

第一章 Linux 命令

Linux 服务器端一般是命令行界面，通过命令和 linux 系统交互。通过命令进行各种操作。

本课程的讲解的命令分类：磁盘管理；文件管理；系统设置；压缩/解压；网络通讯；网络访问；权限管理；管道和重定向；vi 编辑命令；安装软件命令。

1 磁盘管理

(1) 切换目录：cd 目录名

① 由当前目录切换到 /usr/local

例如：cd /usr/local

② 回到根目录（类似于我的电脑）

例如：cd /

(2) 列出当前目录下目录及文件：ll, ls

① 列出目录下的详细内容（相当于 Windows 的列表查看）

例如：在当前目录下执行 ll

```
[root@localhost centos]# ll
total 0
drwxr-xr-x. 2 centos centos 6 Aug  6 00:09 Desktop
drwxr-xr-x. 2 centos centos 6 Aug  6 00:09 Documents
drwxr-xr-x. 2 centos centos 6 Aug  6 00:09 Downloads
drwxr-xr-x. 2 centos centos 6 Aug  6 00:09 Music
drwxr-xr-x. 2 centos centos 6 Aug  6 00:09 Pictures
drwxr-xr-x. 2 centos centos 6 Aug  6 00:09 Public
drwxr-xr-x. 2 centos centos 6 Aug  6 00:09 Templates
drwxr-xr-x. 2 centos centos 6 Aug  6 00:09 Videos
```

② 使用 ll 目录名 查看指定目录下的文件和目录列表

```
[root@localhost centos]# ll /usr/local/
total 0
drwxr-xr-x. 2 root root  6 Nov  5 2016 bin
drwxr-xr-x. 2 root root  6 Nov  5 2016 etc
drwxr-xr-x. 2 root root  6 Nov  5 2016 games
drwxr-xr-x. 2 root root  6 Nov  5 2016 include
drwxr-xr-x. 2 root root  6 Nov  5 2016 lib
drwxr-xr-x. 2 root root  6 Nov  5 2016 lib64
drwxr-xr-x. 2 root root  6 Nov  5 2016 libexec
drwxr-xr-x. 2 root root  6 Nov  5 2016 sbin
drwxr-xr-x. 5 root root 49 Aug  5 23:11 share
drwxr-xr-x. 2 root root  6 Nov  5 2016 src
```

-rw-rw-rw - 表示文件

drw-rw-rw d 表示目录文件夹

③ 只列出文件/目录名

例如：在当前目录执行 ls

```
[root@localhost centos]# ls
Desktop Documents Downloads Music Pictures Public Templates Videos
```

(3) 显示当前所在目录位置： pwd

例如：在当前目录下执行 pwd

```
[root@localhost test]# pwd
/home/test
```

2 文件管理

(1) 创建目录: mkdir 目录名

例如: mkdir test

```
[root@localhost home]# mkdir test
```

查看创建的目录 ls

```
[root@localhost home]# ls  
centos test
```

(2) 删除文件或目录 rm , rm -rf

在 home 目录下测试

```
cd /home
```

① 使用 rm 删除文件

新建空文件

```
touch my.txt
```

例如: rm my.txt

在 test01 目录执行 echo "ok" > my.txt 创建一个 my.txt 的文件

查看

```
cat my.txt
```

在执行 rm mytxt.txt 回车后, 系统会提示是否删除确认

```
[root@localhost test01]# pwd
/home/myfolder/test01
[root@localhost test01]# echo "ok" > my.txt
[root@localhost test01]# ls
my.txt
[root@localhost test01]# rm my.txt
rm: remove regular file 'my.txt'? █
```

在提示位置输入 `y`,表示同意, 文件删除。 输入 `n` 取消删除操作。

```
[root@localhost test01]# rm my.txt
rm: remove regular file 'my.txt'? y
```

② 不用确认的直接删除

例如: `rm -f my.txt`

```
[root@localhost test01]# rm -f my.txt
```

③ 删除目录

`rm` 目录

例如: `rm -rf test01`

`rm` 不加参数, 不能删除目录

```
[root@localhost myfolder]# rm test01/
rm: cannot remove 'test01/': Is a directory
```

`rm -rf test01` 删除文件夹 `r` 表示递归删除, `f` 不做确认提示

```
[root@localhost myfolder]# rm -rf test01
[root@localhost myfolder]# ls
[root@localhost myfolder]# ll
total 0
```

(3) 复制文件命令 cp , cp -rf

① 复制文件: aa.txt 是新的文件名

语法: cp 被复制的文件名 新文件名

例如: cp anaconda-ks.cfg aa.txt

```
[root@localhost ~]# cd /root/
[root@localhost ~]# cp anaconda-ks.cfg aa.txt
[root@localhost ~]# ls
aa.txt          Documents      Music          Templates
anaconda-ks.cfg Downloads     Pictures       Videos
Desktop        initial-setup-ks.cfg Public
```

重复执行命令, 询问是否覆盖

```
[root@localhost ~]# cp anaconda-ks.cfg aa.txt
cp: overwrite 'aa.txt'? █
```

