

Linux基础命令

作者: 张大鹏

日期:2019年8月2日

抖音:lxgzhw

微博:理想国真惠玩

博客:<http://lxgzhw520.com>

GitHub:<https://github.com/lxgzhw520>

001.查看系统版本

```
cat /etc/centos-release
```

002.查看ip地址

```
ifconfig
```

003.终端分类

- 设备终端
- 物理终端
- 虚拟终端
- 图形终端
- 串行终端
- 伪终端

004.查看终端

```
tty
```

005.查看终端 用户名 登录时间

```
who am i
```

006.查看用户名

```
whoami
```

007.查看shell类型

```
echo $SHELL
```

008.查看所有的shell

```
cat /etc/shells
```

009.查看主机名

```
hostname
```

010.切换文件夹

```
cd
```

011.查看提示符格式

```
echo $PS1
```

- `\u` 用户名
- `\h` 主机名
- `\W` 目录名

012.修改提示符格式

```
PS1="\[\e[1;5;41;37m\][\u@\h \W]\\$\\[\e[0m\]"
```

可以将这一句加在 `/root/.bash_profile` 的最后一行,使其永久生效

- 案例:红色的提示符

```
PS1="\[\e[1;31m\][\u@\h \W]\$[\e[0m\]"
```

```
[root@localhost ~]#vim .bash_profile
```

013.判断命令为内部还是外部

type 命名名

如果告诉了命令的地址,则为外部命令

内部命令提示 `xxx is a shell builtin`

- 原理: 根据\$PATH变量来寻找的
- 内部命令
 - 直接在内存中
- 外部命令
 - 找到以后加载到内存中

014.查看外部命令的路径

whereis 外部命令名

案例: `whereis hostname`

015.取别名

alias 变量名="命令名"

案例: `alias cdnet="cd /ect/sysconfig/network-scripts/"`

unalias 取消别名

016.查看文件内容

cat

案例: `cat ~/.bashrc`

017.查看内存使用情况

free -h

018.查看内核版本

```
uname -r
```

019.查看系统时间

```
date
```

用硬件时间覆盖系统时间 `clock -s`

020.查看硬件时间

```
clock
```

同步为系统时间 `clock -w`

021.同步服务器时间

```
yum -y install ntpdate
```

 安装工具

```
ntpdate pool.ntp.org
```

 同步网络时间

```
clock -w
```

 将系统时间同步至硬件时间

022.关机命令

```
poweroff
```

023.重启命令

```
reboot
```

024.指定时间关机

```
shutdown -h +10
```

 十分钟以后关机

`shutdown -c` 取消关机

025.屏幕共享

`yum -y install screen` 安装命令

`rpm -ivh xxx.rpm` 安装rpm包的命令(参考)

步骤:

- 1.开启屏幕共享 `screen -s help`
- 2.用另一个终端连接到同一台服务器
- 3.查看有哪些屏幕共享 `screen -ls`
- 4.连接屏幕共享 `screen -x help`
- 5.退出 `exit`

026.打印命令

`echo`

发出声响 `echo -e "\a"`

027.创建文件

`touch`

- 案例: 创建一个空文件

`touch file.txt`

- 案例: 同时创建多个文件 笛卡尔乘积

`touch /data/file{1,2,3}.{txt,log}`

```
[root@localhost ~]#mkdir /data
[root@localhost ~]#touch /data/file{1,2,3}.{txt,log}
[root@localhost ~]#ls /data
file1.log file1.txt file2.log file2.txt file3.log file3.txt
[root@localhost ~]#
```

- 案例: 自动根据日期生成日志文件

```
[root@localhost ~]#touch /data/`date +%F`.log
[root@localhost ~]#ls /data
2019-08-02.log
[root@localhost ~]#
```

这里用到了命令嵌套,反向单引号里面的的是一个Linux命令,能够生成日期

反向单引号一般是命令调用命令的时候使用,和\$()是等价的

- 案例:生成范围的数字或字母

```
[root@localhost ~]# echo {1..10}
1 2 3 4 5 6 7 8 9 10
[root@localhost ~]# echo {1..10..2}
1 3 5 7 9
[root@localhost ~]# echo {a..Z}
a ` _ ^ ] [ Z
[root@localhost ~]# echo {a..z}
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

028.查看历史命令

history

