

从所有教程的词条中查询...

全部开发者教程

- 1 Linux虚拟机安装配置
- 2 Linux基础命令的使用【选修】
- 3 Linux极速上手
- 4 Linux试炼之配置与shell实战
- 5 Linux总结与走进大数据
- 6 作业
- 第2周-大数据起源之初识Hadoop
- 1 初始Hadoop
- 2 Hadoop的安装方式
- 3 作业
- 第3周-Hadoop之HDFS的使用

Java / 2 Linux基础命令的使用【选修】

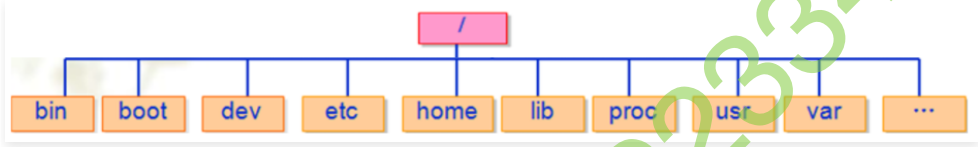


徐老师 · 更新于 2020-08-22

上一节 1 Linux虚拟机安... 3 Linux极速上手 下一节

Linux基本命令的使用

Linux文件系统 和 Windows文件系统有一些区别，Windows是分C盘、D盘、E盘...的，但是在Linux中是有一个最大的目录，称之为根目录，用 / 表示，根目录下面会有很多子目录，这些子目录其实可以理解为 windows中C盘、D盘、E盘。
不同的子目录具有不同的含义，存放不同类型的文件信息。



- bin 存放二进制可执行文件(ls,cat,mkdir等)
- boot 存放用于系统引导时使用的各种文件
- dev 存放设备文件
- etc 存放系统配置文件
- home 存放所有用户文件的根目录
- lib 存放跟文件系统中的程序运行所需要的共享库及内核模块
- proc 虚拟文件系统，存放当前内存的映射
- usr 存放系统应用程序，比较重要的目录/usr/local 管理员软件安装目录
- var 存放运行时需要改变数据的文件
- mnt 挂载目录
- sbin 存储管理级别的二进制执行文件
- root 超级用户主目录
- opt 额外安装的可选应用程序包安装位置

下面开始正式分析Linux中的基本命令。

文件查看

- pwd 列出当前目录的路径
- 查看当前所在的路径
- 注意，~表示是用户家目录，目前我们登陆的是root用户，root用户的家目录是/root

<> 代码块

```
1 [root@localhost ~]# pwd
2 /root
```

ls 列出当前目录下的所有文件

<> 代码块

```
1 [root@localhost ~]# ls
2 anaconda-ks.cfg
```

ll(ls -l缩写) 列出当前目录下的文件（带文件信息）

<> 代码块

```
1 [root@localhost ~]# ll
2 total 4
3 -rw-----. 1 root root 1243 Mar 28 20:59 anaconda-ks.cfg
4 [root@localhost ~]# ls -l
5 total 4
6 -rw-----. 1 root root 1243 Mar 28 20:59 anaconda-ks.cfg
```

ll -a 列出当前目录下的所有文件（包括隐藏文件）

<> 代码块

```
1 [root@localhost ~]# ll -a
2 total 28
3 dr-xr-x---. 2 root root 135 Mar 28 21:00 .
4 dr-xr-xr-x. 17 root root 224 Mar 28 20:58 ..
5 -rw-----. 1 root root 1243 Mar 28 20:59 anaconda-ks.cfg
6 -rw-----. 1 root root 8 Mar 28 21:00 .bash_history
7 -rw-r--r--. 1 root root 18 Dec 29 2013 .bash_logout
8 -rw-r--r--. 1 root root 176 Dec 29 2013 .bash_profile
9 -rw-r--r--. 1 root root 176 Dec 29 2013 .bashrc
10 -rw-r--r--. 1 root root 100 Dec 29 2013 .cshrc
11 -rw-r--r--. 1 root root 129 Dec 29 2013 .tcshrc
```

ll --help 查看ls用法, -help是一个帮助命令

<> 代码块

```
1 [root@localhost ~]# ll --help
2 Usage: ls [OPTION]... [FILE]...
3 List information about the FILES (the current directory by default).
4 Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
5
6 Mandatory arguments to long options are mandatory for short options too.
7 -a, --all do not ignore entries starting with .
8 .....
```