输入 "y" 同意覆盖, 输入 "n" 取消操作

② 复制文件夹

语法: cp -rf 文件夹 新的文件夹名

例如 cp -rf myfolder newfolder

(4) 查看文件全部内容: cat 文件路径

例如: cat /root/anaconda-ks.cfg

```
[root@localhost ~]# cat /root/anaconda-ks.cfg
#version=DEVEL
# System authorization information
auth --enablesshadow --passalgo=sha512
# Use CDROM installation media
cdrom
# Use graphical install
```

(5) 分页查看文件内容: more 文件路径

逐屏输出(按空白键 space 就往下一页)、Enter 一行一行展示

例如: more /root/anaconda-ks.cfg (命令输入完按 Enter)

```
[root@localhost ~]# more /root/anaconda-ks.cfg
#version=DEVEL
# System authorization information
auth --enablesshadow --passalgo=sha512
# Use CDROM installation media
```

屏幕底下, 显示当前屏幕展示总内容的百分比

```
# Root password
rootpw --iscrypted $6$X9Jp.1X0uyJliV5B$z13izmbzkwMdFPmu
rCzKkNCIL4w4RekkCz6ps0Vkw9mbq6/ABpkKabguk77erLdK7dRGW0
--More-- (40%)
```

按 Enter 之后

```
# Root password
rootpw --iscrypted $6$X9Jp.1X0uyJliV5B$z13izf
rCzKKNCIL4w4RekkCz6ps0Vkw9mbq6/ABpkKabguk77e
# System services
--More-- (41%)
```

按空格之后

```
bootloader --append=" crashkernel=auto" --location=mbr
autopart --type=lvm
# Partition clearing information
clearpart --none --initlabel

%packages
@^gnome-desktop-environment
@base
@core
@desktop-debugging
@dialect-up
@directory-client
@fonts
@gnome-desktop
--More-- (69%)
```

(6) 查看文件开头的 n 行数据: head -n 数字

② 例如: head /root/anaconda-ks.cfg (默认是 10 行)

```
[root@localhost ~]#
[root@localhost ~]# head /root/anaconda-ks.cfg
#version=DEVEL
# System authorization information
auth --enableshadow --passalgo=sha512
# Use CDROM installation media
cdrom
# Use graphical install
graphical
# Run the Setup Agent on first boot
firstboot --enable
ignoredisk --only-use=sda
[root@localhost ~]#
```

显示文件开头的10行

② 例如显示文件开始的指定行数

```
[root@localhost ~]# head /root/anaconda-ks.cfg -n 12
#version=DEVEL
# System authorization information
auth --enableshadow --passalgo=sha512
# Use CDROM installation media
cdrom
# Use graphical install
graphical
# Run the Setup Agent on first boot
firstboot --enable
ignoredisk --only-use=sda
# Keyboard layouts
keyboard --vckeymap=us --xlayouts='us','cn'
[root@localhost ~]#
```

显示了文件开头的12行

(7) 显示文件尾部的 n 行数据 tail -n 数字，默认是 10 行

① 例如: tail /root/anaconda-ks.cfg

```
[root@localhost ~]# tail /root/anaconda-ks.cfg
%addon com_redhat_kdump --enable --reserve-mb='auto'
%end
%anaconda
```

② 例如: 显示文件尾部指定的 n 行

```
[root@localhost ~]# tail /root/anaconda-ks.cfg -n 2
pwpolicy luks --minlen=6 --minquality=50 --notstrict --nochanges --notempty
%end
[root@localhost ~]#
```

```
[root@localhost ~]# cd /home/
[root@localhost home]# ll
total 4
drwx-----. 14 centos centos 4096 Aug  6 18:18 centos
drwxr-xr-x.  2 root  root    6 Aug  6 22:41 myfolder
[root@localhost home]# cp -rf myfolder newfolder
[root@localhost home]# ll
total 4
drwx-----. 14 centos centos 4096 Aug  6 18:18 centos
drwxr-xr-x.  2 root  root    6 Aug  6 22:41 myfolder
drwxr-xr-x.  2 root  root    6 Aug  8 21:46 newfolder
[root@localhost home]#
```

(8) 文件内搜索: grep

在文件中(可以多个文件同时)搜索字符串, 可以使用正则表达式

语法: `grep [参数] 搜索的字符串内容 文件名 1 [文件 n]`

准备一个文件 `aa.txt`, 内容如下。 `test.txt` 内容同 `aa.txt`

```
[root@localhost mytest]# cat aa.txt
hello abc
abc this is an idea
java is programing
mysql
tomcat java
javascript
mysql
html
mysql
html
JAVA
ok
```

① 搜索文本 "java" : `grep java aa.txt`

```
[root@localhost mytest]# grep java aa.txt
java is programing
tomcat java
javascript
```

没有大写的 "JAVA" ,默认是区分大小写的搜索

② 搜索文本“ java” 区分大小写: `grep -i java aa.txt`

-i: 不区分大小写

```
[root@localhost mytest]# grep -i java aa.txt  
java is programing  
tomcat java  
javascript  
JAVA
```

