# Day01

## 01/ 大数据课程介绍

什么是大数据？

海量数据的处理

大数据用在哪？

一句话：用在需要对海量数据进行处理的任何场合

举个栗子：

1、搜索引擎——需要对海量文档建索引库

2、各种电商——需要对用户的点击日志（海量）进行各种分析

3、电信运营商——需要对用户的上网行为（海量）进行分析

4、广告系统——需要对用户进行画像（海量）以便于做精准投放

……

大数据学什么？

1、学分布式系统的思想

2、学框架

（hadoop<基础，离线处理> + storm <实时处理的王者> + spark<加分项，技术趋势，擅长内存计算，迭代运算（机器学习）>）

以什么心态学习大数据？

1、平和

2、要有长远目光

以什么方法学习大数据？

1、要有主次之分(重点内容，深入掌握！)

2、一定要多动手实践（说三遍！！！）

3、学习时主要关注点：

a、最基本——知道这个框架是用来干什么的

b、应用——掌握框架的用法（编程规范、api、框架之间的衔接整合）

c、核心框架要掌握一定深度的原理

d、数据分析系统整体性的把握

## 02/ 基础课程内容介绍

Linux & Shell编程基础 1 -2

Zookeeper分布式协调框架及应用 3

Java基础增强：并发编程concurrent包/JVM内存管理/socket编程/NIO&AIO 4

Java基础增强：动态代理/反射RPC 编程基础 5

redis 速用 5

## 03/ Linux基础

学习linux的目的：

不是让你去精通linux

而是掌握一些大数据开发工程师所需具备的linux操作技能

### 3.1 系统安装准备

#### 3.1.1 Linux简介

1、Linux是一套**免费使用和自由传播**的类Unix**操作系统（主要用在服务器上）**，是一个基于POSIX和UNIX的多用户、多任务、支持多线程和多CPU的操作系统。它能运行主要的UNIX工具软件、应用程序和网络协议。它支持32位和64位硬件。Linux继承了Unix以网络为核心的设计思想，是一个性能稳定的多用户网络操作系统。

2、Linux操作系统诞生于1991 年10 月5 日（这是第一次正式向外公布时间）。Linux存在着**许多不同的Linux发行版本，但它们都使用了Linux内核**。Linux可安装在各种计算机硬件设备中，比如手机、平板电脑、路由器、视频游戏控制台、台式计算机、大型机和超级计算机。

