 开了一个入口 通过入口进入到光盘/磁盘分区中
4.入口-挂载点-目录

遇到的错误：
[root@oldboyedu-01 ~]# yum install tree
Loaded plugins: fastestmirror, security
Setting up Install Process
Loading mirror speeds from cached hostfile
Could not retrieve mirrorlist
http://mirrorlist.centos.org/?release=6&arch=x86_64&repo=os&infra=stock error was
14: PYCURL ERROR 6 - "Couldn't resolve host 'mirrorlist.centos.org'"
Error: Cannot find a valid baseurl for repo: base

```
###文件目录详解--/etc/sysconfig/network-scripts/ifcfg-eth0
#网卡的配置文件
DEVICE=eth0                ###网卡的名字
HWADDR=00:0c:29:1e:67:61   #@hardware address硬件地址 MAC地址
TYPE=Ethernet              ###互联网
UUID=3405d22b-1aab-45a5-b57f-c0d005e4f269  ###UUID 系统中唯一的标识
ONBOOT=yes                 #@booton 在重启的时候是否开启网卡（自动运行）
NM_CONTROLLED=yes          ###是否能被network软件进行管理
BOOTPROTO=none             #@网卡获取ip地址的方式
#none/static ip地址是固定的
#dhcp                      自动获取ip地址
IPADDR=10.0.0.200           #@ ip address ip地址
NETMASK=255.255.255.0       ### 子网掩码
GATEWAY=10.0.0.2            #@ 网关 默认的出口
USERCTL=no                  ###是否准许普通用户管理网卡 开 关 重启
```

```
PEERDNS=yes                                ###
DNS1=223.5.5.5                             @@@
DNS2=223.6.6.6                             @@@
```

```
#上网使用 www.baidu.com jd.com taobao.com 域名
##把使用的域名转换为ip地址 ----DNS做 域名解析服务/系统
```

```
###阿里云的DNS
#223.5.5.5
#223.6.6.6
```

```
###114
#114.114.114.114
#114.114.115.115
```

```
###谷歌
#8.8.8.8
```

```
#####常见的DNS有关的错误 error ERROR
```

```
[root@oldboyedu-01 ~]# yum install tree
Loaded plugins: fastestmirror, security
Setting up Install Process
Loading mirror speeds from cached hostfile
Could not retrieve mirrorlist
http://mirrorlist.centos.org/?release=6&arch=x86_64&repo=os&infra=stock error was
14: PYCURL ERROR 6 - "Couldn't resolve host 'mirrorlist.centos.org'"
Error: Cannot find a valid baseurl for repo: base
```

```
ERROR 6 - "Couldn't resolve host 'mirrorlist.centos.org'"
错误      无法      解析      域名 mirrorlist.centos.org
          无法解析域名
```

```
ping: unknown host www.baidu.com
      未知的  域名
```

```
##linux无法上网，屌丝逃离洗浴中心之路(linux进行)
1.ping www.baidu.com      #是否能上网
```

```
2.ping 223.5.5.5          #DNS是否有问题
```

```
[root@oldboyedu-01 ~]# ping www.baidu.com
ping: unknown host www.baidu.com
[root@oldboyedu-01 ~]# ping 223.5.5.5
PING 223.5.5.5 (223.5.5.5) 56(84) bytes of data.
64 bytes from 223.5.5.5: icmp_seq=1 ttl=128 time=50.7 ms
64 bytes from 223.5.5.5: icmp_seq=2 ttl=128 time=50.4 ms
64 bytes from 223.5.5.5: icmp_seq=3 ttl=128 time=49.9 ms
64 bytes from 223.5.5.5: icmp_seq=4 ttl=128 time=49.8 ms
64 bytes from 223.5.5.5: icmp_seq=5 ttl=128 time=50.3 ms
^C
--- 223.5.5.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4810ms
rtt min/avg/max/mdev = 49.840/50.257/50.744/0.416 ms
[root@oldboyedu-01 ~]# #1.有问题
[root@oldboyedu-01 ~]# #2.ping通畅
[root@oldboyedu-01 ~]# #DNS有问题
```

```
##如何修改DNS
###1.在网卡配置文件中添加上
DNS1=223.5.5.5
DNS2=223.6.6.6
```

```
[root@oldboyedu-01 ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
HWADDR=00:0c:29:1e:67:61
TYPE=Ethernet
```

```
UUID=3405d22b-1aab-45a5-b57f-c0d005e4f269
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=none
IPADDR=10.0.0.200
NETMASK=255.255.255.0
GATEWAY=10.0.0.2
USERCTL=no
PEERDNS=yes
IPV6INIT=no
DNS1=223.5.5.5
DNS2=223.6.6.6
```

###2.网卡的配置生效-重启网卡

```
#####重启某一个网卡
ifdown eth0 && ifup eth0
```

```
#####重启所有网卡
/etc/init.d/network restart
```

网卡配置小结:

- 1.屌丝逃离洗浴中心之路
- 2.如何修改DNS
- 3.IPADDR,GATEWAY,ONBOOT,BOOTPROTO,DNS1,DNS2

```
#####文件目录详解--/etc/resolv.conf
配置DNS。
```

```
##网卡配置文件里面的DNS优先于 /etc/resolv.conf
##配置DNS都在网卡里面配置
```

```
#####文件目录详解--/etc/hosts ip地址与域名解析关系 对主机名进行解析
##网站更新
##进行测试-www.jd.com www.baidu.com www.oldboyedu.com
```

```
##www.baidu.com ---->119.75.216.20
###在测试的时候 下单---www.jd.com www.baidu.com 真的下单
####想办法把 www.jd.com www.baidu.com 解析到 10.0.0.200 (测试机器)
```

```
10.0.0.200 www.baidu.com www.jd.com
```

```
##进行测试
```

```
/etc/hosts
```

```
C:\Windows\System32\drivers\etc\hosts
```

```
#####文件目录详解--/etc/sysconfig/network 修改主机名的文件
#####第一个里程碑-查看当前的主机名
[root@oldboyedu-01 ~]# hostname
oldboyedu-01
```

```
#####第二个里程碑-临时修改主机名-重启服务器之后失效
[root@oldboyedu-01 ~]# hostname oldboyedu01-nb
[root@oldboyedu-01 ~]# hostname
oldboyedu01-nb
```

```
#####第三个里程碑-永久修改主机名-重启服务器之后生效
[root@oldboyedu01-nb ~]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=oldboyedu01-nb
```

```
####文件目录详解--/etc/fstab 开机的时候自动挂载的文件 开机的时候给每个磁盘分区开一个入口
#第一列 --表示设备/分区
```