③ 搜索的文本中有空格, 使用引号括起来

例如 `grep "java is" aa.txt`

```
[root@localhost mytest]#  
[root@localhost mytest]# grep "java is" aa.txt  
java is programing  
[root@localhost mytest]#
```

④ 搜索整个单词, 是其他字符串的一部分的不符合条件

-w : 搜索单词

搜索单词 java : `grep -w java aa.txt`

```
[root@localhost mytest]#  
[root@localhost mytest]# grep -w java aa.txt  
java is programing  
tomcat java  
[root@localhost mytest]#
```

⑤ 使用正则, 以java开头的行

例如 `grep "^java" aa.txt`

```
[root@localhost mytest]#  
[root@localhost mytest]# grep "^java" aa.txt  
java is programing  
javascript
```

⑥ 在多个文件搜索

例如 `grep java aa.txt test.txt`

```
[root@localhost mytest]# grep java aa.txt test.txt  
aa.txt:java is programing  
aa.txt:tomcat java  
aa.txt:javascript  
test.txt:java is programing  
test.txt:tomcat java  
test.txt:javascript
```

⑦ 使用管道 “|”

例如: `cat aa.txt | grep java`

3 系统命令

(1) 显示系统日期和时间: `date`

显示系统当前时间

例如: `date`

```
[root@localhost myfolder]# date  
Wed Aug 9 10:14:51 CST 2017
```

(2) 切换用户: `su 用户名`

以其他用户身份使用系统, (类似 windows10 系统, 有些程序以

管理员身份执行)

① 从 root 用户切换到普通用户 (使用安装 centos 时的普通用户 centos)

例如: 以 root 用户登录系统, 执行 su centos

```
[root@localhost myfolder]# su centos
'abrt-cli status' timed out
[centos@localhost myfolder]$ ls
[centos@localhost myfolder]$ cd /home/
[centos@localhost home]$ ls
aa.txt abc.log centos myfolder newfolder
[centos@localhost home]$
```

注意命令提示符 “#” 变为 “\$” , root@localhost 变为 centos@localhost

② 从普通用户切换到 root 用户, 需要输入 root 用户密码

例如 su root

```
[centos@localhost home]$ su root
Password:
[root@localhost home]#
```

Password 输入密码, linux 不显示输入的字符, 输入密码后直接按 “Enter” .

(3) 清楚屏幕: clear

例如: clear

```
[root@localhost myfolder]# date
Wed Aug  9 10:20:04 CST 2017
[root@localhost myfolder]# clear
```

ctrl+L

(4) 重启系统: **reboot**

例如: `reboot`

(5) 关机: **shutdown -h now**

例如: `shutdown -h now`

(6) 查看系统进程: **ps -ef**

`ps` [命令参数]

常用参数:

-e :显示当前所有进程

-f :显示 UID,PPID,C 与 STIME 栏位信息

UID:拥有改程序的用户

PID:程序的进程 id

PPID:父进程的 id

C: CPU 使用的资源百分比

STIME :系统启动时间

TTY: 登录系统的终端位置 (客户端的标识)

TIME:使用掉的 CPU 时间

CMD:进程是有哪些程序启动的

① 查看系统当前运行的进程（程序）

例如：在任意位置执行 `ps -ef`

```
[root@localhost myfolder]#  
[root@localhost myfolder]# ps -ef  
UID          PID     PPID  C  STIME TTY          TIME CMD  
root         1       0  0  09:23 ?            00:00:03 /usr/lib/systemd/systemd  
root         2       0  0  09:23 ?            00:00:00 [kthreadd]  
root         3       2  0  09:23 ?            00:00:00 [ksoftirqd/0]  
root         6       2  0  09:23 ?            00:00:00 [kworker/u256:0]
```

(7) kill 进程: kill

例如:

`kill pid`: 杀掉进程

`kill -9 pid` 强制杀掉进程

例如：先使用 `ps -ef` 查询进程 id ， 然后再使用 `kill pid` ， 这是例子不要真的执行

```
[root@localhost myfolder]# ps -ef | head  
UID          PID     PPID  C  STIME TTY          TIME CMD  
root         1       0  0  09:23 ?            00:00:03 /usr/lib/systemd/systemd  
wicked-root --system --deserialize 21  
root         2       0  0  09:23 ?            00:00:00 [kthreadd]  
root         3       2  0  09:23 ?            00:00:00 [ksoftirqd/0]  
root         6       2  0  09:23 ?            00:00:00 [kworker/u256:0]  
root         7       2  0  09:23 ?            00:00:00 [migration/0]  
root         8       2  0  09:23 ?            00:00:00 [rcu_bh]  
root         9       2  0  09:23 ?            00:00:00 [rcu_sched]  
root        10       2  0  09:23 ?            00:00:00 [watchdog/0]  
root        12       2  0  09:23 ?            00:00:00 [khelper]  
[root@localhost myfolder]# kill 2
```