```
[root@localhost ~]# history
 1  pwd
 2  vim .bash_profile
 3  yum -y install vim
 4  vim .bash_profile
 5  . .bash_profile
 6  screen -ls
 7  exit
 8  yum -y install screen
 9  lsblk
10  screen -s help
11  exit
12  screen -ls
13  ls
14  ifconfig
15  exit
16  screen -s help
17  ls
18  exit
19  screen -ls
20  screen
21  screen -ls
22  screen -x help
23  screen -ls
24  screen -x help
25  screen -x 7534
26  exit
```

- 命令存储在 `.bash_history` 文件中
- 执行命令历史中最近的以s开头的 `!s`
- 执行命令历史中包含某内容的命令 `!?包含的内容`
- `alt + .` 能够将上一个命令的参数作为此次命令的参数

029.查看日历

cal

cal 09 1999 查看1999年9月的日历

cal 2019 查看2019年的日历

cal 2019

2019

January							February							March						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4						1	2						1	2
6	7	8	9	10	11	12	3	4	5	6	7	8	9	3	4	5	6	7	8	9
13	14	15	16	17	18	19	10	11	12	13	14	15	16	10	11	12	13	14	15	16
20	21	22	23	24	25	26	17	18	19	20	21	22	23	17	18	19	20	21	22	23
27	28	29	30	31			24	25	26	27	28			24	25	26	27	28	29	30
														31						
April							May							June						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6				1	2	3	4							1
7	8	9	10	11	12	13	5	6	7	8	9	10	11	2	3	4	5	6	7	8
14	15	16	17	18	19	20	12	13	14	15	16	17	18	9	10	11	12	13	14	15
21	22	23	24	25	26	27	19	20	21	22	23	24	25	16	17	18	19	20	21	22
28	29	30					26	27	28	29	30	31		23	24	25	26	27	28	29
														30						
July							August							September						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6					1	2	3	1	2	3	4	5	6	7
7	8	9	10	11	12	13	4	5	6	7	8	9	10	8	9	10	11	12	13	14
14	15	16	17	18	19	20	11	12	13	14	15	16	17	15	16	17	18	19	20	21
21	22	23	24	25	26	27	18	19	20	21	22	23	24	22	23	24	25	26	27	28
28	29	30	31				25	26	27	28	29	30	31	29	30					

030.计算器命令

bc

bc

bc 1.06.95

Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation, Inc.

This is free software with ABSOLUTELY NO WARRANTY.

For details type `warranty'.

3+3

6

obase=8

33

41

obase=16

33

21

3*3

9

- obase=8 转换为8进制
- obase=16 转换为16进制

- `obase=2` 转换为2进制

031.查看帮助

`whatis` 命令名

查看一个命令的基本介绍

```
[root@localhost ~]# whatis rm
rm (1)          - remove files or directories
```

`type` 命令名

查看是内部命令还是外部命令

```
[root@localhost ~]# type rm
rm is aliased to `rm -i'
```

`man` 命令名

查看命令的详细介绍

```
RM(1)                                User Commands                                RM(1)

NAME
  rm - remove files or directories

SYNOPSIS
  rm [OPTION]... FILE...

DESCRIPTION
  This manual page documents the GNU version of rm.  rm removes each
  specified file.  By default, it does not remove directories.

  If the -I or --interactive=once option is given, and there are more
  than three files or the -r, -R, or --recursive are given, then rm
  prompts the user for whether to proceed with the entire operation.  If
  the response is not affirmative, the entire command is aborted.

  Otherwise, if a file is unwritable, standard input is a terminal, and
  the -f or --force option is not given, or the -i or --interac-
tive=always option is given, rm prompts the user for whether to remove
  the file.  If the response is not affirmative, the file is skipped.

OPTIONS
  Remove (unlink) the FILE(s).

  -f, --force
          ignore nonexistent files and arguments, never prompt
Manual page rm(1) line 1 (press h for help or q to quit)
```


032.查看物理路径

`pwd -P`

```
[root@localhost bin]#pwd -P
/usr/bin
[root@localhost bin]#pwd
/bin
[root@localhost bin]#
```

033.从路径中获取文件或文件夹名字

`basename` 获取文件名字

`dirname` 获取文件夹名字

```
[root@localhost ~]#basename /usr/bin/ls
ls
[root@localhost ~]#dirname /usr/bin/ls
/usr/bin
[root@localhost ~]#
```

034.切换文件夹

`cd` 回到家目录

`cd -` 回到上次的目录

`cd 文件夹路径` 切换到文件夹路径所在位置

035.列出目录内容

`ls -a` 包含隐藏文件

`ls -l` 显示额外的信息

`ls -R` 目录递归

036.查看文件状态

`stat`