#第二列 --表示入口/目录/挂载点

####文件目录详解--/etc/rc.local 开机的时候自动运行的文件
需要开机自启动的软件/命令/服务放在这个文件中

Startup

如何让一个软件/脚本/服务开机自动运行？

- 1./etc/rc.local #服务器运行软件的清单
- 2.通过chkconfig管理 开机自启动

####文件目录详解--/etc/inittab 开机的时候运行级别的配置文件
#init tab(table)

#####1.linux运行级别讲解

- 0 关机状态，不要把运行级别设置为0
- 1 单用户模式 root用户密码忘记了
- 2 多用户模式，但是没有NFS
- 3 完全的多用户模式，命令模式，文本模式
- 4 未使用
- 5 x11,桌面模式，图形化界面模式
- 6 重启状态，不要把运行级别设置为6

#####2.如何查看当前运行级别

[root@oldboyedu01-nb ~]# runlevel

N 3

[root@oldboyedu01-nb ~]# #N 3

[root@oldboyedu01-nb ~]# #上一次使用的运行级别 当前系统的运行级别

#####3.如何切换运行级别-临时

[root@oldboyedu01-nb ~]# #上一次使用的运行级别

当前系统的运行级别

[root@oldboyedu01-nb ~]# init 5

[root@oldboyedu01-nb ~]# runlevel

3 5

#####4.如何永久修改运行级别-重启生效

[root@oldboyedu01-nb ~]# tail -1 /etc/inittab

id:3:initdefault:

小结：

- 1.0-6含义
- 2.如何查看
- 3.如何修改 临时 永久

####文件目录详解--/etc/init.d 服务管理的命令

####文件目录详解--/etc/profile 系统环境变量 别名

#PS1 PATH

#LANG==>/etc/sysconfig/i18n

####文件目录详解--/etc/bashrc 别名

#####国法（全国各地生效）

/etc/profile 系统环境变量 别名 √√√√

/etc/bashrc 别名

#####家规（地方政策 当前用户生效）

~/.bashrc

~/.bash_profile

~表示当前用户的家目录

```
root ~ ===== /root
oldboy ~ ===== /home/oldboy
```

#####/usr/local 编译安装的软件默认的位置

C:\Program Files windows安装软件默认的位置

Linux下面安全软件的方法

- 1.yum (自动解决安装依赖的软件) 点外卖-你缺少什么外面给你解决
- yum install tree
- 2.rpm (缺啥少啥自己解决) 速冻水饺-半成品-缺少的东西自己解决
- 3.编译 自己做饭
买菜切菜-----炒菜-----上菜
./configure-----make-----make install

```
[root@oldboyedu01-nb ~]# ##查看软件是否安装成功
[root@oldboyedu01-nb ~]# rpm -qa |grep tree
tree-1.5.3-3.el6.x86_64
```

#####/usr/share

#####/var/log/messages 系统默认日志信息

#####/var/log/secure 系统用户的登录信息（谁 什么时候 从哪里登录的 是否成功）

#####文件目录详解--/proc/meminfo 内存信息

```
[root@oldboyedu01-nb ~]# free
              total        used         free       shared    buffers     cached
Mem:          3908500      386300      3522200           236       39192      165944
-/+ buffers/cache:        181164      3727336
Swap:         786428           0       786428

[root@oldboyedu01-nb ~]# free -h
              total        used         free       shared    buffers     cached
Mem:           3.7G         377M         3.4G         236K         38M         162M
-/+ buffers/cache:        176M         3.6G
Swap:          767M           0B         767M
```

#####文件目录详解--/proc/cpuinfo 查看cpu信息lscpu

#####文件目录详解--/proc/loadavg 系统的负载

负载:系统的繁忙程度

最近1分钟 最近5分钟 最近15分钟 平均负载

```
[root@oldboyedu01-nb ~]# w
22:20:05 up 12:18,  4 users,  load average: 0.00, 0.00, 0.00
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU WHAT
root      pts/0    10.0.0.1      19:16    2:08m  0.18s  0.18s -bash
root      pts/1    10.0.0.1      11:03    3:02m  0.06s  0.06s -bash
root      pts/2    10.0.0.1      20:32    1:00s  0.18s  0.07s w
root      pts/4    10.0.0.1      20:18    2:01m  0.03s  0.03s -bash
```

#系统的负载接近系统cpu的核心数量

#####文件目录详解--/proc/mounts 显示系统挂载信息

```
df -h
findmnt====cat /proc/mounts
```

####描述Linux系统的启动过程？

- 1.开机自检 BIOS
- 2.MBR引导
- 3.GRUB菜单
- 4.加载内核
- 5.运行init进程
- 6.从/etc/inittab读取运行级别
- 7.根据/etc/rc.sysinit 初始化系统（设置主机名 设置ip）
- 8.根据运行级别启动对应的软件（开机自启动软件）
- 9.运行mingetty显示登录界面

####什么是PATH环境变量，它有什么作用？

x+y=10

x=9

查看

```
echo $LANG $PS1
```

```
[root@oldboyedu43-lnb ~]# echo $PATH
```

```
/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
```

修改环境变量

```
export LANG=en_US.UTF-8
```

PATH作用：

- 1.环境变量
- 2.存放的是命令的位置/目录

执行一个命令的过程？

- 1.ls或 mkdir
- 2.系统会去 PATH里面的位置查找命令 是否存在
- 3.存在就运行
- 4.不存在就报错：command not found 或 no such file or dir

小结：

- 1.PATH作用
- 2.Linux下面执行命令简单过程

###第1题

如何过滤出已知当前目录下oldboy中的所有一级目录(提示:不包含oldboy目录下面目录的子目录及隐藏目录，即只能是第一级目录)？

```
mkdir /oldboy -p
cd /oldboy
mkdir ext/oldboy test xiaodong xiaofan xingfujie -p
touch jeacen oldboy wodi.gz yingsui.gz
tree
```

```
tree
find
ls+grep
```

方法1-tree

```
yum install tree -y
```

```
[root@oldboyedu43-lnb oldboy]# tree -Ld 1
```

```
.
├── ext
├── test
├── xiaodong
├── xiaofan
└── xingfujie
```

5 directories

方法2-find

```
[root@oldboyedu43-lnb oldboy]# find -maxdepth 1 -type d
.
./xingfujie
./test
./xiaodong
./xiaofan
./ext

[root@oldboyedu43-lnb oldboy]# find -maxdepth 1 -type d -name "."
.
[root@oldboyedu43-lnb oldboy]# find -maxdepth 1 -type d ! -name "."
./xingfujie
./test
./xiaodong
./xiaofan
./ext
```