不能随意使用，会被系统进程杀掉，程序不能使用。

4 压缩/解压

(1) tar 压缩 (归档)

tar 用来压缩和解压文件, tar 压缩后的文件也叫归档文件。

语法: tar 参数 要压缩或解压的文件或目录

常用参数:

z: 使用压缩, 生成的文件名是 xxx.tar.gz 这是 linux 中常用的压缩格式。

c: 创建压缩文档

v: 显示压缩, 解压过程中处理的文件名

f: 指定归档文件名, tar 参数后面是归档文件名

x: 从归档文件中释放文件, 就是解压。

t: 列出归档文件内容, 查看文件内容

C: 解压到指定目录, 使用方式 -C 目录, C 是大写的。

① 创建归档文件 (压缩)

语法: tar -zvcf 归档文件名 要归档文件列表

例如: tar -zcvf txtfile.tar.gz aa.txt

```
[root@localhost mytest]# ls
aa.txt  test.txt
[root@localhost mytest]# tar -zcvf txtfile.tar.gz aa.txt
aa.txt
[root@localhost mytest]#
[root@localhost mytest]#
[root@localhost mytest]# ll
total 12
-rw-r--r--. 1 root root 108 Aug  9 14:59 aa.txt
-rw-r--r--. 1 root root 108 Aug  9 15:01 test.txt
-rw-r--r--. 1 root root 192 Aug  9 20:09 txtfile.tar.gz
[root@localhost mytest]#
```

tar 命令执行结果

② 归档多个文件

例如 tar -zcvf txt.tar.gz aa.txt test.txt

```
[root@localhost mytest]# ll
total 8
-rw-r--r--. 1 root root 108 Aug  9 14:59 aa.txt
-rw-r--r--. 1 root root 108 Aug  9 15:01 test.txt
[root@localhost mytest]# tar -zcvf txt.tar.gz aa.txt test.txt
aa.txt
test.txt
[root@localhost mytest]# ls
aa.txt  test.txt  txt.tar.gz
[root@localhost mytest]#
```

也可以使用通配符 * 表示要压缩的文件名的一部分

例如: tar -zcvf txt2.tar.gz *.txt

```
[root@localhost mytest]# rm txt.tar.gz -f
[root@localhost mytest]# ls
aa.txt  test.txt
[root@localhost mytest]# tar -zcvf txt2.tar.gz *.txt
aa.txt
test.txt
[root@localhost mytest]# ls
aa.txt  test.txt  txt2.tar.gz
[root@localhost mytest]#
```

③ 压缩目录

语法: tar -zcvf 归档文件名 目录

例如: tar -zcvf file.tar.gz mytest

```
[root@localhost mytest]# ls
aa.txt  test.txt
[root@localhost mytest]# cd ..  切换到mytest的上级目录
[root@localhost home]# ls
centos  mytest  ←home目录下有mytest文件夹
[root@localhost home]# tar -zcvf file.tar.gz mytest
mytest/
mytest/.mozilla/
mytest/.mozilla/extensions/
mytest/.mozilla/plugins/
mytest/.bash_logout
mytest/.bash_profile  ←v参数的作用,显示处理的
mytest/.bashrc          文件列表,包含了隐藏文件
mytest/.cache/
mytest/.cache/abrt/
mytest/.cache/abrt/lastnotification
mytest/.config/
mytest/.config/abrt/
mytest/aa.txt
mytest/test.txt
[root@localhost home]#
```

结果:

```
[root@localhost home]# ls
centos  file.tar.gz  mytest
```

(2) 查看归档 (压缩) 文件

显示压缩文件的内容。

语法: tar -tf 归档文件名

t: 列出归档 (压缩) 文件的内容

f: 归档文件名

例如: tar -tf file2.tar.gz

```
[root@localhost home]# tar -tf file2.tar.gz
mytest/aa.txt
mytest/test.txt
[root@localhost home]#
```

(3) tar 解压

解压: 释放压缩的文件内容

语法: tar -zxvf 已归档的文件名。

参数: x: 解压,从归档文件中释放文件,

解压 file.tar.gz

语法: tar -zxvf file.tar.gz

```
[root@localhost home]# ls
centos file2.tar.gz file3.tar file.tar.gz mytest
[root@localhost home]# rm -rf mytest/ ← 先删除原来的mytest目录
[root@localhost home]# ls
centos file2.tar.gz file3.tar file.tar.gz
[root@localhost home]# tar -zxvf file.tar.gz
mytest/
mytest/.mozilla/
mytest/.mozilla/extensions/
mytest/.mozilla/plugins/
mytest/.bash_logout
mytest/.bash_profile
mytest/.bashrc
mytest/.cache/
mytest/.cache/abrt/
mytest/.cache/abrt/lastnotification
mytest/.config/
mytest/.config/abrt/
mytest/aa.txt
mytest/test.txt
[root@localhost home]# ls
centos file2.tar.gz file3.tar file.tar.gz mytest
```

原来file.tar.gz就是压缩的mytest目录下的所有内容

解压后,还原归档文件中的内容

5 网络通讯

(1) 查看 ip 信息: ifconfig

语法 ifconfig

例如: 在任意位置输入 ifconfig

```
[root@localhost home]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.103 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::f23:77f3:90c:239f prefixlen 64 scopeid 0x20<link>
```