创建、重命名文件\文件夹

touch filename 创建空文件

创建空文件hello.txt

<> 代码块

```
1 [root@localhost ~]# touch hello.txt
2 [root@localhost ~]# ll
3 total 4
4 -rw-----. 1 root root 1243 Mar 28 20:59 anaconda-ks.cfg
5 -rw-r--r--. 1 root root 0 Mar 28 22:21 hello.txt
```

mkdir 创建目录

创建目录abc

<> 代码块

```
1 [root@localhost ~]# mkdir abc
2 [root@localhost ~]# ll
3 total 4
4 drwxr-xr-x. 2 root root 6 Mar 28 22:22 abc
5 -rw-----. 1 root root 1243 Mar 28 20:59 anaconda-ks.cfg
6 -rw-r--r--. 1 root root 0 Mar 28 22:21 hello.txt
```

[意见反馈](#)[收藏教程](#)[标记书签](#)

mkdir -p 目标目录存在也不报错

当我们在创建目录时不确定这个目录是否已存在的时候，可以使用-p参数，就算目录已存在也不会报错，如果不指定-p参数会报错，会提示目录已存在

创建目录abc

<> 代码块

```
1 [root@localhost ~]# mkdir -p abc
2 [root@localhost ~]# ll
3 total 4
4 drwxr-xr-x. 2 root root 6 Mar 28 22:22 abc
5 -rw-----. 1 root root 1243 Mar 28 20:59 anaconda-ks.cfg
6 -rw-r--r--. 1 root root 0 Mar 28 22:21 hello.txt
7 [root@localhost ~]# mkdir abc
8 mkdir: cannot create directory 'abc': File exists
```

mv 重命名文件\文件夹

修改目录abc的名称为abx

<> 代码块

```
1 [root@localhost ~]# ll
2 total 4
3 drwxr-xr-x. 2 root root 6 Mar 28 22:22 abc
4 -rw-----. 1 root root 1243 Mar 28 20:59 anaconda-ks.cfg
5 -rw-r--r--. 1 root root 0 Mar 28 22:21 hello.txt
6 [root@localhost ~]# mv abc abx
7 [root@localhost ~]# ll
8 total 4
9 drwxr-xr-x. 2 root root 6 Mar 28 22:22 abx
10 -rw-----. 1 root root 1243 Mar 28 20:59 anaconda-ks.cfg
11 -rw-r--r--. 1 root root 0 Mar 28 22:21 hello.txt
```

链接文件

linux有两种链接：硬链接、符号(软)链接

软链接功能类似类似于windows的快捷方式，主要用于节省磁盘空间

首先看硬链接：硬链接相当于对原始文件的一个复制，不能对目录使用硬链接。

命令如下：

ln hello.txt hlink

<> 代码块

```
1 [root@localhost ~]# ll
2 total 4
3 drwxr-xr-x. 2 root root 6 Mar 28 22:22 abx
4 -rw-----. 1 root root 1243 Mar 28 20:59 anaconda-ks.cfg
5 -rw-r--r--. 1 root root 0 Mar 28 22:21 hello.txt
6 [root@localhost ~]# ln hello.txt hlink
7 [root@localhost ~]# ll
8 total 4
9 drwxr-xr-x. 2 root root 6 Mar 28 22:22 abx
10 -rw-----. 1 root root 1243 Mar 28 20:59 anaconda-ks.cfg
11 -rw-r--r--. 2 root root 0 Mar 28 22:21 hello.txt
12 -rw-r--r--. 2 root root 0 Mar 28 22:21 hlink
```