方法3-grep

```
[root@oldboyedu43-lnb oldboy]# ls -l |grep "以d开头的行"
[root@oldboyedu43-lnb oldboy]# ls -l |grep "^d"
drwxr-xr-x  3 root root 4096 Dec 15 00:26 ext
drwxr-xr-x. 2 root root 4096 Dec 11 21:22 test
drwxr-xr-x  2 root root 4096 Dec 15 00:26 xiaodong
drwxr-xr-x  2 root root 4096 Dec 15 00:26 xiaofan
drwxr-xr-x  2 root root 4096 Dec 15 00:26 xingfujie
```

^ 高级货色(三剑客使用) 正则表达式 以.....开头的行

方法4-awk

```
[root@oldboyedu43-lnb oldboy]# ls -l |awk '第2列大于1'
awk: 第2列大于1
awk: ^ invalid char '莫in expression
[root@oldboyedu43-lnb oldboy]# ls -l |awk '$2>1'
total 32
drwxr-xr-x  3 root root 4096 Dec 15 00:26 ext
drwxr-xr-x. 2 root root 4096 Dec 11 21:22 test
drwxr-xr-x  2 root root 4096 Dec 15 00:26 xiaodong
drwxr-xr-x  2 root root 4096 Dec 15 00:26 xiaofan
drwxr-xr-x  2 root root 4096 Dec 15 00:26 xingfujie
```

方法5-ls

```
[root@oldboyedu43-lnb oldboy]# ls -F|grep "/"
ext/
test/
xiaodong/
xiaofan/
xingfujie/
[root@oldboyedu43-lnb oldboy]# #-F 给不通类型的文件 加上不通的标记/尾巴
```

方法6-ls

```
ls -ld */
```

####第2题 假如当前目录是

#如何快速的回到 上一次所在的位置

```
cd -
```

An argument of - is equivalent to \$OLDPWD.

```
cd - #cd $OLDPWD
```

```
cd - #如何快速的回到 上一次所在的位置
```

```
cd . #当前目录
```

复制/移动

```
cd .. #进入当前目录的上级目录
cd ~  #进入当前目录的家目录 回老家
cd    #进入当前目录的家目录 回老家
```

```
#进入到/etc/sysconfig/network-scripts/目录，并查看你所在位置
#进入到上一级目录，并查看你所在位置
#进入到/root目录下面，并显示你所在位置
#快速返回到/你上一次的位置
#回到老家
```

```
[root@oldboyedu43-lnb network-scripts]# cd /etc/sysconfig/network-scripts/
[root@oldboyedu43-lnb network-scripts]# pwd
/etc/sysconfig/network-scripts
[root@oldboyedu43-lnb network-scripts]# cd ..
[root@oldboyedu43-lnb sysconfig]# pwd
/etc/sysconfig
[root@oldboyedu43-lnb sysconfig]# cd /root/
[root@oldboyedu43-lnb ~]# pwd
/root
[root@oldboyedu43-lnb ~]# cd -
/etc/sysconfig
[root@oldboyedu43-lnb sysconfig]# cd
[root@oldboyedu43-lnb ~]# pwd
/root
```

####第3题 一个目录中有很多文件（ls查看时好多屏），想最快速度查看到最近更新的文件。如何看？
#ls显示的时候按照时间顺序 最近的在最下面 按时间顺序逆序/倒叙

```
ls -lrt
```

```
-r  逆序显示内容
-t  按照文件的修改时间
```

```
[root@oldboyedu43-lnb etc]# ls -l --time-style=long-iso
total 1636
drwxr-xr-x.  3 root root   4096 2017-12-08 12:38 abrt
drwxr-xr-x.  4 root root   4096 2017-12-08 12:41 acpi
-rw-r--r--.  1 root root    46 2017-12-12 07:18 adjtime
-rw-r--r--.  1 root root  1512 2010-01-12 21:28 aliases
-rw-r--r--.  1 root root 12288 2017-12-08 12:49 aliases.db
```

####第4题

已知apache/nginx服务的访问日志按天记录在服务器本地目录/app/logs下，由于磁盘空间紧张，现在要求只能保留最近7天访问日志！请问如何解决？
请给出解决办法或配置或处理命令。（提示：可以从apache服务配置上着手，也可以从生成出来的日志上着手。）

```
mkdir -p /app/logs
cd /app/logs
for time in {01..20};do date -s "201705$time"; touch access_www_$(date +%F).log ;done
date -s "20170520"
```

```
[root@oldboyedu43-lnb logs]# find /app/logs/ -type f -mtime +7|xargs ls -l
-rw-r--r-- 1 root root 0 May 1 00:00 /app/logs/access_www_2017-05-01.log
-rw-r--r-- 1 root root 0 May 2 00:00 /app/logs/access_www_2017-05-02.log
-rw-r--r-- 1 root root 0 May 3 00:00 /app/logs/access_www_2017-05-03.log
-rw-r--r-- 1 root root 0 May 4 00:00 /app/logs/access_www_2017-05-04.log
-rw-r--r-- 1 root root 0 May 5 00:00 /app/logs/access_www_2017-05-05.log
-rw-r--r-- 1 root root 0 May 6 00:00 /app/logs/access_www_2017-05-06.log
-rw-r--r-- 1 root root 0 May 7 00:00 /app/logs/access_www_2017-05-07.log
-rw-r--r-- 1 root root 0 May 8 00:00 /app/logs/access_www_2017-05-08.log
-rw-r--r-- 1 root root 0 May 9 00:00 /app/logs/access_www_2017-05-09.log
-rw-r--r-- 1 root root 0 May 10 00:00 /app/logs/access_www_2017-05-10.log
-rw-r--r-- 1 root root 0 May 11 00:00 /app/logs/access_www_2017-05-11.log
-rw-r--r-- 1 root root 0 May 12 00:00 /app/logs/access_www_2017-05-12.log
```


找出/app/logs下面以.log结尾的并且修改时间是7天之前的文件并删除 (ls -l)
find /app/logs/ -type f -name "*.log" -mtime +7

```
#find /app/logs/ -type f -name "*.log" -mtime +7|xargs ls -l
#ls -l $(find /app/logs/ -type f -name "*.log" -mtime +7)
#find /app/logs/ -type f -name "*.log" -mtime +7 -exec ls -l {} \;
```

通过系统软件对日志进行切割。

补充find命令相关题目：

查找/oldboy 下所有以log 结尾的大于1M 的文件复制到/tmp

```
[root@oldboyedu43-lnb logs]# cat /etc/services /etc/services > 1m.log
[root@oldboyedu43-lnb logs]# ls -lh 1m.log
-rw-r--r-- 1 root root 1.3M Dec 12 00:14 1m.log
```

```
find /oldboy -type f -name "*.log" -size +1M
```

```
-size +1M
-size +100k
```

```
find /oldboy -type f -name "*.log"
```

```
mkdir -p /tmp/a /tmp/b /tmp/c /tmp/d
```