补充：linux内核必须加上一个“界面”软件，才能让用户去使用

“界面”分两类：

a、命令行界面（有很多种，最流行的一种是 bash shell）

b、图形界面（也有很多种，目前最流行的有两种：gnome kde）

3、常见发行版

红帽企业版：RedHat Enterprise Linux（RHEL），红帽桌面版Fedora

Centos社区发行版：源自RedHat企业版，相比少了一些收费的应用软件

Ubuntu：桌面版、服务器版、移动端

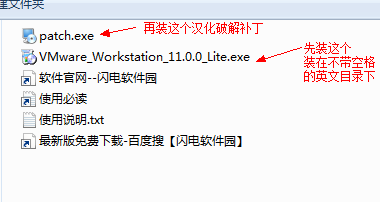
麒麟kylin：国防科技大学发行

红旗RedFlag：中科院发行

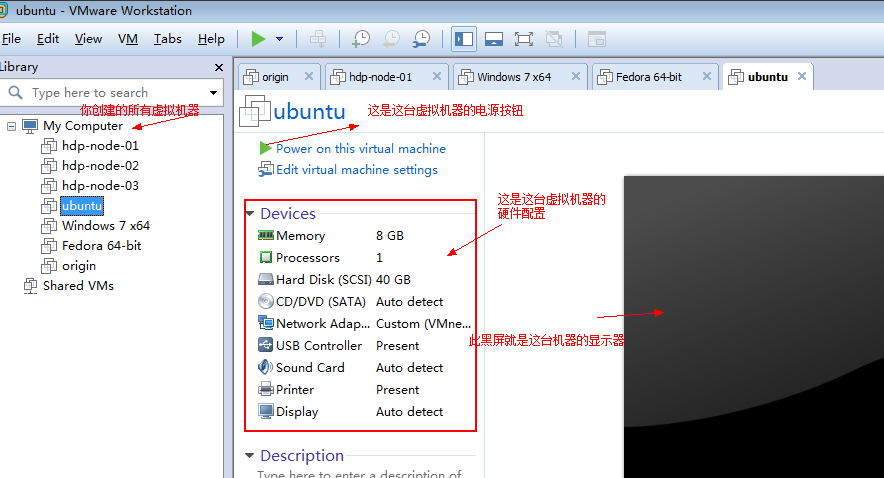
#### 3.1.2 VMware虚拟机软件

Vmware公司出品的一套虚拟化软件，可以在物理机上用软件的方式模拟出虚拟的计算机系统，而虚拟出的计算机系统可以像一台真实的机器一样，安装操作系统，运行软件等

类似的产品还有Oracle出品的免费版虚拟机软件VirtualBox

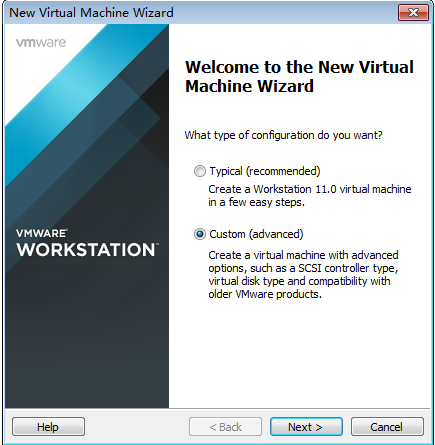


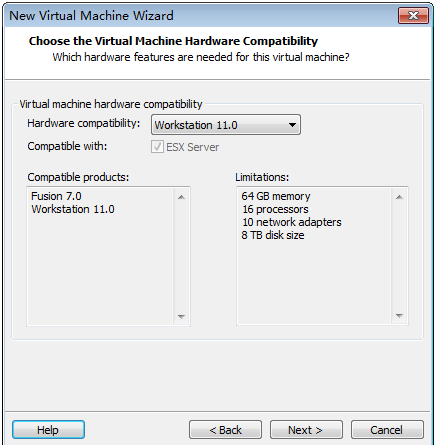
vmware 的使用界面：

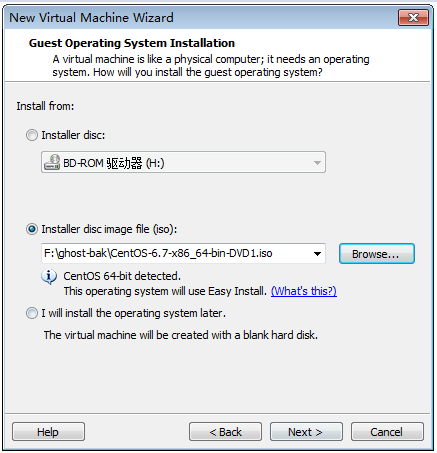


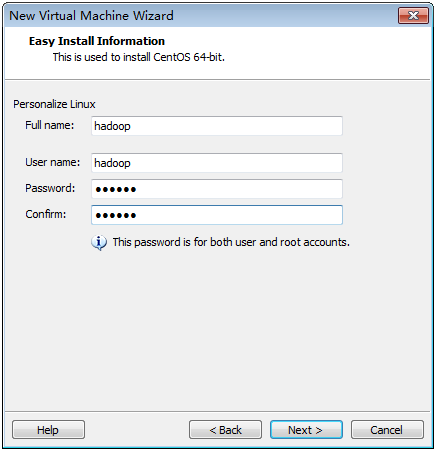
新建虚拟机步骤：

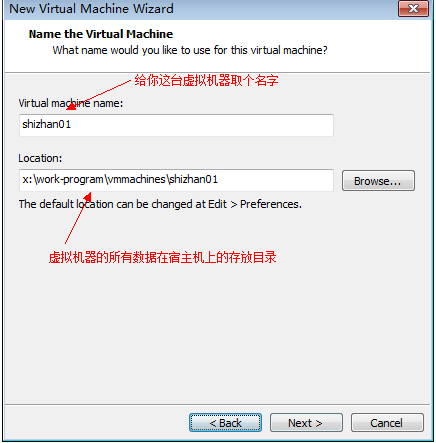
通过菜单： FIle--》new virtual machine



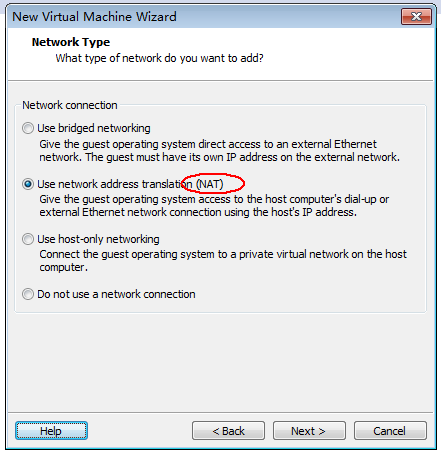








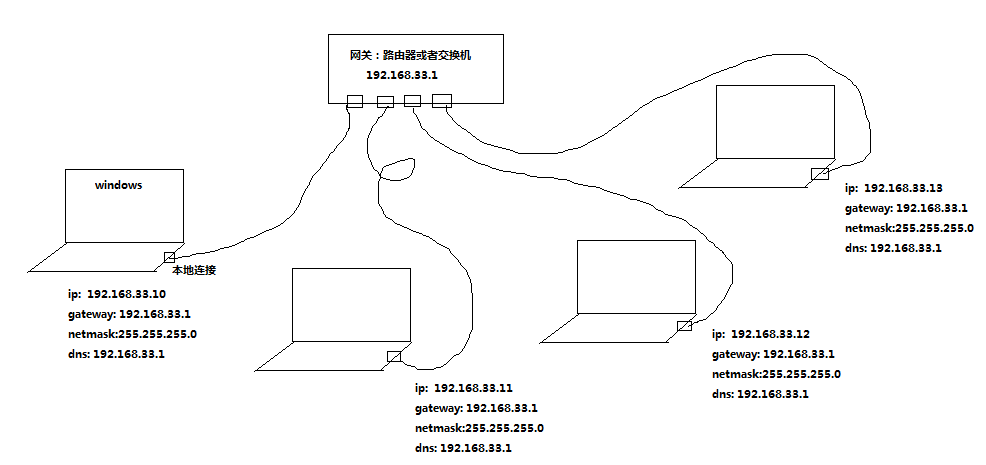
后面进入硬件资源分配，其中cpu给1个，内存至少给1G，网卡的选择如下图：



#### 3.1.3 Linux系统安装/图形界面使用

#### 3.1.4 网络基础及局域网配置

##### 1、简单局域网结构



##### 2、Vmware的3种网络模式配置

##### 3、ssh远程连接linux服务器

首先，准备一款ssh的客户端软件secureCRT

启动secureCRT

点击 闪电 图标，新建连接

填入目标服务器的主机名（域名或者IP），选择22端口，填入登录用户名

然后点击 connect

缓存（accept & save )服务器发送过来的指纹信息

然后会提示输入密码，填入点ok即可

### 3.2 Linux常用命令

#### 3.1.1 基本日常操作命令

1、查看当前所在的工作目录的全路径 pwd

[test@hdp-node-01 ~]$ pwd

*/home/test*

2、查看当前系统的时间 date

[test@hdp-node-01 ~]$ date +%Y-%m-%d

*2016-05-18*

[test@hdp-node-01 ~]$ date +%Y-%m-%d --date="-1 day" #加减也可以 month | year

*2016-05-17*

*设置时间*

*date -s "2016-05-23 01:01"*## 修改时间

*修改时间后，需要写入硬件bios才能在重启之后依然生效*

*hwclock -w*

3、查看有谁在线（哪些人登陆到了服务器）

who 查看当前在线

[test@hdp-node-01 ~]$ who

root tty1 2016-05-18 11:46

hadoop tty3 2016-05-18 11:46

hadoop pts/0 2016-05-18 10:57 (192.168.33.1)

last 查看最近的登陆历史记录

[test@hdp-node-01 ~]$ last -3

hadoop tty3 Wed May 18 11:46 still logged in

root tty1 Wed May 18 11:46 still logged in

hadoop pts/0 192.168.33.1 Wed May 18 10:57 still logged in

4、关机/重启

## 关机（必须用root用户）

shutdown -h now ## 立刻关机

shutdown -h +10 ## 10分钟以后关机

shutdown -h 12:00:00 ##12点整的时候关机

halt # 等于立刻关机

## 重启

shutdown -r now

reboot # 等于立刻重启

5、清屏

clear ## 或者用快捷键 ctrl + l

6、退出当前进程

ctrl+c 有些程序也可以用q键退出

7、挂起当前进程

ctrl+z ## 进程会挂起到后台

bg jobid ## 让进程在后台继续执行

fg jobid ## 让进程回到前台

8、echo

相当于java中System.out.println(userName)

|  |
| --- |
| [root@localhost bbb]# a="婷婷是我的梦中情人"  [root@localhost bbb]#  [root@localhost bbb]# echo a  a  [root@localhost bbb]# echo $a  婷婷是我的梦中情人 |