再看符号(软)链接：

如果想使用软链接，需要添加-s，相当于快捷方式，不能删除原文件

命令如下：

ln -s hello.txt vlink

<> 代码块

```
1 [root@localhost ~]# ll
2 total 4
3 drwxr-xr-x. 2 root root    6 Mar 28 22:22 abx
4 -rw-----. 1 root root 1243 Mar 28 20:59 anaconda-ks.cfg
5 -rw-r--r--. 2 root root    0 Mar 28 22:21 hello.txt
6 -rw-r--r--. 2 root root    0 Mar 28 22:21 hlink
7 [root@localhost ~]# ln -s hello.txt vlink
8 [root@localhost ~]# ll
9 total 4
10 drwxr-xr-x. 2 root root    6 Mar 28 22:22 abx
11 -rw-----. 1 root root 1243 Mar 28 20:59 anaconda-ks.cfg
12 -rw-r--r--. 2 root root    0 Mar 28 22:21 hello.txt
13 -rw-r--r--. 2 root root    0 Mar 28 22:21 hlink
14 lrwxrwxrwx. 1 root root    9 Mar 28 22:33 vlink -> hello.txt
```

切换目录

cd . 当前目录

一个.表示当前目录

<> 代码块

```
1 [root@localhost ~]# pwd
2 /root
3 [root@localhost ~]# cd .
4 [root@localhost ~]# pwd
5 /root
```

cd ... 去上一级目录

两个...表示上一级目录

<> 代码块

```
1 [root@localhost ~]# pwd
2 /root
3 [root@localhost ~]# cd ..
4 [root@localhost /]# pwd
5 /
```

cd / 去根目录

通过在cd后面指定目录，可以切换到指定目录

<> 代码块

```
1 [root@localhost /]# cd /
2 [root@localhost /]# pwd
3 /
4 [root@localhost ~]# cd /bin/
5 [root@localhost bin]# pwd
6 /bin
```

cd ~ 去当前用户主(家)目录

<> 代码块

```
1 [root@localhost bin]# cd ~
2 [root@localhost ~]# pwd
3 /root
```

cd xxx/xxx 直接跳转到某个目录

[意见反馈](#)[收藏教程](#)[标记书签](#)

```
1 [root@localhost ~]# cd abx/test/  
2 [root@localhost test]# pwd  
3 /root/abx/test
```

删除文件\文件夹(目录)

rm命令可以删除文件或者目录，也可以将某个目录及其下属的所有文件及其子目录均删除掉
对于链接文件，只是删除整个链接文件，而原有文件保持不变。

常见用法：

rm 删除文件

删除文件，但是会有提示确认对话，输入y确认删除！

<> 代码块

```
1 [root@localhost test]# touch abc.txt  
2 [root@localhost test]# ll  
3 total 0  
4 -rw-r--r--. 1 root root 0 Mar 29 13:53 abc.txt  
5 [root@localhost test]# rm abc.txt  
6 rm: remove regular empty file 'abc.txt'? y  
7 [root@localhost test]# ll  
8 total 0
```

rm -r 删除目录，需要确认

删除目录需要指定r参数，否则会提示不能删除

r是给rm加入递归(recursion)特性，也就是目标为文件夹时删除文件夹下所有数据
使用rm -r 在删除目录的时候也会有提示确认对话，输入y确认删除

<> 代码块

```
1 [root@localhost abx]# ll  
2 total 0  
3 drwxr-xr-x. 2 root root 6 Mar 29 14:01 test  
4 [root@localhost abx]# rm test  
5 rm: cannot remove 'test': Is a directory  
6 [root@localhost abx]# rm -r test  
7 rm: remove directory 'test'? y  
8 [root@localhost abx]# ll  
9 total 0
```

rm -f 强制删除

给rm加入强制(force)特性，也就是遇到删除时不需要询问即可直接删除

注意：这个操作还是比较危险的，建议慎用，因为删除之后就找不到了

Linux系统中没有回收站

<> 代码块

```
1 [root@localhost abx]# touch a.txt  
2 [root@localhost abx]# ll  
3 total 0  
4 -rw-r--r--. 1 root root 0 Mar 29 14:03 a.txt  
5 [root@localhost abx]# rm -f a.txt  
6 [root@localhost abx]# ll  
7 total 0
```