方法1 find+\$()
#cp 次处是find命令的结果 /tmp/a/
cp \$(find /app/logs/ -type f -name "*.log") /tmp/a/

方法2 find + -exec
find /app/logs/ -type f -name "*.log" -exec cp {} /tmp/b/ \;

方法3 find + |xargs
find /app/logs/ -type f -name "*.log" |xargs cp -t /tmp/c

####第5题 调试系统服务时，希望能实时查看系统日志/var/log/messages的更新,如何做?
tail -f /var/log/secure

####第6题

```
[root@oldboyedu-39-nb logs]# echo {1..5}
1 2 3 4 5
[root@oldboyedu-39-nb logs]# echo {a..f}
a b c d e f
[root@oldboyedu-39-nb logs]# echo {1..5}
1 2 3 4 5
[root@oldboyedu-39-nb logs]# #stu1 stu2 stu3 ... stu5
[root@oldboyedu-39-nb logs]# echo stu{1..5}
stu1 stu2 stu3 stu4 stu5
[root@oldboyedu-39-nb logs]# echo stu{1..5}|xargs -n1
stu1
stu2
stu3
stu4
stu5
[root@oldboyedu-39-nb logs]# echo stu{1..5}|xargs -n1 >/oldboy/nginx.conf
[root@oldboyedu-39-nb logs]# cat /oldboy/nginx.conf
stu1
stu2
stu3
stu4
stu5
```

###方法1

```
[root@oldboyedu-39-nb oldboy]# cat -n nginx.conf
 1  stu1
 2  stu2
 3  stu3
 4  stu4
 5  stu5
```

###方法2 vi /vim

```
:set nu    #显示行号
:set nonu  #取消显示行号
```

###方法3 grep

```
[root@oldboyedu-39-nb oldboy]# grep -n "." nginx.conf
1:stu1
2:stu2
3:stu3
4:stu4
5:stu5
```

####. 正则表达式里面的 . 表示任意一个字符

###方法4 sed

```
[root@oldboyedu-39-nb oldboy]# sed '=' nginx.conf |xargs -n2
1 stu1
2 stu2
3 stu3
4 stu4
5 stu5
```

###方法5 awk

```
[root@oldboyedu-39-nb oldboy]# awk '显示行号' nginx.conf
awk: 显示行号
awk: ^ invalid char '?'in expression
[root@oldboyedu-39-nb oldboy]# awk '{print NR}' nginx.conf
1
2
3
4
5
[root@oldboyedu-39-nb oldboy]# awk '{print NR,$0}' nginx.conf
1 stu1
2 stu2
3 stu3
4 stu4
5 stu5
```

###方法6

```
[root@oldboyedu-39-nb oldboy]# nl nginx.conf
 1  stu1
 2  stu2
 3  stu3
 4  stu4
 5  stu5
```

###小结:

- 1.显示文件的行号 cat vi/vim
- 2.其他扩展 了解

####第7题

- 1.每个运行级别的含义
- 2.怎么配置
- 3.怎么查看当前系统的运行级别
- 4.临时切换运行级别

####第8题

```
# chkconfig iptables off
# chkconfig |grep ipt
iptables          0:off    1:off    2:off    3:off    4:off    5:off    6:off
# chkconfig --level 3 iptables on
# chkconfig |grep ipt
iptables          0:off    1:off    2:off    3:on     4:off    5:off    6:off
#
# chkconfig --level 3 iptables off
# chkconfig |grep ipt
iptables          0:off    1:off    2:off    3:off    4:off    5:off    6:off
```

如何彻底关闭 iptables ?

```
1./etc/init.d/iptables stop    临时 关闭正在运行的防火墙
2.chkconfig iptables off      永久 让iptables不会在开机的时候自启动
3.检查
chkconfig |grep ipt
/etc/init.d/iptables status
```

##第9题 linux中显示中文乱码

#第一步-排查

#第1个里程碑-看看linux系统的字符集

```
echo $LANG
```

#第2个里程碑-远程连接工具 xshell/SecureCRT/putty 字符集

#第二步-修改 修复 修改字符集

#第一个里程碑-修改xshell字符集（推荐）

#第二个里程碑-修改linux系统的字符集

###1.临时

```
export LANG=en_US.UTF-8
```

###2.永久

```
[root@oldboyedu-39-nb ~]# cat /etc/sysconfig/i18n
```

```
LANG="en_US.UTF-8"
```

```
SYSFONT="latarcyrheb-sun16"
```

###3.生效

```
source /etc/sysconfig/i18n
```

###11题

windows压缩

linux 打包(把文件放在筐里)及压缩(把这个框挤碎了)

####创建一个压缩包

```
#tar zcvf /tmp/etc.tar.gz /etc/
```

#z----压缩工具---gzip 最常用一种 压缩之后我们一般给 压缩包命名位 xxxx.tar.gz

#c----创建-----create

#v----显示压缩/解压过程

#f----file-----指定压缩包的名字

###查看压缩包里面的内容

```
tar ztf /tmp/etc.tar.gz
```

#t---list-----列表 显示

###解压----解压到当前目录

```
# cd /tmp/
```

```
# pwd
```

```
/tmp
```

```
# tar zxvf /tmp/etc.tar.gz
```

#x-----extract 解压

###创建压缩包

```
tar zcf /tmp/etc.tar.gz /etc/
```

###查看压缩包

```
tar tf /tmp/etc.tar.gz
```

#解压

```
tar xf /tmp/etc.tar.gz
```

###问题:"创建压缩包"的时候会提示

```
tar: Removing leading `/' from member names
```

tar: 把每个文件开头的/删除掉了。

```
/etc/hosts -----> etc/hosts
```

tar命令提示你: 我在创建压缩包的时候 把压缩包中的文件 绝对路径----->相对路径

```
tar: Removing leading `/' from hard link targets
```

###小结:核心 为了安全----tar命令 把你使用的绝对路径----变化为-----相对路径

```
[root@oldboyedu-39-nb tmp]# tar zcf /tmp/hosts.tar.gz /etc/hosts
```

```
tar: Removing leading `/' from member names
```

```
[root@oldboyedu-39-nb tmp]# tar tf /tmp/hosts.tar.gz
```

```
etc/hosts
```

###在我们打包的时候 使用绝对路径 tar会自动的变为 相对路径

绝对路径----->相对路径

####1.假设压缩包里面还是绝对路径

```
[root@oldboyedu-39-nb tmp]# tar tf /tmp/hosts.tar.gz
```

```
/etc/hosts
```