#### 3.2.2 目录操作

##### 1 查看目录信息

ls / ## 查看根目录下的子节点（文件夹和文件）信息

ls -al ## -a是显示隐藏文件 -l是以更详细的列表形式显示

ls -l 有一个别名： ll 可以直接使用ll <是两个L>

##### 2 切换工作目录

cd /home/hadoop ## 切换到用户主目录

cd ~ ## 切换到用户主目录

cd 什么路径都不带，则回到用户的主目录

cd - ## 回退到上次所在的目录

##### 3 创建文件夹

mkdir aaa ## 这是相对路径的写法

mkdir /data ## 这是绝对路径的写法

mkdir -p aaa/bbb/ccc ## 级联创建目录

##### 4 删除文件夹

rmdir aaa ## 可以删除空目录

rm -r aaa ## 可以把aaa整个文件夹及其中的所有子节点全部删除

rm -rf aaa ## 强制删除aaa

##### 5 修改文件夹名称

mv aaa angelababy

mv本质上是移动

mv install.log aaa/ 将当前目录下的install.log 移动到aaa文件夹中去

rename 可以用来批量更改文件名

|  |
| --- |
| [root@localhost bbb]# ll  total 0  -rw-r--r--. 1 root root 0 May 22 15:58 1.txt  -rw-r--r--. 1 root root 0 May 22 15:58 2.txt  -rw-r--r--. 1 root root 0 May 22 15:58 3.txt  [root@localhost bbb]# rename .txt .txt.bak \*  [root@localhost bbb]# ll  total 0  -rw-r--r--. 1 root root 0 May 22 15:58 1.txt.bak  -rw-r--r--. 1 root root 0 May 22 15:58 2.txt.bak  -rw-r--r--. 1 root root 0 May 22 15:58 3.txt.bak |

#### 3.2.3 文件操作

##### 1 创建文件

touch somefile.1

## 创建一个空文件

echo "i miss you,my baby">somefile.2

## 利用**重定向**“>”的功能，将一条指令的输出结果写入到一个文件中，会覆盖原文件内容，如果指定的文件不存在，则会创建出来

echo "huangxiaoming ,gun dan">>somefile.2

## 将一条指令的输出结果追加到一个文件中，不会覆盖原文件内容

##### 2 vi文本编辑器

**1、最基本用法**

vi somefile.4

1/ 首先会进入“**一般模式**”，此模式只接受各种命令快捷键，不能编辑文件内容

2/ 按i键，就会从一般模式进入**编辑模式**，此模式下，敲入的都是文件内容

3/ 编辑完成之后，按**Esc键退出编辑模式**，回到一般模式；

4/ 再按**：**，进入“**底行命令模式**”，输入**wq**命令，回车即可保存退出

**2、常用快捷键**

一些有用的快捷键（在一般模式下使用）：

a 在光标后一位开始插入

A 在该行的最后插入

I 在该行的最前面插入

gg 直接跳到文件的首行

G 直接跳到文件的末行

dd 删除一行

3dd 删除3行

yy 复制一行

3yy 复制3行

p 粘贴

u undo

~~ctrl + r redo~~

v 进入字符选择模式，选择完成后，按y复制，按p粘贴

ctrl+v 进入块选择模式，选择完成后，按y复制，按p粘贴

shift+v 进入行选择模式，选择完成后，按y复制，按p粘贴

**3、查找并替换**

（在底行命令模式中输入）

1显示行号

:set nu

2 隐藏行号

:set nonu

3 查找关键字

:/you ## 效果：查找文件中出现的you，并定位到第一个找到的地方，按n可以定位到下一个匹配位置（按N定位到上一个）

4 替换操作

:s/sad/bbb 查找光标所在行的第一个sad，替换为bbb

:%s/sad/bbb 查找文件中所有sad，替换为bbb

~~:1,$s/hadoop/root/g 将第一行到最后一行的hadoop替换为root~~

~~:1,$s/hadoop/root/c 将第一行到最后一行的hadoop替换为root(有提示)~~

##### 3 拷贝/删除/移动

cp somefile.1 /home/hadoop/

rm /home/hadoop/somefile.1

rm -rf /home/hadoop/somefile.1

mv /home/hadoop/somefile.1 ../

##### 4 查看文件内容

cat somefile 一次性将文件内容全部输出（控制台）

分页查看文件的命令：

more somefile 可以翻页查看, 下翻一页(空格) 上翻一页（b） 退出（q）

less somefile 可以翻页查看,下翻一页(空格) 上翻一页（b），上翻一行(↑) 下翻一行（↓） 可以搜索关键字（/keyword）

跳到文件末尾： G

跳到文件首行： gg

退出less ： q

tail -10 install.log 查看文件尾部的10行

tail +10 install.log 查看文件 10-->末行

**tail -f** install.log 小f跟踪文件的唯一inode号，就算文件改名后，还是跟踪原来这个inode表示的文件

tail -F install.log 大F按照文件名来跟踪

head -10 install.log 查看文件头部的10行

##### 5 打包压缩

1、gzip压缩

gzip a.txt

2、解压

~~gunzip a.txt.gz~~

**gzip -d a.txt.gz**

3、bzip2压缩

bzip2 a

4、解压

bunzip2 a.bz2

bzip2 -d a.bz2

5、**打包**：将指定文件或文件夹

tar -cvf bak.tar ./aaa

将/etc/password追加文件到bak.tar中(r)

tar -rvf bak.tar /etc/password

6、解压

tar -xvf bak.tar

7、**打包并压缩**

**tar -zcvf a.tar.gz aaa/**

**8、解包并解压缩(重要的事情说三遍!!!)**

**tar -zxvf a.tar.gz**

解压到/usr/下

tar -zxvf a.tar.gz **-C /usr**

9、查看压缩包内容

tar -ztvf a.tar.gz

zip/unzip

10、打包并压缩成bz2

tar -jcvf a.tar.bz2

11、解压bz2

tar -jxvf a.tar.bz2

#### 3.2.4 查找命令

##### 1 常用查找命令的使用

1、查找可执行的命令所在的路径：

which ls

2、查找可执行的命令和帮助的位置：

whereis ls

3、从某个文件夹开始查找文件

**find / -name "hadooop\*"**

find / -name "hadooop\*" -ls

5、查找并删除

find / -name "hadooop\*" -ok rm {} \;

find / -name "hadooop\*" -exec rm {} \;