rm -rf 递归删除目录及其文件

Linux中最危险的操作，最具破坏性

rf参数可以强制递归删除任何数据，并且没有任何提示，慎用！慎用！慎用！

<> 代码块

```
1 [root@localhost ~]# ll
2 drwxr-xr-x. 2 root root    6 Mar 29 14:03 abx
3 -rw-----. 1 root root 1243 Mar 28 20:59 anaconda-ks.cfg
4 [root@localhost ~]# mkdir -p abx/test/aaa
5 [root@localhost ~]# cd abx/test/aaa/
6 [root@localhost aaa]# touch a.txt
7 [root@localhost aaa]# cd ~
8 [root@localhost ~]# rm -rf abx
9 [root@localhost ~]# ll
10 -rw-----. 1 root root 1243 Mar 28 20:59 anaconda-ks.cfg
```

复制\粘贴\剪切

常见用法:

cp 复制&粘贴文件

复制hello.txt文件, 复制后的文件名为hello-bak.txt

<> 代码块

```
1 [root@localhost ~]# ll
2 -rw-r--r--. 1 root root    0 Mar 28 22:21 hello.txt
3 [root@localhost ~]# cp hello.txt hello-bak.txt
4 [root@localhost ~]# ll
5 -rw-r--r--. 1 root root    0 Mar 29 14:20 hello-bak.txt
6 -rw-r--r--. 1 root root    0 Mar 28 22:21 hello.txt
```

cp -r 复制&粘贴文件或目录

复制目录, 需要指定r参数

<> 代码块

```
1 [root@localhost ~]# mkdir abc
2 [root@localhost ~]# ll
3 drwxr-xr-x. 2 root root    6 Mar 29 14:21 abc
4 [root@localhost ~]# cp abc xyz 【错误用法, 复制目录必须指定-r参数】
5 cp: omitting directory 'abc'
6 [root@localhost ~]# cp -r abc xyz
7 [root@localhost ~]# ll
8 drwxr-xr-x. 2 root root    6 Mar 29 14:21 abc
9 drwxr-xr-x. 2 root root    6 Mar 29 14:21 xyz
```

mv 移动 (剪切) 文件或目录

将目录xyz移动到目录abc下面

<> 代码块

```
1 [root@localhost ~]# ll
2 drwxr-xr-x. 2 root root    6 Mar 29 14:21 abc
3 drwxr-xr-x. 2 root root    6 Mar 29 14:21 xyz
4 [root@localhost ~]# ll abc/
5 total 0
6 [root@localhost ~]# mv xyz abc
7 [root@localhost ~]# ll abc/
8 drwxr-xr-x. 2 root root    6 Mar 29 14:21 xyz
```

远程复制

scp命令用于在网络中不同主机之间复制文件或目录。scp是有Security的文件copy, 基于ssh登录, 如果没有配置密钥和授权, 需要输入主机密码

[意见反馈](#)[收藏教程](#)[标记书签](#)

从本地复制 (192.168.182.131) 到远程主机 (192.168.182.130)
scp /root/hello.txt 192.168.182.130:/root

<> 代码块

```
1 [root@localhost ~]# scp /root/hello.txt 192.168.182.130:/root/
2 The authenticity of host '192.168.182.130 (192.168.182.130)' can't be established
3 ECDSA key fingerprint is SHA256:uUG2QrWRlZxcwfv6GUot9DV59c+iFugZ7FhR89m2S00.
4 ECDSA key fingerprint is MD5:82:9d:01:51:06:a7:14:24:a9:16:3d:a1:5e:6d:0d:16.
5 Are you sure you want to continue connecting (yes/no)? yes 【第一次会提示此信息】
6 Warning: Permanently added '192.168.182.130' (ECDSA) to the list of known hosts
7 root@192.168.182.130's password: 【在这里输入192.168.182.130机器的密码即可】
8 hello.txt                                100% 0 0.0KB/s 00:00
```