```
[root@localhost ~]#stat /data/2019-08-02.log
  File: '/data/2019-08-02.log'
  Size: 0                Blocks: 0          IO Block: 4096   regular empty file
Device: fd00h/64768d    Inode: 1773        Links: 1
Access: (0644/-rw-r--r--)  Uid: (   0/   root)   Gid: (   0/   root)
Context: unconfined_u:object_r:default_t:s0
Access: 2019-08-02 08:44:56.365852457 +0800
Modify: 2019-08-02 08:44:56.365852457 +0800
Change: 2019-08-02 08:44:56.365852457 +0800
 Birth: -
```

- 访问时间
- 修改时间:文件内容发生改变
- 改变时间:文件所有者等非内容信息发生改变

037.复制文件和目录

cp

- 案例:复制文件

```
[root@localhost ~]#ls /data/
2019-08-02.log
[root@localhost ~]#cp /data/2019-08-02.log /data/test.log
[root@localhost ~]#ls /data/
2019-08-02.log test.log
[root@localhost ~]#
```

- 案例:复制文件夹

```
[root@localhost ~]#mkdir /data/dir01
[root@localhost ~]#ls /data/
2019-08-02.log dir01 test.log
[root@localhost ~]#cp -r /data/dir01/ /data/dir02
[root@localhost ~]#ls /data/
2019-08-02.log dir01 dir02 test.log
[root@localhost ~]#
```

038.移动文件夹

mv

- 案例:移动文件夹

```
[root@localhost ~]#ls /data/
2019-08-02.log dir01 dir02 test.log
[root@localhost ~]#mv /data/test.log /data/dir01/
[root@localhost ~]#ls /data/dir01/
test.log
[root@localhost ~]#
```

- 案例:重命名文件夹

```
[root@localhost ~]# mv /data/dir01/ /data/dir
[root@localhost ~]# ls /data/
2019-08-02.log  dir  dir02
[root@localhost ~]#
```

039.删除文件夹

rm

- 案例:删除文件

```
[root@localhost ~]# ls /data/
2019-08-02.log  dir  dir02
[root@localhost ~]# rm /data/*.log
rm: remove regular empty file '/data/2019-08-02.log'? y
[root@localhost ~]# ls /data/
dir  dir02
[root@localhost ~]#
```

- 案例:删除文件夹

```
[root@localhost ~]# rm -rf /data/dir*
[root@localhost ~]# ls /data/
[root@localhost ~]#
```

040.创建目录

mkdir

- 案例:创建一个目录

```
[root@localhost ~]# ls /data/
[root@localhost ~]# mkdir /data/dir01
[root@localhost ~]# ls /data/
dir01
```

- 案例:创建多个目录

```
[root@localhost ~]# mkdir /data/dir02 /data/dir03
[root@localhost ~]# ls /data/
dir01  dir02  dir03
```

- 案例:递归创建多级目录

```
[root@localhost ~]#mkdir -pv /data/dir04/t/t/t/
mkdir: created directory '/data/dir04'
mkdir: created directory '/data/dir04/t'
mkdir: created directory '/data/dir04/t/t'
mkdir: created directory '/data/dir04/t/t/t/'
[root@localhost ~]#tree /data/
```

```
/data/
├── dir01
├── dir02
├── dir03
└── dir04
    ├── t
    │   └── t
    │       └── t
```

```
7 directories, 0 files
```

041.查看目录树

`yum -y install tree` 安装命令

`tree /data/` 执行命令

- 案例:查看/data下面的目录树

```
[root@localhost ~]#tree /data/
```

```
/data/
├── dir01
├── dir02
└── dir03
```

```
3 directories, 0 files
```

042.删除空目录

`rmdir`

- 案例:递归删除/data/dir04下的空目录(包含dir04)

```
[root@localhost ~]#rmdir -pv /data/dir04/t/t/t/
rmdir: removing directory, '/data/dir04/t/t/t/'
rmdir: removing directory, '/data/dir04/t/t'
rmdir: removing directory, '/data/dir04/t'
rmdir: removing directory, '/data/dir04'
rmdir: removing directory, '/data'
rmdir: failed to remove directory '/data': Directory not empty
```