(2) 测试网络连通: ping

语法: ping ip 或者域名

例如: ping www.baidu.com

查看之后, ctrl+C, 退出查看, 回到 Linux 命令行

```
[root@localhost home]# ping www.baidu.com
PING www.a.shifen.com (119.75.216.20) 56(84) bytes of data:
64 bytes from 127.0.0.1 (119.75.216.20): icmp_seq=1 ttl=56 time=5.77 ms
64 bytes from 127.0.0.1 (119.75.216.20): icmp_seq=2 ttl=56 time=19.6 ms
64 bytes from 127.0.0.1 (119.75.216.20): icmp_seq=3 ttl=56 time=5.29 ms
^C
--- www.a.shifen.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2008ms
rtt min/avg/max/mdev = 5.291/10.247/19.678/6.671 ms
```

```
[root@localhost home]# ping 192.168.1.102
PING 192.168.1.102 (192.168.1.102) 56(84) bytes of data.
64 bytes from 192.168.1.102: icmp_seq=1 ttl=64 time=0.539 ms
64 bytes from 192.168.1.102: icmp_seq=2 ttl=64 time=0.361 ms
^C
--- 192.168.1.102 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.361/0.450/0.539/0.089 ms
```

6 网络访问

(1) curl : 使用 url 访问网络的文件传输工具。

curl 是利用 URL 语法在命令行方式下工作的开源文件传输工具

常用来①：测试网络访问；②：模拟用户访问

抓取百度的首页内容

例如：curl www.baidu.com

```
[root@localhost centos]# curl www.baidu.com
<!DOCTYPE html>
<!--STATUS OK--><html> <head><meta http-equiv=content-type content=text/html;c
harset=utf-8><meta http-equiv=X-UA-Compatible content=IE=Edge><meta content=al
ways name=referrer><link rel=stylesheet type=text/css href=http://s1.bdstatic.
com/r/www/cache/bdorz/baidu.min.css><title>百度一下, 你就知道</title></head> <
```

模拟用户访问

例如：查询 studentId=5 的学生。

```
[root@localhost centos]# curl http://192.168.1.101/myweb/queryStudent.do?sid=5
curl: (7) Failed connect to 192.168.1.101:80; No route to host
[root@localhost centos]#
```

(2) wget : 下载

wget linux 上下载资源的工具，比如下载软件的安装包。一般软件

都会给出一个下载地址

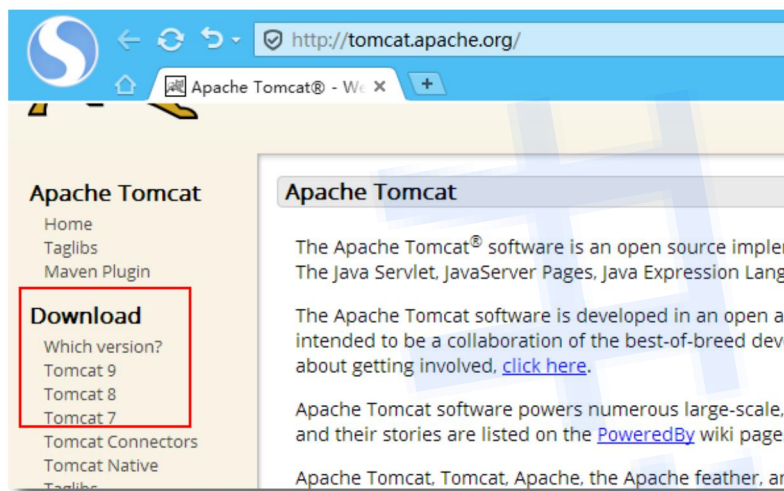
先安装 wget 命令

```
yum -y install wget
```

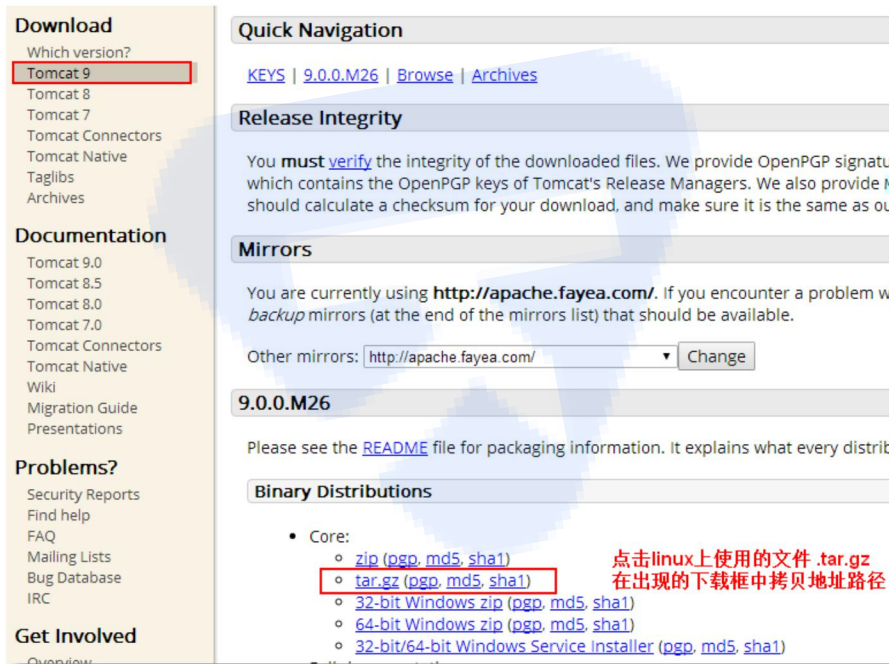
语法: wget 下载的资源地址

例如: 下载一个 tomcat 软件。

首先要得到 tomcat 的下载地址, 同浏览器下载 tomcat, 得到下载地址



点击左侧 Download “Tomcat 9”