登录到192.168.182.130机器，确认文件是否成功复制过去

<> 代码块

```
1 [root@localhost ~]# cd /root/
2 [root@localhost ~]# ll
3 total 4
4 -rw-----. 1 root root 1243 Mar 28 20:59 anaconda-ks.cfg
5 -rw-r--r--. 1 root root 0 Mar 29 14:30 hello.txt
6 [root@localhost ~]#
```

提示：
显示进度在scp后添加-v
复制目录在scp后添加-r
静默复制模式在scp后添加-q
scp -rq /root/abc 192.168.182.130:/root

<> 代码块

```
1 [root@localhost ~]# scp -rq /root/abc/ 192.168.182.130:/root/
2 root@192.168.182.130's password: 【在这里输入192.168.182.130机器的密码即可】
```

登录到192.168.182.130机器，确认目录是否成功复制过去

<> 代码块

```
1 [root@localhost ~]# ll
2 total 4
3 drwxr-xr-x. 3 root root 17 Mar 29 14:33 abc
4 -rw-----. 1 root root 1243 Mar 28 20:59 anaconda-ks.cfg
5 -rw-r--r--. 1 root root 0 Mar 29 14:30 hello.txt
```

文件属性

[root@localhost ~]# ll

total 4						
drwxr-xr-x.	3	root	root	17	Mar 29 14:24	abc
-rw-----.	1	root	root	1243	Mar 28 20:59	anaconda-ks.cfg
-rw-r--r--.	1	root	root	0	Mar 29 14:20	hello-bak.txt
-rw-r--r--.	1	root	root	0	Mar 28 22:21	hello.txt
第一段	第二段	第三段	第四段	第五段	第六段	第七段

黑带七段
第一段：权限

其余字符每3个一组 (rwx)，读 (r)、写 (w)、执行 (x)

第一组：文件所有者的权限是读、写和执行

第二组：与文件所有者同一组的用户的权限

第三组：不与文件所有者同组的其他用户的权限

也可用数字表示为：r=4, w=2, x=1，如：权限6可以表示为r+w=6

第二段：目录/链接个数

对于目录文件，表示它的第一级子目录的个数

注意：此处的值要减2才等于该目录下的子目录的实际个数(目录下默认包含.和...这两个目录)

对于其他文件，默认是1

第三段：所属用户

第四段：所属组

第五段：文件大小 (字节)

第六段：最后修改时间

第七段：文件\文件夹名称

chmod 分配权限

常见用法：

chmod u+x xxx.txt 给当前所有者添加执行权限【x表示是执行权限】

针对hello.txt文件，给当前用户添加执行权限

<> 代码块

```
1 [root@localhost ~]# ll
2 -rw-r--r--. 1 root root  0 Mar 28 22:21 hello.txt
3 [root@localhost ~]# chmod u+x hello.txt
4 [root@localhost ~]# ll
5 -rwxr--r--. 1 root root  0 Mar 28 22:21 hello.txt
```

chmod 777 xxx.txt 添加rwxrwxrwx权限 ◆

给hello.txt添加777权限

<> 代码块

```
1 [root@localhost ~]# chmod 777 hello.txt
2 [root@localhost ~]# ll
3 -rwxrwxrwx. 1 root root  0 Mar 28 22:21 hello.txt
```

chmod -R 777 xxx 给指定目录递归添加rwxrwxrwx权限

给abc目录及其子目录中的所有内容添加777权限

<> 代码块

```
1 [root@localhost ~]# ll
2 drwxr-xr-x. 3 root root 17 Mar 29 14:24 abc
3 [root@localhost ~]# cd abc/
4 [root@localhost abc]# ll
5 total 0
6 drwxr-xr-x. 2 root root 6 Mar 29 14:21 xyz
7 [root@localhost abc]# cd ..
8 [root@localhost ~]# chmod -R 777 abc
9 [root@localhost ~]# ll
10 drwxrwxrwx. 3 root root 17 Mar 29 14:24 abc
11 [root@localhost ~]# cd abc/
12 [root@localhost abc]# ll
13 total 0
14 drwxrwxrwx. 2 root root 6 Mar 29 14:21 xyz
```