```
[root@localhost ~]#tree /data/
```

```
/data/
├── dir01
├── dir02
└── dir03
```

```
3 directories, 0 files
```

043.查看文件的类型

file

- 案例:查看文件的类型

```
[root@localhost ~]#file anaconda-ks.cfg
anaconda-ks.cfg: ASCII text
```

- 案例:查看文件夹的类型

```
[root@localhost ~]#file /data/dir01
/data/dir01: directory
```

044.重定向

> 写入 w模式

>> 追加 a模式

- 案例:将输出重定向到文件

```
[root@localhost ~]#ls
anaconda-ks.cfg
[root@localhost ~]#ls > /data/ls.out
[root@localhost ~]#cat /data/ls.out
anaconda-ks.cfg
```

- 案例:将输出重定向到另一个终端

```
[root@localhost ~]#ls > /dev/pts/1
```

set -C 禁止覆盖

set +C 允许覆盖

- 案例:标准错误的重定向

```
[root@localhost ~]#> /data/ls.out
[root@localhost ~]#cat /data/ls.out
[root@localhost ~]#ls >> /data/ls.out
[root@localhost ~]#ccc 2 >> /data/ls.out
-bash: ccc: command not found
[root@localhost ~]#ccc 2>> /data/ls.out
[root@localhost ~]#cat /data/ls.out
anaconda-ks.cfg
-bash: ccc: command not found
[root@localhost ~]#
```

- 案例:快速生成一份月历

```
[root@localhost ~]#cal 2019 > /data/cal.txt
[root@localhost ~]#cat /data/cal.txt
2019

    January                February                March
Su Mo Tu We Th Fr Sa    Su Mo Tu We Th Fr Sa    Su Mo Tu We Th Fr Sa
    1  2  3  4  5          1  2                      1  2
  6  7  8  9 10 11 12      3  4  5  6  7  8  9          3  4  5  6  7  8  9
13 14 15 16 17 18 19      10 11 12 13 14 15 16         10 11 12 13 14 15 16
20 21 22 23 24 25 26      17 18 19 20 21 22 23         17 18 19 20 21 22 23
27 28 29 30 31           24 25 26 27 28                24 25 26 27 28 29 30
                                           31

    April                  May                    June
Su Mo Tu We Th Fr Sa    Su Mo Tu We Th Fr Sa    Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6          1  2  3  4                1
  7  8  9 10 11 12 13      5  6  7  8  9 10 11          2  3  4  5  6  7  8
14 15 16 17 18 19 20      12 13 14 15 16 17 18         9 10 11 12 13 14 15
21 22 23 24 25 26 27      19 20 21 22 23 24 25         16 17 18 19 20 21 22
28 29 30                26 27 28 29 30 31          23 24 25 26 27 28 29
                                           30

    July                   August                September
Su Mo Tu We Th Fr Sa    Su Mo Tu We Th Fr Sa    Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6          1  2  3                1  2  3  4  5  6  7
  7  8  9 10 11 12 13      4  5  6  7  8  9 10          8  9 10 11 12 13 14
14 15 16 17 18 19 20      11 12 13 14 15 16 17         15 16 17 18 19 20 21
21 22 23 24 25 26 27      18 19 20 21 22 23 24         22 23 24 25 26 27 28
28 29 30 31           25 26 27 28 29 30 31          29 30
```

- 案例:标准输入的重定向

< 将文件内容作为左边的输入内容

```
[root@localhost data]#tr '\n' '\t' < f1
a      b      c      [root@localhost data]#
```

- 案例:将小写转换为大写

```
[root@localhost data]#tr 'a-z' 'A-Z' < f1
A
B
C

[root@localhost data]#cat f1
a
b
c
```

- 案例: 删除指定的内容 -d

```
[root@localhost data]#tr -d '\n' < f1
abc[root@localhost data]#
```

045.管道

|

将前一个命令的执行结果作为后一个命令的参数

- 案例:将ls的结果全部转换为大写

```
[root@localhost data]#ls | tr 'a-z' 'A-Z'
CAL.TXT
F1
F2
LS.OUT
TR
```

046.搜索内容

grep

- 案例:搜索inet

```
[root@localhost data]#ifconfig|grep inet
inet 172.16.116.162 netmask 255.255.255.0 broadcast 172.16.116.255
inet6 fe80::db8d:56eb:3d63:c578 prefixlen 64 scopeid 0x20<link>
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
```

047.重定向且继续显示

tee

- 案例

```
[root@localhost data]#ls | tee ls.out
cal.txt
f1
f2
ls.out
tr
```