6、查找用户为hadoop的文件

find /usr -user hadoop -ls

7、查找用户为hadoop的文件夹

find /home -user hadoop -type d -ls

8、查找权限为777的文件

find / -perm -777 -type d -ls

\*\*、显示历史命令

history

##### 2 grep命令

1/ 基本使用

查询包含hadoop的行

grep hadoop /etc/password

grep aaa ./\*.txt

2/ cut截取以:分割保留第七段

grep hadoop /etc/passwd | cut -d: -f7

3/ 查询不包含hadoop的行

grep -v hadoop /etc/passwd

4/ 正则表达包含hadoop

grep 'hadoop' /etc/passwd

5/ 正则表达(点代表任意一个字符)

grep 'h.\*p' /etc/passwd

6/ 正则表达以hadoop开头

grep '^hadoop' /etc/passwd

7/ 正则表达以hadoop结尾

grep 'hadoop$' /etc/passwd

~~正则表达式的简单规则：~~

. : 任意一个字符

a\* : 任意多个a(零个或多个a)

a? : 零个或一个a

a+ : 一个或多个a

.\* : 任意多个任意字符

\. : 转义.

o\{2\} : o重复两次

查找不是以#开头的行

grep -v '^#' a.txt | grep -v '^$'

以h或r开头的

grep '^[hr]' /etc/passwd

不是以h和r开头的

grep '^[^hr]' /etc/passwd

不是以h到r开头的

grep '^[^h-r]' /etc/passwd

#### 3.2.5 文件权限的操作

##### 1 linux文件权限的描述格式解读

drwxr-xr-x （也可以用二进制表示 111 101 101 --> 755）

d：标识节点类型（d：文件夹 -：文件 l:链接）

r：可读 w：可写 x：可执行

第一组rwx： ## 表示这个文件的**拥有者**对它的权限：可读可写可执行

第二组r-x： ## 表示这个文件的**所属组用户**对它的权限：可读，不可写，可执行

第三组r-x： ## 表示这个文件的**其他用户**（相对于上面两类用户）对它的权限：可读，不可写，可执行

##### 2 修改文件权限

chmod g-rw haha.dat ## 表示将haha.dat对所属组的rw权限取消

chmod o-rw haha.dat ## 表示将haha.dat对其他人的rw权限取消

chmod u+x haha.dat ## 表示将haha.dat对所属用户的权限增加x

chmod a-x haha.dat ## 表示将haha.dat对所用户取消x权限

也可以用数字的方式来修改权限

chmod 664 haha.dat

就会修改成 rw-rw-r--

如果要将一个文件夹的所有内容权限统一修改，则可以-R参数

chmod -R 770 aaa/

##### 3 修改文件所有权

<只有root权限能执行>

chown angela aaa ## 改变所属用户

chown :angela aaa ## 改变所属组

chown angela:angela aaa/ ## 同时修改所属用户和所属组

*补充：*

*r: 对文件来说，是可读取内容； 对文件夹来说，是可以ls*

*w: 对文件来说，是可修改文件的内容；对文件夹来说，是可以在其中创建或者删除子节点*

*x： 对文件来说，是能否运行这个文件；对文件夹来说，是能否cd进入这个目录*

#### 3.2.6 基本的用户管理

需要我们掌握的：

添加一个用户：

1、 useradd spark

2、 passwd spark 根据提示设置密码；

即可

删除一个用户：

userdel -r spark 加一个-r就表示把用户及用户的主目录都删除

##### 1 添加用户

添加一个tom用户，设置它属于users组，并添加注释信息

分步完成：**useradd tom**

usermod -g users tom

usermod -c "hr tom" tom

一步完成：useradd -g users -c "hr tom" tom

设置tom用户的密码

passwd tom

##### 2 修改用户

修改tom用户的登陆名为tomcat

usermod -l tomcat tom

将tomcat添加到sys和root组中

usermod -G sys,root tomcat

查看tomcat的组信息

groups tomcat

##### 3 用户组操作

添加一个叫america的组

groupadd america

将jerry添加到america组中

usermod -g america jerry

将tomcat用户从root组和sys组删除

gpasswd -d tomcat root

gpasswd -d tomcat sys

将america组名修改为am

groupmod -n am america

##### 4 为用户配置sudo权限

用root编辑 vi /etc/sudoers

在文件的如下位置，为hadoop添加一行即可

root ALL=(ALL) ALL

hadoop ALL=(ALL) ALL

然后，hadoop用户就可以用sudo来执行系统级别的指令

[hadoop@shizhan ~]$ sudo useradd huangxiaoming

#### 3.2.7 系统管理操作

##### 1 挂载外部存储设备

可以挂载光盘、硬盘、磁带、光盘镜像文件等

1/ 挂载光驱

mkdir /mnt/cdrom 创建一个目录，用来挂载

mount -t iso9660 -o ro **/dev/cdrom** /mnt/cdrom/ 将设备/dev/cdrom挂载到 挂载点 ： /mnt/cdrom中

2/ 挂载光盘镜像文件（.iso文件）

mount -t iso9660 -o loop **/home/hadoop/Centos-6.7.DVD.iso** /mnt/centos

*注：挂载的资源在重启后即失效，需要重新挂载。要想自动挂载，可以将挂载信息设置到/etc/fstab配置文件中，如下：*

|  |
| --- |
| /dev/cdrom /mnt/cdrom iso9660 defaults 0 0  /root/CentOS-6.7-x86\_64-bin-DVD1.iso /mnt/centos iso9660 defaults,ro,loop 0 0 |

3/ 卸载 umount

umount /mnt/cdrom

\*\* 存储空间查看

df -h

##### 2 统计文件或文件夹的大小

du -sh /mnt/cdrom/packages ## 统计指定路径下的所有子目录和文件的大小

df -h 查看磁盘的剩余空间

##### 3 系统服务管理

service --status-all # 查看系统所有的后台服务进程

service sshd status # 查看指定的后台服务进程的状态

service sshd stop

service sshd start

service sshd restart

配置后台服务进程的开机自启

chkconfig httpd on ## 让httpd服务开机自启

chkconfig httpd off ## 让httpd服务开机不要自启

|  |
| --- |
| *[root@localhost ~]# chkconfig httpd off*  *[root@localhost ~]# chkconfig --list | grep httpd*  *httpd 0:off 1:off 2:off 3:off 4:off 5:off 6:off*  *[root@localhost ~]# chkconfig --level 35 httpd on*  *[root@localhost ~]# chkconfig --list | grep httpd*  *httpd 0:off 1:off 2:off 3:on 4:off 5:on 6:off* |

##### 4 系统启动级别管理

vi /etc/inittab

# Default runlevel. The runlevels used are:

# 0 - halt (Do NOT set initdefault to this)

# 1 - Single user mode

# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)