内容查看

意见反馈

收藏教程

标记书签

cat 显示文本内容, 类似windows中的type(顺序输出)

<> 代码块

```
1 [root@localhost ~]# cat anaconda-ks.cfg
2 #version=DEVEL
3 #System authorization information
4 auth --enablesshadow --passalgo=sha512
5 .....
```

cat -b 显示行号输出

<> 代码块

```
1 [root@localhost ~]# cat -b anaconda-ks.cfg
2      1 #version=DEVEL
3      2 # System authorization information
4      3 auth --enablesshadow --passalgo=sha512
5      4 # Use CDROM installation media
6      .....
```

分屏显示 more

用一次显示一屏, 没有显示完时最后一行显示进度。回车显示下一行, 按b显示上一页, 空格显示下一页, q退出。

根据当前屏幕大小显示一屏内容

<> 代码块

```
1 [root@localhost ~]# more anaconda-ks.cfg
2 #version=DEVEL
3 #System authorization information
4 auth --enablesshadow --passalgo=sha512
5 #Use CDROM installation media
6 cdrom
7 #Use graphical install
8 graphical
9 #Run the Setup Agent on first boot
10 firstboot --enable
11 ignoredisk --only-use=sda
12 #Keyboard layouts
13 keyboard --vckeymap=us --xlayouts='us'
14 #System language
15 lang en_US.UTF-8
16
17 #Network information
18 network --bootproto=dhcp --device=ens33 --ipv6=auto --activate
19 network --hostname=localhost.localdomain
20
21 #Root password
22 rootpw --iscrypted $6$5x2f0WF8AMiHgQTV$ExkpEX6Sq1EfZVHaP4RxfY1s3B0o
23 dX2ouFfaTX2S0TttzWz7tX3L3cWRFeb1M4qfGUA2FGzrkyhlGfp4psze.
24 --More-- (48%)
```

压缩、解压

常见用法: 压缩和解压

参数:

- z 是否同时具有 gzip 的属性? 亦即是否需要用 gzip 压缩?
- c 创建一个压缩文件的参数指令(create 的意思);
- x 解开一个压缩文件的参数指令!

... 压缩的文档由二进制文件...

意见反馈

收藏教程

标记书签

-f 使用档案名字，这个参数是最后一个参数，后面只能接档案名！

注意：特别注意，在参数的下达中，c/x 仅能存在一个！不可同时存在！

压缩：

tar -zcvf 打包及压缩（gzip方式）

将abc目录的内容打包压缩为abc.tar.gz

<> 代码块

```
1 [root@localhost ~]# ll
2 drwxrwxrwx. 3 root root 17 Mar 29 14:24 abc
3 [root@localhost ~]# tar -zcvf abc.tar.gz abc
4 abc/
5 abc/xyz/
6 [root@localhost ~]# ll
7 drwxrwxrwx. 3 root root 17 Mar 29 14:24 abc
8 -rw-r--r--. 1 root root 130 Mar 29 15:24 abc.tar.gz
```

解压：

tar -zxvf 解压（gzip包）

先把abc.tar.gz移动到test目录下，再解压缩

<> 代码块

```
1 [root@localhost ~]# ll
2 drwxrwxrwx. 3 root root 17 Mar 29 14:24 abc
3 -rw-r--r--. 1 root root 130 Mar 29 15:24 abc.tar.gz
4 [root@localhost ~]# mkdir test
5 [root@localhost ~]# cd test/
6 [root@localhost test]# mv ../abc.tar.gz .
7 [root@localhost test]# ll
8 total 4
9 -rw-r--r--. 1 root root 130 Mar 29 15:24 abc.tar.gz
10 [root@localhost test]# tar -zxvf abc.tar.gz
11 abc/
12 abc/xyz/
13 [root@localhost test]# ll
14 total 4
15 drwxrwxrwx. 3 root root 17 Mar 29 14:24 abc
16 -rw-r--r--. 1 root root 130 Mar 29 15:24 abc.tar.gz
```

