

# Linux文件管理(下)

## 一、文本处理命令

### 1、查看文件内容

#### ①cat查看及合并

命令：cat

作用：查看文件内容

语法：#cat 文件名称

#cat 文件1 文件2 > 文件3

用法一：cat 文件名

示例代码：

#cat 1.txt

含义：显示1.txt文件的内容

特别注意：cat 命令用于查看文件内容时，不论文件内容有多少，都会一次性显示。如果文件非常大，那么文件开头的内容就看不到了。cat 命令适合查看不太大的文件。

```
File Edit View Search Terminal Help
[root@localhost local]# vim 1.txt
[root@localhost local]# cat 1.txt
beijing
shanghai
guangzhou
shenzhen
[root@localhost local]#
```

用法二：cat 文件1 文件2 > 文件3

示例代码：

#cat 1.txt 2.txt > 3.txt

含义：将1.txt和2.txt文件内容合并后，输出到3.txt

```
File Edit View Search Terminal Help
[root@localhost local]# cat 1.txt 2.txt > 3.txt
[root@localhost local]# cat 3.txt
beijing
shanghai
guangzhou
shenzhen
111
222
333
444
```

## ②more分屏显示文件

命令：more

作用：分屏查看文件

语法：#more 文件名

==流程：more在读取文件时，默认已经加载文件的全部内容。==

另外：more不能通过上下方向键显示上下内容，退出时会在Shell中留下刚显示的内容

```
用法一：more 文件名
示例代码：
#more /var/log/boot.log
含义：分页显示/var/log/boot.log文件的内容
```

```
root@localhost:/var/log
File Edit View Search Terminal Help
[root@localhost log]# more /var/log/boot.log █
```

more 命令的执行会打开一个交互界面，下面是一些常用交互命令：

回车键	向下移动一行。
d	向下移动半页。
空格键	向下移动一页。
b	向上移动一页，后期引入功能，早期more只能前进不能后退
q	退出 more。

```
[ OK ] Stopped dracut pre-udev hook.
[ OK ] Stopped dracut cmdline hook.
[ OK ] Stopped Create Static Device Nodes in /dev.
[ OK ] Stopped Create list of required static device nodes for the current kernel.
[ OK ] Closed udev Kernel Socket.
"boot.log-20200219" line 46 █
```

输入:f的结果

## ③less分屏显示文件

命令：less （less is more，more的增强版）

作用：分屏查看文件

语法：# less 文件名

==流程：不是加载整个文件，而是一点一点进行加载，相对而言，读取大文件时，效率比较高。==

另外：less可以通过上下方向键显示上下内容，退出时不会在Shell中留下刚显示的内容

示例代码：使用less命令查看/var/log/boot.log的文件内容

用法一：less 文件名  
示例代码：  
#less /var/log/boot.log  
含义：分页显示/var/log/boot.log文件的内容

```
File Edit View Search Terminal Help
[root@localhost log]# less /var/log/boot.log
[root@localhost log]# █
```

less 命令的执行也会打开一个交互界面，下面是一些常用交互命令（和more类似）：

回车键	向下移动一行。
d	向下移动半页。
空格键	向下移动一页。
b	向上移动一页。
上下方向键	向上与向下移动，less命令特有功能键
less -N 文件名称	显示行号
/ 字符串	搜索指定的字符串。
q	退出less

cat,more,less三者的对比

	cat	more	less
作用	显示小文件(一屏以内)	显示大文件（超过一屏）	显示大文件（超过一屏）
交互命令	无	有	有
上下键翻行	无	无	有

### ④head显示文件开头

命令：head  
作用：查看一个文件的前n 行，如果不指定n，则默认显示前10 行。  
语法：#head [参数选项] 文件名  
常见参数：-n 表示显示前n行的内容，n等于行数

用法一: head 文件名

示例代码:

#head /var/log/boot.log

含义: 显示/var/log/boot.log文件的内容, 默认为前10行

用法二: head -n 文件名

示例代码:

#head -3 /var/log/boot.log

含义: 显示/var/log/boot.log文件的前3行内容

```
File Edit View Search Terminal Help
[root@localhost ~]# head /var/log/boot.log
    Mounting Configuration File System...
[ OK ] Started Show Plymouth Boot Screen.
[ OK ] Started Forward Password Requests to Plymouth Directory Watch.
[ OK ] Reached target Paths.
[ OK ] Reached target Basic System.
[ OK ] Mounted Configuration File System.
[ OK ] Found device /dev/mapper/centos-root.
    Starting File System Check on /dev/mapper/centos-root...
[ OK ] Started File System Check on /dev/mapper/centos-root.
[ OK ] Started dracut initqueue hook.
[root@localhost ~]# head -3 /var/log/boot.log
    Mounting Configuration File System...
[ OK ] Started Show Plymouth Boot Screen.
[ OK ] Started Forward Password Requests to Plymouth Directory Watch.
[root@localhost ~]#
```

## ⑤tail显示文件结尾 (重点)

命令: tail

作用: 查看一个文件的最后n行, 如果n不指定默认显示最后10行

语法: # tail -n 文件路径【n表示数字】

常见参数: -n 显示最后n行的内容, n等于行数

-f 输出文件变化后新增加的数据

用法一: tail 文件名

示例代码:

#tail /var/log/boot.log

含义: 显示/var/log/boot.log文件的内容, 默认为最后10行

用法二: tail -n 文件名

示例代码:

#tail -5 /var/log/boot.log

含义: 显示/var/log/boot.log文件的最后5行内容

```

File Edit View Search Terminal Help
[root@localhost ~]# tail /var/log/boot.log
[ OK ] Started Notify NFS peers of a restart.
[ OK ] Started Availability of block devices.
[ OK ] Started Permit User Sessions.
[ OK ] Started Job spooling tools.
      Starting GNOME Display Manager...
[ OK ] Started Command Scheduler.
[ OK ] Started OpenSSH server daemon.
Feb  5 23:17:41 localhost NET[6784]: /etc/sysconfig/network-scripts/ifup-post : updated /etc/resolv.conf
[ OK ] Started GNOME Display Manager.
[ OK ] Started Virtualization daemon.
[root@localhost ~]# tail -5 /var/log/boot.log
[ OK ] Started Command Scheduler.
[ OK ] Started OpenSSH server daemon.
Feb  5 23:17:41 localhost NET[6784]: /etc/sysconfig/network-scripts/ifup-post : updated /etc/resolv.conf
[ OK ] Started GNOME Display Manager.
[ OK ] Started Virtualization daemon.
[root@localhost ~]#

```

用法二: `tail -f 文件名`

示例代码:

`#tail -f /var/log/messages`

含义: 显示/var/log/messages文件中, 执行tail -f 命令后, 新增的数据。

注意: 作用相当于查看一个文件动态变化的内容, 一般用于查看系统的日志的变化

按下ctrl+c可以退出查看状态

```

[root@localhost ~]# tail -f /var/log/messages
Feb  5 23:18:09 localhost dbus[6332]: [system] Successfully activated service 'org.freedesktop.problems'
Feb  5 23:18:24 localhost dbus[6332]: [system] Failed to activate service 'org.bluez': timed out
Feb  5 23:18:24 localhost pulseaudio: GetManagedObjects() failed: org.freedesktop.DBus.Error.NoReply: Did not receive a reply. Possible causes include: the remote application did not send a reply, the message bus security policy blocked the reply, the reply timeout expired, or the network connection was broken.
Feb  5 23:18:41 localhost journal: g_simple_action_set_enabled: assertion 'G_IS_SIMPLE_ACTION (simple)' failed
Feb  5 23:20:01 localhost systemd: Started Session 2 of user root.
Feb  5 23:20:56 localhost journal: g_simple_action_set_enabled: assertion 'G_IS_SIMPLE_ACTION (simple)' failed
Feb  5 23:21:22 localhost journal: g_simple_action_set_enabled: assertion 'G_IS_SIMPLE_ACTION (simple)' failed
Feb  5 23:22:56 localhost journal: g_simple_action_set_enabled: assertion 'G_IS_SIMPLE_ACTION (simple)' failed
Feb  5 23:23:14 localhost journal: g_simple_action_set_enabled: assertion 'G_IS_SIMPLE_ACTION (simple)' failed
Feb  5 23:24:23 localhost journal: g_simple_action_set_enabled: assertion 'G_IS_SIMPLE_ACTION (simple)' failed