####2.解压

####3.解压后 会把里面的文件放在 /etc/hosts ----- 这会导致 原有的/etc/hosts被覆盖了。

####4.所以tar命令为了安全 你在打包的时候 会把压缩包中的绝对路径----->相对路径

```
tar: Removing leading `/' from member names
```

##如何不提示这个警告-----使用相对路径打包

etc/hosts为例

```
[root@oldboyedu-39-nb tmp]# cd /
```

```
[root@oldboyedu-39-nb /]# tar zcf /tmp/hosts.tar.gz etc/hosts
```

```
[root@oldboyedu-39-nb /]# tar tf /tmp/hosts.tar.gz
```

```
etc/hosts
```

b.请用tar打包/etc整个目录（打包及压缩，但需要排除/etc/services文件）。

```
[root@oldboyedu-39-nb /]# tar zcf /tmp/etc-pai.tar.gz /etc/ --exclude=services
```

###排除所有文件名叫services的文件

```
tar: Removing leading `/' from member names
```

```
tar: Removing leading `/' from hard link targets
```

```
[root@oldboyedu-39-nb /]# tar tf /tmp/etc-pai.tar.gz |grep services
```

```
etc/init/readahead-disable-services.conf
```

```
[root@oldboyedu-39-nb /]# tar tf /tmp/etc.tar.gz |grep services
```

```
etc/services
```

```
etc/init/readahead-disable-services.conf
```

##排除---精确版本 加上位置

```
[root@oldboyedu-39-nb /]# tar zcf /tmp/etc-pai.tar.gz /etc/ --exclude=etc/services
tar: Removing leading '/' from member names
tar: Removing leading '/' from hard link targets
[root@oldboyedu-39-nb /]# tar tf /tmp/etc-pai.tar.gz |grep services
etc/sysconfig/services
etc/init/readahead-disable-services.conf
```

#把你要排除的名单写到一个文件中 /tmp/paichu.txt (了解)

```
#tar zcf /tmp/etc-pai.tar.gz /etc/ --exclude-from=/tmp/paichu.txt
```

c.请把a点命令的压缩包,解压到/tmp指定目录下(最好只用tar命令实现)。

```
tar xf /tmp/etc-pai.tar.gz -C /opt
```

把/etc/hosts /etc/sysconfig/network /etc/sysconfig/il8n /etc/init.d/ 打包压缩
/tmp/conf.tar.gz
解压到/opt目录

```
[root@oldboyedu-39-nb /]# tar zcf /tmp/conf.tar.gz /etc/hosts /etc/sysconfig/network
/etc/sysconfig/il8n /etc/init.d/
tar: Removing leading '/' from member names
[root@oldboyedu-39-nb /]# tar tf /tmp/conf.tar.gz
etc/hosts
etc/sysconfig/network
etc/sysconfig/il8n
etc/init.d
[root@oldboyedu-39-nb /]# tar xf /tmp/conf.tar.gz -C /opt/
[root@oldboyedu-39-nb /]# ls /opt/
etc rh
```

####tar

#1.创建压缩包

```
tar zcf /tmp/oldboy.tar.gz /oldboy
```

#2.查看压缩包中的内容

```
tar tf /tmp/oldboy.tar.gz
```

#3.解压---解压到当前目录

```
tar xf /tmp/oldboy.tar.gz
tar xf /tmp/oldboy.tar.gz -C /opt
```

#4.创建压缩包的时候 排除

```
tar zcf /tmp/etc-pai.tar.gz /etc/ --exclude=etc/services
```

第11题 已知如下命令及结果:

```
mkdir -p /oldboy
echo "I am oldboy,myqq is 31333741">/oldboy/oldboy.txt
```

- 现在需要从文件中过滤出"oldboy"和"31333741"字符串,请给出命令.
- 如果需要从文件中过滤出"oldboy,31333741"字符串,请再给出命令.

a. 现在需要从文件中过滤出"oldboy"和"31333741"字符串,请给出命令.

方法1-sed-sed

```
[root@oldboyedu01-nb oldboy]# sed 's#I am ##g' oldboy.txt |sed 's#,myqq is##g'
oldboy 31333741
```

方法2-sed/tr+awk

```
[root@oldboyedu01-nb oldboy]# sed 's#,# #g' oldboy.txt |awk '{print $3,$6}'
oldboy 31333741
```

```
[root@oldboyedu01-nb oldboy]# tr " " " " <oldboy.txt |awk '{print $3,$6}'
oldboy 31333741
```

方法3-awk指定多个分隔符号

```
[root@oldboyedu01-nb oldboy]# cat oldboy.txt
I am oldboy,myqq is 31333741
[root@oldboyedu01-nb oldboy]# awk -F " " '{print $1}' oldboy.txt
I am oldboy
```

```
[root@oldboyedu01-nb oldboy]# awk -F "[, ]" '{print $3,$6}' oldboy.txt
oldboy 31333741
```

-F "[,]" 表示以逗号或者空格作为菜刀 分隔符

b. 如果需要从文件中过滤出“oldboy,31333741”字符串, 请再给出命令.

```
[root@oldboyedu01-nb oldboy]# awk -F "[, ]" '{print $3,$6}' oldboy.txt
oldboy 31333741
[root@oldboyedu01-nb oldboy]# awk -F "[, ]" '{print $3,""$6}' oldboy.txt
oldboy,31333741
[root@oldboyedu01-nb oldboy]# awk -F "[, ]" '{print $3"$1"$6}' oldboy.txt
oldboy$131333741
[root@oldboyedu01-nb oldboy]# awk -F "[, ]" '{print $3" $1 "$6}' oldboy.txt
oldboy $1 31333741
```

小结:

1.tr命令

2.awk指定分隔符 指定多个分隔符

第12题 如何查看/etc/services文件的有多少行?

```
[root@oldboyedu01-nb oldboy]# wc -l /etc/services
10774 /etc/services
```

屌丝去洗浴中心之路

3.