输出及显示

echo：将内容输出到设备，类似java里面的system.out.println()

常见用法：

echo "hello\t\t world！" 不解析转义字符

echo -e "hello\t\t world！" 解析转义字符

echo \$PATH 输出环境变量

注意：在打印变量信息的时候，使用echo \${PATH} 也可以，效果是一样的

<> 代码块

```
1 [root@localhost ~]# echo "hello\t\t world！"
2 hello\t\t world!
3 [root@localhost ~]# echo -e "hello\t\t world！"
4 hello      world!
5 [root@localhost ~]# echo $PATH
6 /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
7 [root@localhost ~]# echo ${PATH}
8 /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
```

软件安装和卸载

第一种：压缩包安装方式，直接解压，配置相应环境变量即可使用

第二种：在线安装，使用yum

yum集成了连接网络，软件安装，删除，更新等功能，yum在配置好repo后，机器只要连网，就能智能化安装软件，使用yum安装的好处在于可以自动安装软件需要的依赖包

安装

yum install -y 安装

升级

yum update 不跟则更新全部

查找和显示

yum info 显示包信息

yum list 不跟则显示已安装或可安装包

删除程序

yum remove

清除缓存

yum clean all 清除所有缓存（包含文件、旧软件）

在这里大家先了解yum的基本语法，等后面具体使用到的时候我们再演示。

查看操作历史

history保留了最近执行的命令记录，默认可以保留1000。

历史清单从0开始编号到最大值。

常见用法：

history N 显示最近N条命令

history -c 清除所有的历史记录

history -w xxx.txt 保存历史记录到文本xxx.txt

<> 代码块

```
1 [root@localhost ~]# history 10
2 142 ll
3 143 tar -zxvf abc.tar.gz
4 144 ll
5 145 cd .
6 146 ll
7 147 echo "hello\t\t world! "
8 148 echo -e "hello\t\t world! "
9 149 env
10 150 echo $PATH
11 151 history 10
12 [root@localhost ~]# history -w his.txt
13 [root@localhost ~]# ll
14 -rw-----. 1 root root 1395 Mar 29 15:41 his.txt
15 [root@localhost ~]# more his.txt
16 ip addr
17 ls /
18 pwd
19 ls
20 ll
21 .....
```

磁盘使用情况

使用df命令查看硬盘使用情况

查看磁盘使用情况

<> 代码块

```
1 Filesystem      Size  Used Avail Use% Mounted on
2 devtmpfs        898M   0  898M   0% /dev
3 tmpfs           910M   0  910M   0% /dev/shm
4 tmpfs           910M  9.5M  901M   2% /run
5 tmpfs           910M   0  910M   0% /sys/fs/cgroup
6 /dev/mapper/centos-root 17G  1.2G   16G   7% /
7 /dev/sda1       1014M 150M  865M  15% /boot
8 tmpfs           182M   0  182M   0% /run/user/0
```

清屏小命令

在控制台操作一会以后，屏幕上就显示了一屏内容，看起来乱成一团，可以使用clear命令清屏，类似于windows中的cls命令。

<> 代码块

```
1 [root@localhost ~]# clear
```

查看内存使用情况

free 查看内存和交换空间的使用情况

常见用法：

free -m：显示内存单位为MB

free -h：根据值的大小，显示易于识别的单位

<> 代码块

```
1 [root@localhost ~]# free -m
2              total        used         free       shared  buff/cache   availab
3 Mem:         1819         169        1509           9         140         15
4 Swap:         2047           0        2047
5 [root@localhost ~]# free -h
6              total        used         free       shared  buff/cache   availab
7 Mem:          1.8G         169M        1.5G         9.5M        140M        1.
8 Swap:          2.0G           0B         2.0G
```

关机重启快捷命令

shutdown -h now 关机

reboot -h now 重启

exit 退出当前登录状态

[1 Linux虚拟机安装配置](#) < 上一节下一节 > [3 Linux极速上手](#)[我要提出意见反馈](#)