# 3 - Full multiuser mode ## 没有图形界面的全功能的多用户的启动级别

# 4 - unused

# 5 - X11 ## 有图形界面的启动级别

# 6 - reboot (Do NOT set initdefault to this)

#

id:3:initdefault: ## 配置默认启动级别

## 通常将默认启动级别设置为：3

##### 5 进程管理

top

free

ps -ef | grep ssh

kill -9 2358 ## 将指定进程号的进程杀死

注意：grep搜索关键词的时候会把自己也搜索出来，对比以下两种写法

|  |
| --- |
| *[root@localhost ~]# ps -ef | grep sixunhuan*  *root 2857 2465 30 02:41 pts/0 00:00:07 sh sixunhuan.sh*  *root 2874 2858 0 02:42 pts/1 00:00:00 grep sixunhuan*  *[root@localhost ~]# ps -ef | grep sixunhuan | grep -v grep*  *root 2857 2465 34 02:41 pts/0 00:00:25 sh sixunhuan.sh*  *[root@localhost ~]# kill -9 2857* |

#### 3.2.8 SSH免密登陆配置

##### 1 SSH工作机制

**1、相关概念**

SSH 为 Secure Shell（安全外壳协议） 的缩写。

很多ftp、pop和telnet在本质上都是不安全的，因为它们在网络上用明文传送口令和数据，别有用心的人非常容易就可以截获这些口令和数据。而SSH就是专为远程登录会话和其他网络服务提供安全性的协议。

SSH的具体实现是由**客户端和服务端**的软件组成的

服务端是一个守护进程(sshd)，他在后台运行并响应来自客户端的连接请求。

客户端包含ssh程序以及像scp（远程拷贝）、slogin（远程登陆）、sftp（安全文件传输）等其他的应用程序。

**2、认证机制**

从客户端来看，SSH提供两种级别的安全验证。

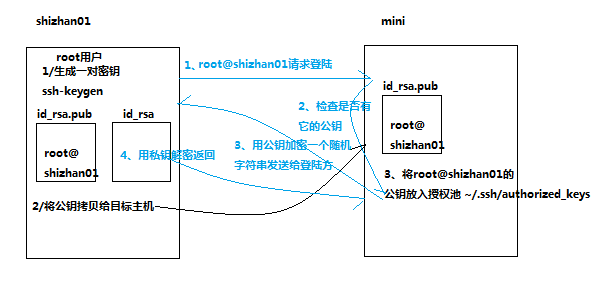
* 第一种方式（基于口令的安全验证）

只要你知道自己帐号和口令，就可以登录到远程主机。

* 第二种方式（基于密钥的安全验证）

需要依靠密匙，也就是你必须为自己创建一对密匙，并把公用密匙放在需要访问的服务器上。如果你要连接到SSH服务器上，客户端软件就会向服务器发出请求，请求用你的密匙进行安全验证。服务器收到请求之后，先在该服务器上你的主目录下寻找你的公用密匙，然后把它和你发送过来的公用密匙进行比较。如果两个密匙一致，服务器就用公用密匙加密“质询”（challenge）并把它发送给客户端软件。客户端软件收到“质询”之后就可以用你的私人密匙解密再把它发送给服务器。

##### 2 密钥登陆方式配置



假如 A 要登陆 B

在A上操作：

1/ 首先生成密钥对

ssh-keygen (提示时，直接回车即可)

2/ 再将A自己的公钥拷贝并追加到B的授权列表文件authorized\_keys中

ssh-copy-id B

#### 3.2.9 网络管理

##### 1 主机名配置

1/ 查看主机名

hostname

2/ 修改主机名(重启后无效)

hostname hadoop

3/ 修改主机名(重启后永久生效)

vi /ect/sysconfig/network

4/ centos7修改主机名

hostnamectl set-hostname hadoop01 z将主机名改为hadoop01

##### 2 IP地址配置

修改IP地址

1/ 方式一：setup

用root输入setup命令，进入交互式修改界面

2/ 方式二：修改配置文件

(重启后永久生效)

vi /etc/sysconfig/network-scripts/ifcfg-eth0

3/ 方式三：ifconfig命令

(重启后无效)

ifconfig eth0 192.168.12.22

##### 3 域名映射

/etc/hosts文件 用于在通过主机名进行访问时做ip地址解析之用

所以，你想访问一个什么样的主机名，就需要把这个主机名和它对应的ip地址配置在/etc/hosts文件中

|  |
| --- |
| [root@angelababy01 ~]# vi /etc/hosts  127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4  ::1 localhost localhost.localdomain localhost6 localhost6.localdomain6  192.168.33.11 angelababy01  192.168.33.12 angelababy02  192.168.33.13 angelababy03 |

##### 4 网络服务管理

1/ 后台服务管理

service network status 查看指定服务的状态

service network stop 停止指定服务

service network start 启动指定服务

service network restart 重启指定服务

service --status-all 查看系统中所有的后台服务

2/ 设置后台服务的自启配置

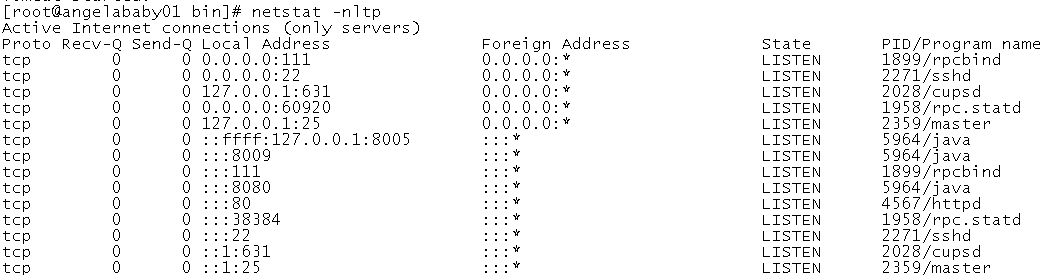
chkconfig 查看所有服务器自启配置

chkconfig iptables off 关掉指定服务的自动启动

chkconfig iptables on 开启指定服务的自动启动

##### 5 系统中网络进程的端口监听情况

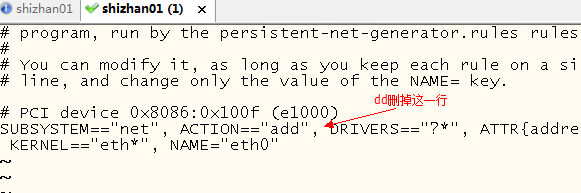
netstat -nltp



##### \*\* 补充：Linux的网卡坏了怎么办

1、先删掉旧网卡

vi /etc/udev/rules.d/70-persistent-net.rules



2、halt linux系统

3、通过vmware卸载原来的网卡，再新增一块网卡

4、70-persistent-net.rules 里面会自动生成那块新网卡的硬件信息

5、编辑网卡的配置文件/etc/sysconfig/network-script/ifcfg-eth0

删掉其中的UUID HWADDR

并且修改IP配置等，如下：

|  |
| --- |
| DEVICE="eth0"  BOOTPROTO=none  ONBOOT="yes"  TYPE="Ethernet"  IPADDR=192.168.33.20  PREFIX=24  GATEWAY=192.168.33.1  DNS1=192.168.33.1  NAME="System eth0" |