```

注意: ctrl+c 键, 在命令行中c不再表示copy, 而表示cancel (取消), 在程序执行过程中, ==通常都可以使用ctrl+c来终止程序运行或者退出程序==

## 2、统计文件信息

### ①wc统计文件内容数量

命令: `wc`, `wc = word count`

作用: 用于统计文件内容信息 (包含行数、单词数、字节数)

语法: `# wc [参数选项] 文件名`

常见参数:

-l: 表示lines, 行数 (以回车/换行符为标准)

-w: 表示words, 单词数 依照空格来判断单词数量

-c: 表示bytes, 字节数 (空格, 回车, 换行)

用法一: `wc -lwc 文件名`

示例代码:

`#wc -lwc /var/log/boot.log`

含义: 统计/var/log/boot.log文件的行数, 单词数, 字节数

注意: `wc`命令选项可以混在一起搭配使用, 但选项的顺序不影响输出结果, 第一个是行数, 第二个是单词数, 第三个数字字节数。

```
File Edit View Search Terminal Help
[root@localhost log]# wc -lwc /var/log/boot.log
 211 1539 11934 /var/log/boot.log
[root@localhost log]#
```

行数      单词数      字节数

## ②du统计文件大小

命令: `du`

作用: 查看文件或目录(会递归显示子目录)占用磁盘空间大小

语法: `# du [参数选项] 文件名或目录名`

常见参数: `-s`: summaries, 只显示汇总的大小, 统计文件夹的大小

`-h`: 表示以高可读性的形式进行显示, 如果不写`-h`, 默认以KB的形式显示文件大小

用法一: `du 文件名`

示例代码:

`#du /var/log/boot.log`

含义: 统计/var/log/boot.log文件的大小

用法二: `du -h 文件名`

示例代码:

`#du -h /var/log/boot.log`

含义: 统计/var/log/boot.log文件的大小, 以高可读性显示

```
[root@localhost log]# du /var/log/boot.log
12      /var/log/boot.log
[root@localhost log]# du -h /var/log/boot.log
12K     /var/log/boot.log
[root@localhost log]#
```

示例代码: 统计 /var/log 目录大小

用法三: du 目录名

示例代码:

```
#du /var/log/
```

含义: 统计/var/log/目录的大小, 包含目录下每一个单独文件的大小

用法四: du -s 目录名

示例代码:

```
#du -s /var/log/
```

含义: 统计/var/log/boot.log文件的大小, 汇总只显示目录大小

用法五: du -sh 目录名

示例代码:

```
#du -sh /var/log/
```

含义: 统计/var/log/boot.log文件的大小, 汇总只显示目录大小, 并采用高可读性

```
[root@localhost ~]# du /var/log/
0      /var/log/samba/old
0      /var/log/samba
0      /var/log/ppp
0      /var/log/glusterfs
0      /var/log/amanda
100    /var/log/audit
0      /var/log/speech-dispatcher
4      /var/log/cups
0      /var/log/pluto/peer
0      /var/log/pluto
0      /var/log/chrony
0      /var/log/libvirt/qemu
0      /var/log/libvirt
0      /var/log/swtpm/libvirt/qemu
0      /var/log/swtpm/libvirt
0      /var/log/swtpm
0      /var/log/ntpstats
0      /var/log/sss
84     /var/log/gdm
4      /var/log/tuned
20     /var/log/sa
0      /var/log/qemu-ga
3428   /var/log/anaconda
0      /var/log/rhsm
4224   /var/log/
[root@localhost log]# du -s /var/log/
4224   /var/log/
[root@localhost log]# du -sh /var/log/
4.2M   /var/log/
[root@localhost log]#
```

### 3、文本处理



## ①find文件查找（重点）

命令：find

作用：用于查找文档（其选项有55 个之多）

语法：# find 路径范围 选项1 选项1的值 [选项2 选项2 的值...]