1)查看22端口是否开启 telnet

2)sshd远程连接进程是否在运行*****

ps -ef

```
[root@oldboyedu01-nb oldboy]# ps -ef |grep "sshd"
root          1509      1  0 17:51 ?                00:00:00 /usr/sbin/sshd
root          1669    1509  0 17:51 ?                00:00:00 sshd: root@pts/0
root          1795    1509  0 18:17 ?                00:00:00 sshd: root@pts/1
root          1813    1671  0 18:17 pts/0          00:00:00 grep sshd
```

```
[root@oldboyedu01-nb oldboy]# ps -ef |grep "/sshd"
root          1509      1  0 17:51 ?                00:00:00 /usr/sbin/sshd
root          1817    1671  0 18:19 pts/0          00:00:00 grep /sshd
[root@oldboyedu01-nb oldboy]# ps -ef |grep "/sshd"|wc -l
2
```

第13题 过滤出/etc/services 文件包含3306或1521两数字所在的行的内容。

```
[root@oldboyedu01-nb oldboy]# egrep "3306|1521" /etc/services
mysql          3306/tcp                # MySQL
mysql          3306/udp                # MySQL
ncube-lm       1521/tcp                # nCube License Manager
ncube-lm       1521/udp                # nCube License Manager
[root@oldboyedu01-nb oldboy]# #egrep === grep -E 支持高级正则 (公鸡里的战斗机)
```

第14题命令行及shell中加单引号和加双引号的区别小结

单引号 所见即所得 吃啥吐啥

```
[root@oldboyedu01-nb oldboy]# echo 'hello lls $LANG $(hostname) `pwd`'
hello lls $LANG $(hostname) `pwd`
```

双引号 里面的特殊符号会被解析

```
[root@oldboyedu01-nb oldboy]# echo "hello lls $LANG $(hostname) `pwd`"
hello lls en_US.UTF-8 oldboyedu01-nb /oldboy
```

```
[root@oldboyedu01-nb ~]# ls -lhi /etc/hosts
390174 -rw-r--r--. 2 root root 200 Sep 25 20:19 /etc/hosts
[root@oldboyedu01-nb ~]# #文件属性====ls -lhi 每一列的含义
```

						用户属于的家庭	文件修改时间	文件名
390174	-	rw-r--r--.	2		root root	200		
Sep 25 20:19	/etc/hosts							
inode号码	文件的类型	文件的权限.(与selinux有关)	文件硬链接数量	主人	家人	文件大小		

文件名不是文件的属性。

inode和block简介

获得一个硬盘到可以向硬盘中存放数据

硬盘（房子），分区(划分隔断),格式化创建文件系统(装修-不同风格),挂载(安装门窗)

inode

1) 含义

index node索引节点

用来存放文件属性的空间。通过inode号码找到这个空间的。

inode号码----家庭住址

inode空间----家庭

文件名不存放在inode中。

2) 怎么来的

格式化创建文件系统

3) 特点

1#inode存放文件属性

2#存放block的位置(指向block的指针)

3#创建一个文件要占用一个inode

4#inode节点号相同的文件，互为硬链接文件，可以认为是一个文件的不同入口。

硬链接数量====超市入口的数量

5#inode在某一个文件系统（分区），是唯一的。

4) 怎么用/看*****

ls -li (看某些文件的inode号码)

df -li (系统里面inode用了多少剩余多少)

block

1) 含义

数据块

block实际存放数据的位置

2) 怎么来的

格式化创建文件系统

3) 特点

- 1#block是实际存放数据的位置
- 2#block 4k(centos 6.x) 1k 8k
- 3#创建大文件会占用多个block, 如果文件很小1k, 4k剩余的空间会被浪费。
- 4#创建一个非空文件要占用一个inode和至少1个block
- 5#每读取一个block就会消耗一次磁盘I/O (input/output 磁盘读写)。

4) 怎么用/怎么看

```
df -h(磁盘block使用情况)
ls -lh
```

小结:

inode与block小结:

- 1.格式化创建文件系统
- 2.在同一个文件系统(分区), 两个文件inode相同, 互为硬链接
- 3.创建一个非空文件要占用一个inode和至少1个block
- 4.如何查看inode和block的使用情况 df -i df -h

查看oldboy.txt文件过程-了解。

no space left on device磁盘空间不足-因为block满了。

```
seq 1000000000 >/tmp/big
```

#第一个里程碑-看看到底哪里满了

```
[root@oldboyedu01-nb ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda3	8.8G	8.8G	0	100%	/
tmpfs	1.9G	0	1.9G	0%	/dev/shm
/dev/sda1	190M	40M	141M	22%	/boot

#第二个里程碑-是哪一个目录占得空间大, 最终定位到目录或文件

```
[root@oldboyedu01-nb ~]#
```

```
[root@oldboyedu01-nb ~]# du -sh /*|grep G
```

```
1.1G    /usr
```

```
7.4G    /var
```

```
[root@oldboyedu01-nb ~]# du -sh /var/*|grep G
```

```
7.3G    /var/log
```

```
[root@oldboyedu01-nb ~]# du -sh /var/log/*|grep G
```

```
7.3G    /var/log/messages
```

```
[root@oldboyedu01-nb ~]# ls -lh /var/log/messages
```

```
-rw----- 1 root root 7.3G Oct 16 19:20 /var/log/messages
```

#第三个里程碑-问清楚是否可以删除, 然后在操作。

【Y】企业案例: 如果向磁盘写入数据提示如下错误: No space left on device (磁盘空间不足), 通过df -h查看磁盘空间, 发现没满, 请问可能原因是什么? 企业场景什么情况下会导致这个问题发生? inode用光了。

模拟环境的命令。

```
mkdir -p /app/logs
```

```
dd if=/dev/zero of=/dev/sdc bs=8K count=10
```

```
ls -l /dev/sdc
```

```
mkfs.ext4 /dev/sdc
```

```
mount -o loop /dev/sdc /app/logs
```

```
[root@oldboyedu01-nb ~]# cd /app/logs/
```

```
[root@oldboyedu01-nb logs]# touch stu{1..6}.log
```

```
touch: cannot touch `stu6.log': No space left on device
```

```
[root@oldboyedu01-nb logs]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda3	8.8G	1.6G	6.9G	19%	/
tmpfs	1.9G	0	1.9G	0%	/dev/shm
/dev/sda1	190M	40M	141M	22%	/boot
/dev/sdc	73K	14K	55K	21%	/app/logs


```
[root@oldboyedu01-nb logs]# df -i
Filesystem      Inodes  IUsed  IFree  IUse% Mounted on
/dev/sda3       593344 58530  534814    10% /
tmpfs           488562    1  488561     1% /dev/shm
/dev/sda1       51200   39   51161     1% /boot
/dev/sdc         16     16     0    100% /app/logs
```

文件类型

```
f file      普通文件
d directory 目录
l softlink  软链接(快捷方式)
```

```
f file      普通文件
```

```
[root@oldboyedu01-nb ~]# ls -l /tmp/etc.tar.gz /bin/ls /etc/hosts
-rwxr-xr-x. 1 root root 117048 Mar 23 2017 /bin/ls
-rw-r--r--. 2 root root 200 Sep 25 20:19 /etc/hosts
-rw-r--r-- 1 root root 9744299 Oct 20 2017 /tmp/etc.tar.gz
```