1. service network restart 重启网络服务即可

##### \*\* 补充：Centos7克隆机上不了网怎么办

如果使用的是centOs7克隆机器后上不了网的话 参考:http://blog.csdn.net/u013248434/article/details/70276328

参考该文章不需要去删除 /etc/udev/rules.d/70-persistent-ipoib.rules文件

需要将 etc/sysconfig/network-scripts/ipcfg-ens32修改成下面的就可以了

注:ipcfg-ens32 文件名你的机器不一定和我一样但是ipcfg- 是一样的后面的部分来自下面配置中的NAME的值

TYPE="Ethernet"

BOOTPROTO=none

DEFROUTE="yes"

PEERDNS="yes"

PEERROUTES="yes"

IPV4\_FAILURE\_FATAL="no"

IPV6INIT="yes"

IPV6\_AUTOCONF="yes"

IPV6\_DEFROUTE="yes"

IPV6\_PEERDNS="yes"

IPV6\_PEERROUTES="yes"

IPV6\_FAILURE\_FATAL="no"

IPV6\_ADDR\_GEN\_MODE="stable-privacy"

NAME="ens32"

DEVICE="ens32"

ONBOOT="yes"

IPADDR=192.168.175.132

GATEWAY=192.168.175.2

NETMASK=255.255.255.0

DNS1=192.168.175.2

## 04/ Linux上常用软件安装

### 4.1 Linux系统软件安装方式

Linux上的软件安装有以下几种常见方式：

1. 二进制发布包

软件已经针对具体平台编译打包发布，只要解压，修改配置即可

1. RPM发布包

软件已经按照redhat的包管理工具规范RPM进行打包发布，需要获取到相应的软件RPM发布包，然后用RPM命令进行安装

1. Yum在线安装

软件已经以RPM规范打包，但发布在了网络上的一些服务器上，可用yum在线安装服务器上存在的rpm软件，并且会自动解决软件安装过程中的库依赖问题

（注：类似于maven）

1. 源码编译安装

软件以源码工程的形式发布，需要获取到源码工程后用相应开发工具进行编译打包部署

### 4.2 JAVA软件安装——JDK安装

1、上传jdk压缩包

通过sftp工具上传即可

2、解压jdk压缩包

tar -zxvf jdk-7u67-linux-x64.gz -C /usr/local/

3、修改环境变量PATH

vi /etc/profile

在文件最后加两行：

export JAVA\_HOME=/usr/local/jdk1.7.0\_67

export PATH=$PATH:$JAVA\_HOME/bin

4、让环境变量生效

source /etc/profile

即可

### 4.3 JAVA软件安装——Tomcat安装

tar -zxvf /soft/apache-tomcat-7.0.47.tar.gz -C /usr/local/

cd /usr/local/apache-tomcat-7.0.47/bin/

./startup.sh

### 4.4 RPM方式软件安装——MySql数据库安装

#### 4.4.4 desktop版的centos中安装mysql

**没有上述那么麻烦，基本上都不缺依赖，过程如下：**

1、先装server

rpm -ivh MySQL-server-5.6.26-1.linux\_glibc2.5.x86\_64.rpm

发现了包冲突，执行删除包的命令解决：

rpm -e mysql-libs-5.1.73-5.el6\_6.x86\_64 --nodeps

然后继续重新安装server

rpm -ivh MySQL-server-5.6.26-1.linux\_glibc2.5.x86\_64.rpm

一定要确保server安装成功！！！

2、再装client

rpm -ivh MySQL-client-5.6.26-1.linux\_glibc2.5.x86\_64.rpm

3、启动server

service mysql start

检查启动是否成功：

service mysql status 或者 查看端口：netstat -nltp 看是否有进程在监听3306

4、用客户端连接到mysql-server上进行root密码的修改

注：在server安装时已经生成了一个随机的root密码 ，查看： cat /root/.mysql\_secret

然后，运行一个脚本来交互式地修改root密码即可：/usr/bin/mysql\_secure\_installation

#### 4.4.1 mini版的centos安装mysql服务

可以用yum方式来安装，此处演示在本地使用rpm包安装

*[root@mini ~]# rpm -ivh MySQL-server-5.1.73-1.glibc23.i386.rpm*

1、报错——缺少依赖包perl

error: Failed dependencies:

perl is needed by MySQL-server-5.1.73-1.glibc23.i386

*解决：yum install -y perl-devel*

2、报错——缺少依赖包libc.so.6

*[root@mini ~]# rpm -ivh MySQL-server-5.1.73-1.glibc23.i386.rpm*

error: Failed dependencies:

libc.so.6 is needed by MySQL-server-5.1.73-1.glibc23.i386

*解决：yum install -y libc.so.6*

3、报错——缺少依赖包 libgcc\_s.so.1

*[root@mini ~]# rpm -ivh MySQL-server-5.1.73-1.glibc23.i386.rpm*

error: Failed dependencies:

libgcc\_s.so.1 is needed by MySQL-server-5.1.73-1.glibc23.i386

*解决：yum install -y libc.so.1*

4、报错——依赖包冲突

*[root@mini ~]# rpm -ivh MySQL-server-5.1.73-1.glibc23.i386.rpm*

conflicts with file from package mysql-libs-5.1.73-5.el6\_6.x86\_64

*解决：*

*卸载依赖的冲突包： rpm -e mysql-libs-5.1.73-5.el6\_6 --nodeps*

5、安装完成——注意提示信息

再次执行安装

[root@mini ~]# rpm -ivh MySQL-server-5.1.73-1.glibc23.i386.rpm

*Preparing... ########################################### [100%]*

*1:MySQL-server ########################################### [100%]*

*PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !*

*To do so, start the server, then issue the following commands:*

*/usr/bin/mysqladmin -u root password 'new-password'*

*/usr/bin/mysqladmin -u root -h mini password 'new-password'*

*Alternatively you can run:*

*/usr/bin/mysql\_secure\_installation*

*\*\* 补充：*

如果发生rpm包冲突conflict，怎么处理

首先，找到有哪些包冲突

rpm -qa | grep mysql

然后，卸载冲突的包

rpm -e mysql-libs-5.1.73-5.el6\_6.x86\_64 --nodeps

#### 4.4.2 安装mysql客户端程序

1、用rpm命令安装客户端

*[root@mini ~]# rpm -ivh MySQL-client-5.1.73-1.glibc23.i386.rpm*

*error: Failed dependencies:*

*libncursesw.so.5 is needed by MySQL-client-5.1.73-1.glibc23.i386*

报错缺依赖，解决之：

yum install -y libncursesw.so.5 ## yum本地源需要更换光盘为dvd2

再次执行安装

[root@mini ~]# rpm -ivh MySQL-client-5.1.73-1.glibc23.i386.rpm

*Preparing... ########################################### [100%]*

*1:MySQL-client ########################################### [100%]*

#### 4.4.3 修改mysql用户root密码

安装完成后，先确定mysql服务进程是否启动

netstat -nltp

如果有mysql，会监听3306端口

如果没有，则手动启动

service mysql start

1、按照前面的提示，修改密码

[root@mini ~]# /usr/bin/mysql\_secure\_installation

进入交互式提示设置，按提示操作即可

补充：在资料中又给了一个5.6的mysql版本，这个安装完成后，mysql的root密码是自动生成的，注意看提示，密码保存在哪个位置

/root/.mysql\_secret

安装完成后，直接用这个生成的随机密码即可登录mysql

进入mysql后当然可以使用sql来修改root的密码

mysql>set password=password("root");

mysql>flush privileges;