不同浏览器，操作方式略有不同



tomcat 9 下载地址：

<http://apache.fayea.com/tomcat/tomcat-9/v9.0.0.M26/bin/apache-tomcat-9.0.0.M26.tar.gz>

在 linux 中使用 wget tomcat 下载地址

粘贴地址提示：右键有 copy 按钮或者使用 shift + insert 快捷键

```
[root@localhost mytest]# ll
total 8
-rw-r--r--. 1 root root 26 Aug 12 21:58 t1.txt
-rw-r--r--. 1 root root 108 Aug 9 15:01 test.txt
[root@localhost mytest]#
[root@localhost mytest]# wget http://apache.fayea.com/tomcat/tomcat-9/v9.0.0.M26/bin/apache-tomcat-9.0.0.M26.tar.gz
--2017-08-13 16:12:16-- http://apache.fayea.com/tomcat/tomcat-9/v9.0.0.M26/bin/apache-tomcat-9.0.0.M26.tar.gz
Resolving apache.fayea.com (apache.fayea.com)... 119.6.242.165, 119.6.242.164
Connecting to apache.fayea.com (apache.fayea.com)|119.6.242.165|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9376594 (8.9M) [application/x-gzip]
Saving to: 'apache-tomcat-9.0.0.M26.tar.gz'
19% [=====>] 1,864,501 686KB/s
```

进度 100%，下载完成

```
100%[=====] 9,376,594 503KB/s in 18s
2017-08-13 16:12:34 (504 KB/s) - 'apache-tomcat-9.0.0.M26.tar.gz' saved [9376594/9376594]
```

查看当前目录内容，下载的文件：apache-tomcat-9.0.0.M26.tar.gz

```
[root@localhost mytest]# ll
total 9168
-rw-r--r--. 1 root root 9376594 Aug 3 05:01 apache-tomcat-9.0.0.M26.tar.gz
-rw-r--r--. 1 root root 26 Aug 12 21:58 t1.txt
-rw-r--r--. 1 root root 108 Aug 9 15:01 test.txt
[root@localhost mytest]#
```

7 权限管理

2-7-1 介绍:

权限指用户或程序能够对目录，文件执行的操作。

执行 ll 或 ls -l 显示文件和目录详细信息

```

[root@localhost ~]# ll
total 40
-rw-r--r--. 1 root root 0 Aug 5 23:23 1
lrwxrwxrwx. 1 root root 7 Aug 5 23:11 bin -> usr/bin
dr-xr-xr-x. 4 root root 4096 Aug 5 23:52 boot
drwxr-xr-x. 20 root root 3300 Aug 11 21:02 dev
drwxr-xr-x. 138 root root 8192 Aug 11 21:01 etc
drwxr-xr-x. 4 root root 65 Aug 9 21:10 home
lrwxrwxrwx. 1 root root 7 Aug 5 23:11 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 Aug 5 23:11 lib64 -> usr/lib64
drwxr-xr-x. 2 root root 6 Nov 5 2016 media
drwxr-xr-x. 2 root root 6 Nov 5 2016 mnt
drwxr-xr-x. 3 root root 16 Aug 5 23:38 opt

```

文件类型 | 链接数 | 所有者 | 所有者所在的组 | 文件大小, 字节单位 | 修改时间 | 文件名, -> 链接文件指向的目标文件

- : 表示文件

l : 软链接文件 (windows 快捷方式)

d : 目录

c : 字符设备文件, 一次传输一个字节的设备被称为字符设备。例如键盘, 鼠标

linux 权限机制采用 UGO 模式。其中 u(user) 表示所属用户、g(group) 表示所属组、o(other) 表示除了所属用户、所属组之外的情况。



A、 读权限：读取文件内容，对文件执行 cat ,more ,less 这样的查看命令，对目录来说可以读取目录中文件列表

r--read 读权限 4

B、 写权限：编辑、新增、修改文件内容

w--write 写权限 2

C、 执行权限：进入目录，访问目录中的文件内容。

x--execute 执行权限 1

D、 常见权限设置

$rwX = 4 + 2 + 1 = 7$

常见 644、755、777 三种权限

创建文件用户就是文件的拥有者，用户所在的组就是文件的所在组。除了创建文件的用户都是其他用户。root 有最高的权限。

E、 权限的尝试

以 root 身份登录系统，执行 cd / 切换到根目录，再执行 ls -l，查看 root 目录的权限如下：

```
dr-xr-x---. 15 root root 4096 Aug 11 21:04 root
```