常用参数：

-name：按照文档名称进行搜索（支持模糊搜索）

\*：通配符，匹配任意个任意字符

-type：按照文档的类型进行搜索，文档类型的值，f (file)表示文件，d(directory)表示文件夹

用法一：find 路径范围 选项1 选项1的值 选项2 选项2的值

示例代码：

```
#find /var/ -name boot.log -type f
```

含义：在/var/目录下，查找名称等于boot.log，类型是文件的文档

```
[root@localhost etc]# find /var/ -name boot.log -type f
/var/log/boot.log
```

路径范围	选项1	选项1的值	选项2	选项2的值
/var/	-name	boot.log	-type	f

A var表示要查找的目录名字

B var表示，在var目录内查找

用法二：find 路径范围 选项1 选项1的值使用通配符 选项2 选项2的值

示例代码：

```
#find /var/log -name "*.log" -type f
```

含义：在/var/log目录下，查找所有.log结尾，类型是文件的文档，\*.log需要用引号引起来。



```
[root@localhost etc]# find /var/log -name "*.log" -type f
/var/log/audit/audit.log
/var/log/gdm/:0.log
/var/log/gdm/:0-greeter.log
/var/log/tuned/tuned.log
/var/log/anaconda/anaconda.log
/var/log/anaconda/X.log
/var/log/anaconda/program.log
/var/log/anaconda/packaging.log
/var/log/anaconda/storage.log
/var/log/anaconda/ifcfg.log
/var/log/anaconda/ks-script-pp8qZS.log
/var/log/anaconda/journal.log
/var/log/boot.log
/var/log/vmware-vmtoolsd.log
/var/log/vmware-network.log
/var/log/Xorg.9.log
/var/log/wpa_supplicant.log
/var/log/yum.log
/var/log/Xorg.0.log
/var/log/vmware-vmtoolsd.log
[root@localhost etc]# █
```

---

用法一：find 路径范围 选项1 选项1的值

示例代码：

#find /var/log -type d

含义：在/var/log目录下，查找所有文件夹

```
[root@localhost etc]# find /var/log -type d
/var/log
/var/log/samba
/var/log/samba/old
/var/log/ppp
/var/log/glusterfs
/var/log/amanda
/var/log/audit
/var/log/speech-dispatcher
/var/log/cups
/var/log/pluto
/var/log/pluto/peer
/var/log/chrony
/var/log/libvirt
/var/log/libvirt/qemu
/var/log/swtpm
/var/log/swtpm/libvirt
/var/log/swtpm/libvirt/qemu
/var/log/ntpstats
/var/log/sss
/var/log/gdm
/var/log/tuned
/var/log/sa
/var/log/qemu-ga
/var/log/anaconda
/var/log/rhsm
[root@localhost etc]# █
```

## ②grep搜索文件内容（重要）

命令：grep

作用：在文件中直接找到包含指定信息的那些行，并把这些信息显示出来

语法：#grep 要查找的内容 文件名

用法一: grep 查找的内容 文件名

示例代码:

```
#grep network boot.log
```

含义: 在boot.log文件中, 查找包含network的行

```
File Edit View Search Terminal Help
[root@localhost log]# grep network boot.log
Starting Read and set NIS domainname from /etc/sysconfig/network...
[ OK ] Started Read and set NIS domainname from /etc/sysconfig/network.
Starting Import network configuration from initramfs...
[ OK ] Started Import network configuration from initramfs.
Starting LSB: Bring up/down networking...
[ OK ] Started LSB: Bring up/down networking.
[root@localhost log]#
```