##二进制文件 命令

```
[root@oldboyedu01-nb ~]# file /bin/ls
/bin/ls: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked (uses
shared libs), for GNU/Linux 2.6.18, stripped
```

##文本文件

```
[root@oldboyedu01-nb ~]# file /etc/hosts
/etc/hosts: ASCII text
```

##数据文件 data

```
[root@oldboyedu01-nb ~]# file /tmp/etc.tar.gz
/tmp/etc.tar.gz: gzip compressed data, from Unix, last modified: Fri Oct 20 01:36:16 2017
```

扩展名-文件后面的小尾巴

```
windows      windows通过扩展名区分不同类型的文件。
linux        扩展名是用来给我们看的。
```

```
.sh    脚本文件
.conf  配置文件
```

用户和用户组简介

整体对于一个文件/目录来说：用户可以分为3类

```
主人      家人      陌生人
所有者的名字 所有者属于的组的名字
```

```
[root@oldboyedu01-nb tmp]# ll -hi /etc/hosts
390174 -rw-r--r--. 2 root root 200 Sep 25 20:19 /etc/hosts
```

假设我是oldboy用户，我与这个文件有什么关系

oldboy

```
[root@oldboyedu01-nb ~]# id oldboy
uid=500(oldboy) gid=500(oldboy) groups=500(oldboy)
#id查询用户信息
#属于哪个用户组
```

linux中存在于系统中的用户如何分类

```
UID    user  id  用户的id号码====身份证号码    \\\\\\
GID    group id  用户组的id号码===户口本的号码
```

```
1.皇帝 root      UID
0
```

2. 傀儡 虚拟用户 1-499 系统中每个进程要有一个对应的用户，无法使用的。

3. 贫民百姓 普通用户 500+

如何保护皇帝

1. 禁止root远程登录
2. 修改ssh默认的端口号22
3. 定期分析/var/log/secure 记录着用户的登录信息

#与用户有关的文件

/etc/passwd 存放的是用户的信息

/etc/shadow 用户密码的信息

/etc/group 用户组的信息

/etc/passwd

root :x:0 :0 :root:/root :/bin/bash

oldboy:x:500:500: :/home/oldboy:/bin/bash

第一列:用户名字

第二列:存放密码的地方

第三列:UID

第四列:GID

第五列:用户的说明信息，默认为空

第六列:用户的家目录

~

root ~ /root

oldboy ~ /home/oldboy

第七列:用户使用的shell,用户使用的命令解释器

/bin/bash-----centos默认的命令解释器

/sbin/nologin----用户无法登陆系统，相当于是傀儡用户

linux基础权限介绍

r 4 read可以读取文件内容

w 2 write修改文件内容

x 1 execute可以运行文件(命令 脚本)

- 0 什么权限都没有

- rw- r-- r--. 2 root root 200 Sep 25 20:19 /etc/hosts

主人 家人 陌生人

420 400 400

6 4 4

[root@oldboyedu01-nb ~]# ls -l /bin/ls

-rwxr-xr-x. 1 root root 117048 Mar 23 2017 /bin/ls

421401401

7 5 5

软硬链接

软链接-windows快捷方式

符号链接

怎么来的

```
[root@oldboyedu01-nb oldboy]# ln -s oldboy.txt oldboy.txt-soft
[root@oldboyedu01-nb oldboy]# ls -l oldboy.txt*
-rw-r--r--. 1 root root 29 Oct 16 17:52 oldboy.txt
lrwxrwxrwx 1 root root 10 Oct 17 09:27 oldboy.txt-soft -> oldboy.txt
```

特点

1. 存放着源文件的位置
2. 源文件被删除
3. 节约时间

硬链接

相当于是超市的前后门，通过不同的大门进入超市

在同一个分区中，两个文件或多个文件的inode号码相同，这些文件互为硬链接

怎么来的

ln 创建

特点

1. 相当于是超市的前后门，硬链接数量就是超市一共有多少门
2. 删除硬链接或源文件，文件还是可以使用的
3. 把硬链接和源文件都删除，文件被删除====硬链接数量为0

```
[root@oldboyedu01-nb oldboy]# ls -l oldboy.txt*
-rw-r--r--. 1 root root 29 Oct 16 17:52 oldboy.txt
lrwxrwxrwx 1 root root 10 Oct 17 09:27 oldboy.txt-soft -> oldboy.txt
[root@oldboyedu01-nb oldboy]# ln oldboy.txt oldboy.txt-hard
[root@oldboyedu01-nb oldboy]# ls -lhi oldboy.txt*
400490 -rw-r--r--. 2 root root 29 Oct 16 17:52 oldboy.txt
400490 -rw-r--r--. 2 root root 29 Oct 16 17:52 oldboy.txt-hard
400479 lrwxrwxrwx 1 root root 10 Oct 17 09:27 oldboy.txt-soft -> oldboy.txt
```

描述linux下软链接和硬链接的区别（面试题）

解答：

- 1) 如何创建：
 - a) 默认不带参数情况下，ln命令创建的是硬链接，带-s参数的ln命令创建的是软链接。
- 2) 含义：
 - a) 硬链接文件与源文件的inode节点号相同，而软链接文件相当于windows下面的快捷方式（inode节点号与源文件不同）
- 3) 特点：
 - a) 不能对目录创建硬链接，但可以创建软链接，对目录的软链接会经常被用到
 - b) 软链接可以跨文件系统，硬链接不可以跨文件系统。
- 4) 怎么没了（源文件，软链接，硬链接与删除）
 - a) 除软链接文件，对源文件及硬链接文件无任何影响；
 - b) 除文件的硬链接文件，对源文件及软链接文件无任何影响；
 - c) 除链接文件的源文件，对硬链接文件无影响，会导致其软链接失效（红底白字闪烁状）；
 - d) 时删除源文件及其硬链接文件，整个文件才会被"真正"的删除。

文件删除原理

- 1) 删除这个文件所有相关的硬链接-rm---删除厕所所有的入口
- 2) 这个文件的进程调用数也为0 -----没有人在使用厕所

1. 硬链接数量为0
2. 进程调用数为0

某个文件没有被彻底删除，导致的磁盘空间满了。

模拟思路

向/var/log/secure 追加8亿行
seq 800000000

```
[root@oldboyedu01-nb oldboy]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3        8.8G  8.8G    0 100% /
tmpfs            1.9G   0  1.9G   0% /dev/shm
/dev/sda1        190M   40M  141M  22% /boot
/dev/sdc          73K   14K   55K  21% /app/logs
```