表明其他用户对 root 目录没有任何权限，查看目录内容都不行。

实验：切换到 centos 普通用户，查看 root 目的内容

```
[root@localhost ~]#  
[root@localhost ~]# su centos  
[centos@localhost ~]$ cd /  
[centos@localhost ~]$ ll root  
ls: cannot open directory root: Permission denied  
[centos@localhost ~]$  
[centos@localhost ~]$
```

1.1.2 2-7-2 权限设置

(1) 修改文件权限 chmod

语法 chmod UGO 权限 文件/目录

例如：先以 root 用户创建文件 aa.txt，查看文件的权限，其他用户只要 r 读的权限。centos 对 aa.txt 来说就是其他用户。

```
[root@localhost ~]# echo "hello linux" > aa.txt  
[root@localhost ~]# ll aa.txt  
-rw-r--r--. 1 root root 12 Aug 11 22:59 aa.txt  
[root@localhost ~]#
```

adduser 时，创建用户的过程更像是一种人机对话，系统会提示你输入各种信息，然后会根据这些信息帮你创建新用户

```
adduser centos
```

```
passwd centos12345
```

锁定用户 centos 账号(重开一个 session 测试)

```
usermod -L centos 锁定后账号不可使用
```

解锁用户 centos 账号

```
usermod -U centos
```

切换到 centos 用户执行， cat aa.txt 是没有问题的

```
[root@localhost ~]# su centos
[centos@localhost ~]$ cat aa.txt
hello linux
[centos@localhost ~]$
```

修改 aa.txt 内容, 执行 vi aa.txt ,在保存文件后报错。

```
edit |hello linux
~
~
~
~
-- INSERT -- W10: Warning: Changing a readonly file
centos@localhost:~
```

使用 root 用户给 centos 分配 w 权限。切换回 root 用户 (root 用户是 aa.txt 的拥有者)。分配权限方式: 拥有者的不变 rw=6, 所在组设置 0 没有权限, 主要给其他用户设置 rw=6。

```
[root@localhost ~]# ll aa.txt
-rw-r--r--. 1 root root 12 Aug 11 23:11 aa.txt
[root@localhost ~]# chmod 606 aa.txt
[root@localhost ~]# ll aa.txt
-rw----rw-. 1 root root 12 Aug 11 23:11 aa.txt
```

切换到 centos

```
[root@localhost ~]#
[root@localhost ~]# su centos
[centos@localhost ~]# ll aa.txt
-rw----rw-. 1 root root 12 Aug 11 23:17 aa.txt
[centos@localhost ~]# cat aa.txt
hello linux
[centos@localhost ~]# vi aa.txt
edit file hello linux
[centos@localhost ~]# cat aa.txt
hello linux
[centos@localhost ~]#
```

注意要给 centos 分配对 aa.txt 的读 (r) 权限。否则不能查看文件内容。

```
[root@localhost ~]# chmod 602 aa.txt
[root@localhost ~]# ll aa.txt
-rw----w-. 1 root root 22 Aug 11 23:18 aa.txt
[root@localhost ~]# su centos
[centos@localhost ~]$ cat aa.txt
cat: aa.txt: Permission denied
[centos@localhost ~]$
```

(2) 修改文件所有者 chown

语法: chown 新的所有者用户 被修改的文件

例如: 修改原来 root 拥有的文件 aa.txt 改为 centos, 所有者有操作的读写权限。

```
chown centos aa.txt
```

```
[root@localhost ~]#
[root@localhost ~]# ll aa.txt
-rw----w-. 1 root root 22 Aug 11 23:18 aa.txt
[root@localhost ~]# chown centos aa.txt
[root@localhost ~]# ll aa.txt
-rw----w-. 1 centos root 22 Aug 11 23:18 aa.txt
[root@localhost ~]#
```

8 管道和重定向

(1) 重定向输出覆盖: >

向文件输出内容, 文件不存在则新建文件再输出; 文件存在先清

空文件，再写入内容。

例如： `echo "write some" > t1.txt`

```
[root@localhost mytest]# ll
total 8
-rw-r--r--. 1 root root 108 Aug  9 14:59 aa.txt
-rw-r--r--. 1 root root 108 Aug  9 15:01 test.txt
[root@localhost mytest]#
[root@localhost mytest]# echo "write some" > t1.txt
[root@localhost mytest]# cat t1.txt
write some
[root@localhost mytest]# ls
aa.txt t1.txt test.txt
[root@localhost mytest]#
```

t1.txt文件不存在，新建文件后再输出数据到文件

(2) 重定向输出追加： >>

向文件输出内容，文件不存在则新建文件再输出；文件存在，则新的内容追加到文件的后面。

例如： `echo "hello new word" >> t1.txt`

```
[root@localhost mytest]# ls
aa.txt t1.txt test.txt
[root@localhost mytest]# cat t1.txt
write some
[root@localhost mytest]# echo "hello new word" >> t1.txt
[root@localhost mytest]#
[root@localhost mytest]# cat t1.txt
write some
hello new word
[root@localhost mytest]#
```

存在文件 t1.txt，在原来内容的后面加入新的内容 “hello new word”