4.5.4 开启mysql的远程登录权限

mysql>GRANT ALL PRIVILEGES ON \*.\* TO 'myuser'@'%' IDENTIFIED BY 'mypassword' WITH GRANT OPTION;

mysql>FLUSH PRIVILEGES;

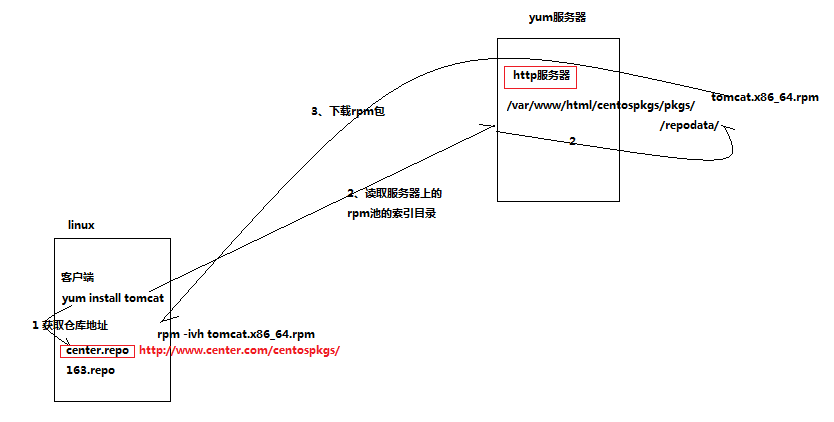
### 4.5 YUM方式软件安装

#### 4.5.1 从网络源使用yum安装软件

yum install mysql

#### 4.5.2 使用本地Yum源安装软件

yum仓库服务器本质上就是一台http服务器，服务器的目录中放置了rpm包，及rpm包的索引信息文件，即可为yum客户端提供rpm文件下载



既然如此，则完全可以在“本地”制作私有的yum库来为内网中的linux提供yum安装源

#### 4.5.3 制作私有yum源的两种方式

##### 1、制作基于本地磁盘路径的yum源

本地yum源，即yum源就在系统的本地目录中，所以连web服务器都不用，只要准备好rpm库存放路径及相关索引配置信息即可

示例：将Centos-DVD.iso盘制作成一个本地yum源

##### 2、制作基于内网web服务器的yum源

总结：

1、在一台机器上部署一个httpd 的 web服务器，确保httpd运行成功

2、将光盘镜像文件插入虚拟机的光驱

3、将光驱挂载到httpd的html/centos目录中

4、修改yum的repo库配置文件/etc/yum.repos.d/centos-local-web.repo，如下：

|  |
| --- |
| [c6-local-web]  name=CentOS-local-web  baseurl=http://angelababy01/centos6  gpgcheck=0  enabled=1  gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6 |

5、接下来，就可以用yum安装库中拥有的任何软件了

yum install -y tomcat6.x86\_64

详解版：

1、先安装一个http服务器： 首选httpd

先找一下我们要的包在仓库中有没有

yum list | grep httpd

yum install -y httpd

2、安装完成后，检查httpd的运行状态

service httpd status

3、在httpd服务的web目录中放入rpm库

比如，将centos安装光盘中的rpm库放入：

mkdir /var/www/html/centos

cp -r /mnt/cdrom/\* /var/www/html/centos/

*注：上面这种方式比较浪费空间，可以不用拷贝，其实只要创建一个软连接即可*

ln -s /mnt/cdrom /var/www/html/centos

4、用浏览器访问一下httpd服务，看是否能看到rpm库

http://mini1/centos

5、配置yum客户端repo地址文件

将本地http服务器加入repo地址

首先，将内置的源全部disable掉

cd /etc/yum.repos.d/

rename .repo .repo.bak \*

然后，新建一个repo源，

vi innet.repo

|  |
| --- |
| [innet]  name=innet  baseurl=http://192.168.33.20/centos  gpgcheck=0  enabled=1  gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6  ~ |

接下来，检查我们自定义的源是否生效

|  |
| --- |
| [root@shizhan01 yum.repos.d]# yum repolist  Loaded plugins: fastestmirror, refresh-packagekit, security  Loading mirror speeds from cached hostfile  innet | 4.0 kB 00:00  innet/primary\_db | 4.6 MB 00:00  repo id repo name status  innet innet 6,575  repolist: 6,575 |