#正常的排错过程。

```
[root@oldboyedu01-nb oldboy]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3        8.8G  8.8G    0 100% /
tmpfs            1.9G   0  1.9G   0% /dev/shm
/dev/sda1        190M   40M  141M  22% /boot
/dev/sdc          73K   14K   55K  21% /app/logs
[root@oldboyedu01-nb oldboy]# du -sh /* |grep G
du: cannot access `/proc/2904/task/2904/fd/4': No such file or directory
du: cannot access `/proc/2904/task/2904/fdinfo/4': No such file or directory
du: cannot access `/proc/2904/fd/4': No such file or directory
du: cannot access `/proc/2904/fdinfo/4': No such file or directory
1.1G    /usr
7.4G    /var
[root@oldboyedu01-nb oldboy]# du -sh /var/* |grep G
7.3G    /var/log
[root@oldboyedu01-nb oldboy]# du -sh /var/log/* |grep G
7.3G    /var/log/secure
[root@oldboyedu01-nb oldboy]# \rm -f /var/log/secure
```

#磁盘空间满了

```
[root@oldboyedu01-nb oldboy]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3        8.8G  8.8G    0 100% /
tmpfs            1.9G   0  1.9G   0% /dev/shm
/dev/sda1        190M   40M  141M  22% /boot
/dev/sdc          73K   14K   55K  21% /app/logs
[root@oldboyedu01-nb oldboy]# du -sh /* |grep G
du: cannot access `/proc/2914/task/2914/fd/4': No such file or directory
du: cannot access `/proc/2914/task/2914/fdinfo/4': No such file or directory
du: cannot access `/proc/2914/fd/4': No such file or directory
du: cannot access `/proc/2914/fdinfo/4': No such file or directory
1.1G    /usr
```

系统中某个文件没有被彻底删除。
如何找出硬链接数量为0， 进程调用数不为0

```
[root@oldboyedu01-nb oldboy]# ls -lsof |grep delete
-rsyslogd 1250      root      2w      REG      8,3 7803924480      130420
/var/log/secure (deleted)
软件名称                                     文件大小                                     文件名
```

(deleted)---标记 这个文件的硬链接数量为0， 进程调用数不为0。

##重启对应的服务

```
[root@oldboyedu01-nb oldboy]# /etc/init.d/rsyslog restart
Shutting down system logger: [ OK ]
Starting system logger:      [ OK ]
[root@oldboyedu01-nb oldboy]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3        8.8G  1.6G  6.9G  19% /
tmpfs            1.9G   0  1.9G   0% /dev/shm
/dev/sda1        190M   40M  141M  22% /boot
/dev/sdc          73K   14K   55K  21% /app/logs
```

磁盘空间不足-no space left on device

- 1.block满了正常-500G 存放505G视频
- 2.inode满了, 大量的小文件
- 3.block 硬链接数量为0 进程调用数不为零 文件没有被彻底删除磁盘空间不足

linux修改、访问、改变时间简介

mtime	modified	time	文件的修改时间	文件内容的变化时间
ctime	change	time	文件属性的改变时间	硬链接数量 文件大小 文件权限
atime	access	time	文件的访问时间	cat

```
[root@oldboyedu01-nb ~]# ls -l oldboy.txt
-rw-r--r--. 1 root root 17 Sep 25 16:47 oldboy.txt
[root@oldboyedu01-nb ~]# stat oldboy.txt
  File: `oldboy.txt'
  Size: 17          Blocks: 8          IO Block: 4096   regular file
Device: 803h/2051d Inode: 262707       Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2017-09-25 16:47:54.628044632 +0800
Modify: 2017-09-25 16:47:53.059040702 +0800
Change: 2017-09-25 16:47:53.101042388 +0800
[root@oldboyedu01-nb ~]#
```

```
[root@oldboyedu01-nb ~]# stat oldboy.txt
  File: `oldboy.txt'
  Size: 17          Blocks: 8          IO Block: 4096   regular file
Device: 803h/2051d Inode: 262707       Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2017-09-25 16:47:54.628044632 +0800
Modify: 2017-09-25 16:47:53.059040702 +0800
Change: 2017-09-25 16:47:53.101042388 +0800
[root@oldboyedu01-nb ~]#
```

```
#mtime
[root@oldboyedu01-nb ~]# echo hello >>oldboy.txt
[root@oldboyedu01-nb ~]# stat oldboy.txt
  File: `oldboy.txt'
  Size: 23          Blocks: 8          IO Block: 4096   regular file
Device: 803h/2051d Inode: 262707       Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2017-09-25 16:47:54.628044632 +0800
Modify: 2017-10-17 21:45:10.447937840 +0800
Change: 2017-10-17 21:45:10.447937840 +0800
```

#ctime change time

```
[root@oldboyedu01-nb ~]# stat oldboy.txt
  File: `oldboy.txt'
  Size: 23          Blocks: 8          IO Block: 4096   regular file
Device: 803h/2051d Inode: 262707       Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2017-09-25 16:47:54.628044632 +0800
Modify: 2017-10-17 21:45:10.447937840 +0800
Change: 2017-10-17 21:45:10.447937840 +0800
```

```
[root@oldboyedu01-nb ~]# ln oldboy.txt oldboy.txt-hard
[root@oldboyedu01-nb ~]# stat oldboy.txt
  File: `oldboy.txt'
  Size: 23          Blocks: 8          IO Block: 4096   regular file
Device: 803h/2051d Inode: 262707       Links: 2
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2017-09-25 16:47:54.628044632 +0800
Modify: 2017-10-17 21:45:10.447937840 +0800
Change: 2017-10-17 21:47:35.328930646 +0800
```

#atime

```
[root@oldboyedu01-nb ~]#
[root@oldboyedu01-nb ~]# stat oldboy.txt
  File: `oldboy.txt'
  Size: 23          Blocks: 8          IO Block: 4096   regular file
```

```
Device: 803h/2051d Inode: 262707 Links: 2
Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)
Access: 2017-09-25 16:47:54.628044632 +0800
Modify: 2017-10-17 21:45:10.447937840 +0800
Change: 2017-10-17 21:47:35.328930646 +0800
[root@oldboyedu01-nb ~]# cat oldboy.txt
爱生活 爱李
hello
[root@oldboyedu01-nb ~]# stat oldboy.txt
  File: `oldboy.txt'
  Size: 23      Blocks: 8      IO Block: 4096   regular file
Device: 803h/2051d Inode: 262707 Links: 2
Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)
Access: 2017-10-17 21:50:13.969932450 +0800
Modify: 2017-10-17 21:45:10.447937840 +0800
Change: 2017-10-17 21:47:35.328930646 +0800
```