(3) 管道 |

管道就是用 “|” 连接两个命令，以前面一个命令的输出作为后

面命令的输入，用于把管道左边的输出作为右边的输入。

例如：echo "hello linux" | grep "linux"

```
[root@localhost mytest]#  
[root@localhost mytest]# echo "hello linux" | grep "linux"  
hello linux  
[root@localhost mytest]#
```

对字符串搜索

查看某个进程是否存在

例如：ps -ef | grep "cat"

```
[root@localhost mytest]#  
[root@localhost mytest]# ps -ef | grep "java"  
root      4518   3451  0 22:16 pts/0    00:00:00 grep --color=auto java  
[root@localhost mytest]#  
[root@localhost mytest]# ps -ef | grep "cat"  
root      3214   2836  0 21:19 ?          00:00:00 /usr/bin/gnome-software --ga  
pplication-service  
root      4534   3451  0 22:16 pts/0    00:00:00 grep --color=auto cat  
[root@localhost mytest]#
```

9 vi 和 vim 编辑器

vi 是 linux 下标配的一个纯字符界面的文本编辑器，由于不是图形界面，相关的操作都要通过键盘输入命令来完成，需要记住一些常用的操作命令

vim 是 vi 的升级版，完全兼容 vi，vim 也可以完全当成 vi 使用，vim 是在 vi 的基础上增加一些功能，比如语法着色等

9-1 启动 vi 编辑器

语法：vi 文件名

说明：如果文件存在，则打开该文件，如果参文件不存在，则会新建该文件（不是立即创建，而是在保存的时候创建）

9-2 vi 常用操作

vi 操作分为**命令模式**和**编辑模式**

命令模式：按 Esc 键，进入命令模式，命令模式下无法编辑

编辑模式：按 a 或者 i 字母键，进入编辑模式（此时，底部会出现 insert）。在命令模式下按 :wq (冒号键 w 键 q 键)保存退出，按 :q! 不保存退出。

从命令模式进入编辑模式按 a 或者 i 字母键。

从编辑模式进入命令模式按 Esc 键。

编辑文件命令

- 1) dd：删除光标所在行
- 2) yy：复制光标所在行到缓冲区
- 3) p：粘贴缓冲区中的内容
- 4) gg：光标回到文件第一行
- 5) GG：光标回到文件最后一行
- 6) ^：光标移动至当前行的行首
- 7) \$：光标移动至当前行的行尾

- 8) /关键字: 按斜杠/键, 可以输入想搜索的字符, 然后确定进行搜索, 如果第一次查找的关键字不是想要的, 可以一直按 n 键往后查找 到想要的关键字为止

10 安装软件命令

(1) yum 命令介绍

yum: Yum(全称为 Yellow dog Updater, Modified): 是一个在 RedHat、Fedora 以及 CentOS 中的一种软件包管理器, 能够从指定的服务器自动下载软件包并且进行安装, 可以自动处理软件包之间的依赖性关系, 并且一次安装所有依赖的软件包, 无须繁琐地一次次下载、安装。

(2) yum 使用

查找软件包命令: yum search 安装包名称中的部分关键字

例如: yum search java

```
[root@localhost home]#  
[root@localhost home]# yum search java | grep JDK  
icedtea-web.x86_64 : Additional Java components for OpenJDK - Java browser  
java-1.6.0-openjdk.x86_64 : OpenJDK Runtime Environment  
java-1.6.0-openjdk-demo.x86_64 : OpenJDK Demos  
java-1.6.0-openjdk-devel.x86_64 : OpenJDK Development Environment  
java-1.6.0-openjdk-javadoc.x86_64 : OpenJDK API Documentation  
java-1.6.0-openjdk-src.x86_64 : OpenJDK Source Bundle  
java-1.7.0-openjdk.x86_64 : OpenJDK Runtime Environment
```

① 安装指定的软件命令: yum install 安装包名称

安装 openJdk , 开源的 java 虚拟机。

例如: yum install java-1.7.0-openjdk

② 删除软件包命令: yum remove 安装包名称

删除 openJdk

例如: yum remove java-1.7.0-openjdk

③ 列出所有已安装的软件包命令: yum list installed

```
[root@localhost home]#  
[root@localhost home]# yum list installed  
Loaded plugins: fastestmirror, langpacks  
Loading mirror speeds from cached hostfile  
* base: mirror.bit.edu.cn  
* extras: mirror.bit.edu.cn  
* updates: mirror.bit.edu.cn  
Installed Packages  
GConf2.x86_64 3.2.6-8.el7  
GeoIP.x86_64 1.5.0-11.el7  
ModemManager.x86_64 1.6.0-2.el7  
ModemManager-glib.x86_64 1.6.0-2.el7  
NetworkManager.x86_64 1:1.4.0-12.el7  
NetworkManager-ads1.x86_64 1:1.4.0-12.el7
```

服务器的镜像地址, 从这里查找需要的软件

④ 清除已安装软件包的下载文件命令: yum clean all (yum 命令下载的安装包都放在/var/cache/yum 目录)

11 快捷键

shift + insert 是粘贴

tab 是补全代码

y:同意

n:否定

访问历史命令, 方向键的上下箭头