用法二: grep 查找的内容 多个文件

示例代码:

```
#grep network /var/log/*
```

含义: 在/var/log目录下的所有文件中, 查找包含network的行

```
File Edit View Search Terminal Help
[root@localhost log]# grep network /var/log/*
grep: /var/log/amanda: Is a directory
grep: /var/log/anaconda: Is a directory
grep: /var/log/audit: Is a directory
/var/log/boot.log: Starting Read and set NIS domainname from /etc/sysconfig/network...
/var/log/boot.log:[ OK ] Started Read and set NIS domainname from /etc/sysconfig/network.
/var/log/boot.log: Starting Import network configuration from initramfs.
```

### ③输出重定向

场景: 一般命令的输出都会显示在终端中, 有些时候需要将一些命令的执行结果想要保存到文件中进行后续的分析/统计, 则这时候需要使用到的输出重定向技术。

>: 标准输出重定向:覆盖输出, 会覆盖掉原先的文件内容

>>: 追加重定向:追加输出, 不会覆盖原始文件内容, 会在原始内容末尾继续添加

语法: #有输出的命令 重定向符号 文件名

用法一: 使用 > 进行覆盖输出

示例代码:

```
#ll > llroot.txt
```

含义: 将ll命令输出的内容, 写入到llroot.txt文件中, 如果llroot.txt文件中有其他内容, 则先清空, 在写入

注意: 文件路径中的文件可以是不存在的文件, 例如llroot.txt可以不存在, 执行命令时会自动创建这个文件

```
File Edit View Search Terminal Help
[root@localhost ~]# ll
total 12
-rw-r--r--. 1 root root    0 Feb  9 16:53 1.txt
-rw-----. 1 root root 1755 Feb  4 22:15 anaconda-ks.cfg
drwxr-xr-x. 2 root root   36 Feb  8 17:18 chuanzhiboke
drwxr-xr-x. 2 root root    6 Feb  7 12:00 Desktop
drwxr-xr-x. 2 root root    6 Feb  4 22:34 Documents
drwxr-xr-x. 2 root root    6 Feb  4 22:34 Downloads
-rw-r--r--. 1 root root    0 Feb  8 16:55 heimacxy.txt
-rw-r--r--. 1 root root 1803 Feb  4 22:20 initial-setup-ks.cfg
drwxr-xr-x. 2 root root    6 Feb  4 22:34 Music
drwxr-xr-x. 2 root root 4096 Feb 10 15:52 Pictures
drwxr-xr-x. 2 root root    6 Feb  4 22:34 Public
drwxr-xr-x. 2 root root    6 Feb  4 22:34 Templates
drwxr-xr-x. 2 root root    6 Feb  4 22:34 Videos
[root@localhost ~]# ll > llroot.txt
[root@localhost ~]# cat llroot.txt
total 12
-rw-r--r--. 1 root root    0 Feb  9 16:53 1.txt
-rw-----. 1 root root 1755 Feb  4 22:15 anaconda-ks.cfg
drwxr-xr-x. 2 root root   36 Feb  8 17:18 chuanzhiboke
```

用法二：使用 >> 进行追加输出

示例代码：

```
#ll >> llroot.txt
```

含义：将ll命令输出的内容，追加写入到llroot.txt文件中

注意：文件路径中的文件可以是不存在的文件，例如llroot.txt可以不存在，执行命令时会自动创建这个文件

```
File Edit View Search Terminal Help
[root@localhost ~]# cat llroot.txt
12345678
abcdefg
[root@localhost ~]# ll >> llroot.txt
[root@localhost ~]# cat llroot.txt
12345678
abcdefg
total 16
-rw-r--r--. 1 root root    0 Feb  9 16:53 1.txt
-rw-----. 1 root root 1755 Feb  4 22:15 anaconda-ks.cfg
drwxr-xr-x. 2 root root   36 Feb  8 17:18 chuanzhiboke
drwxr-xr-x. 2 root root    6 Feb  7 12:00 Desktop
```