再接下来，就可以使用yum像从公网安装软件一样在内网服务器上下载软件进行安装了

yum install -y 你要的软件

### 4.6 C语言软件源码编译安装——redis服务器安装

用源码工程来编译安装

1/ 到官网下载最新stable版

2/ 解压源码并进入目录 tar -zxvf redis-2.8.19.tar.gz -C ./redis-src/

3/ make

如果报错提示缺少gcc，则安装gcc ： yum install -y gcc

如果报错提示：Newer version of jemalloc required

则在make时加参数：make MALLOC=libc

4/ 安装redis，指定安装目录,如 /usr/local/redis

make PREFIX=/usr/local/redis install

6/ 拷贝一份配置文件到安装目录下

切换到源码目录，里面有一份配置文件 redis.conf，然后将其拷贝到安装路径下

cp redis.conf /usr/local/redis/

7/ 启动redis

cd /usr/local/redis

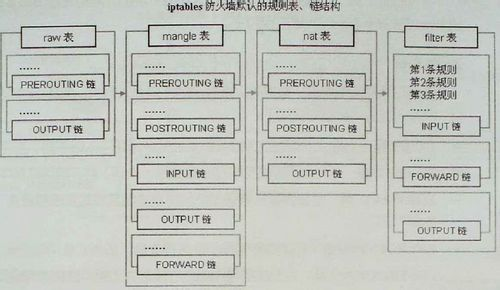
bin/redis-server redis.conf

## 01/ linux操作增强

### 1.1 防火墙配置

防火墙根据配置文件/etc/sysconfig/iptables来控制本机的“出、入”网络访问行为

其对行为的配置策略有四个策略表



#### 必备技能

注意 centos7中的防火墙服务名称为firewalld 对应的指令中的iptables需要改为firewalld

查看防火墙状态

service iptables status

关闭防火墙

service iptables stop

启动防火墙

service iptables start

禁止防火墙自启

chkconfig iptables off

#### ~~1.1.2 扩展了解~~

1、列出iptables规则

iptables -L -n

列出iptables规则并显示规则编号

iptables -L -n --line-numbers

2、列出iptables nat表规则（默认是filter表）

iptables -L -n -t nat

3、清除默认规则（注意默认是filter表，如果对nat表操作要加-t nat）

#清除所有规则

iptables -F

#重启iptables发现规则依然存在，因为没有保存

service iptables restart

#保存配置

service iptables save

4、禁止ssh登陆（若果服务器在机房，一定要小心）

iptables -A INPUT -p tcp --dport 22 -j DROP

#删除规则

iptables -D INPUT -p tcp --dport 22 -j DROP

加入一条INPUT规则开放80端口

iptables -I INPUT -p tcp --dport 80 -j ACCEPT

~~加入一条NAT规则~~

~~iptables -t nat -A PREROUTING -d 119.254.228.20 -p tcp -m tcp --dport 8022 -j DNAT --to-destination 172.16.0.111:22~~

~~删除一条NAT规则~~

~~iptables -t nat -D PREROUTING 6~~

### 1.2 高级文本处理命令

#### 1.2.1 cut命令

cut命令可以从一个文本文件或者文本流中提取文本列。

cut语法

[root@www ~]# cut -d'分隔字符' -f fields ## 用于有特定分隔字符

[root@www ~]# cut -c 字符区间 ## 用于排列整齐的信息

选项与参数：

-d：后面接分隔字符。与 -f 一起使用；

-f：依据 -d 的分隔字符将一段信息分割成为数段，用 -f 取出第几段的意思；

-c：以字符 (characters) 的单位取出固定字符区间；

PATH 变量如下

[root@www ~]# echo $PATH

*/bin:/usr/bin:/sbin:/usr/sbin:/usr/local/bin:/usr/X11R6/bin:/usr/games*

*# 1 | 2 | 3 | 4 | 5 | 6 | 7*

将 PATH 变量取出，找出第五个路径。

# echo $PATH | cut -d':' -f 5

*/usr/local/bin*

将 PATH 变量取出，找出第三和第五个路径。

#echo $PATH | cut -d ':' -f 3,5

*/sbin:/usr/local/bin*

将 PATH 变量取出，找出第三到最后一个路径。

echo $PATH | cut -d ':' -f 3-

*/sbin:/usr/sbin:/usr/local/bin:/usr/X11R6/bin:/usr/games*

将 PATH 变量取出，找出第一到第三个路径。

#echo $PATH | cut -d ':' -f 1-3

*/bin:/usr/bin:/sbin:*

将 PATH 变量取出，找出第一到第三，还有第五个路径。

#echo $PATH | cut -d ':' -f 1-3,5

*/bin:/usr/bin:/sbin:/usr/local/bin*

实用例子:只显示/etc/passwd的用户和shell

#cat /etc/passwd | cut -d ':' -f 1,7

*root:/bin/bash*

*daemon:/bin/sh*

*bin:/bin/sh*

#### 1.2.2 sed命令

**1/ 删除：d命令**

$ sed '2d' example -----删除example文件的第二行。

$ sed '2,$d' example -----删除example文件的第二行到末尾所有行。

$ sed '$d' example -----删除example文件的最后一行。

$ sed '/test/'d example -----删除example文件所有包含test的行。

**2/ 替换：s命令**

$ sed 's/test/mytest/g' example

*## 在整行范围内把test替换为mytest。如果没有g标记，则只有每行第一个匹配的test被替换成mytest。*

$ sed -n 's/^test/mytest/p' example

*## (-n)选项和p标志一起使用表示只打印那些发生替换的行。也就是说，如果某一行开头的test被替换成mytest，就打印它。*

$ sed 's/^192.168.0.1/&localhost/' example

*## &符号表示追加一个串到找到的串后。所有以192.168.0.1开头的行都会被替换成它自已加 localhost，变成192.168.0.1localhost。*

$ sed -n 's/\(love\)able/\1rs/p' example

*## love被标记为1，所有loveable会被替换成lovers，而且替换的行会被打印出来。*

$ sed 's#10#100#g' example

*## 不论什么字符，紧跟着s命令的都被认为是新的分隔符，所以，“#”在这里是分隔符，代替了默认的“/”分隔符。表示把所有10替换成100。*

*选定行的范围：逗号*

$ sed -n '/test/,/check/p' example

*## 所有在模板test和check所确定的范围内的行都被打印。*

$ sed -n '5,/^test/p' example

*## 打印从第五行开始到第一个包含以test开始的行之间的所有行。*

$ sed '/test/,/check/s/$/sed test/' example

*## 对于模板test和west之间的行，每行的末尾用字符串sed test替换。*

*多点编辑：e命令*

$ sed -e '1,5d' -e 's/test/check/' example

*## (-e)选项允许在同一行里执行多条命令。如例子所示，第一条命令删除1至5行，第二条命令用check替换test。命令的执行顺序对结果有影响。如果两个命令都是替换命令，那么第一个替换命令将影响第二个替换命令的结果。*

$ sed --expression='s/test/check/' --expression='/love/d' example

*## 一个比-e更好的命令是--expression。它能给sed表达式赋值。*

**3/ 从文件读入：r命令**

$ sed '/test/r file' example

*-----file里的内容被读进来，显示在与test匹配的行下面，如果匹配多行，则file的内容将显示在所有匹配行的下面。*

**4/ 写入文件：w命令**

$ sed -n '/test/w file' example

-----在example中所有包含test的行都被写入file里。

**5/ 追加命令：a命令**

$ sed '/^test/a\\--->this is a example' example

*## '--->this is a example'被追加到以test开头的行后面，sed要求命令a后面有一个反斜杠。*

**6/ 插入：i命令**

$ sed '/test/i\\some thing new -------------------------' example

如果test被匹配，则把反斜杠后面的文本插入到匹配行的前面。

**7/ 下一个：n命令**

$ sed '/test/{ n; s/aa/bb/; }' example

-----如果test被匹配，则移动到匹配行的下一行，替换这一行的aa，变为bb，并打印该行，然后继续。

**8/ 退出：q命令**

$ sed '10q' example

-----打印完第10行后，退出sed。

#### 1.2.3 awk命令

awk是一个强大的文本分析工具，相对于grep的查找，sed的编辑，awk在其对数据分析并生成报告时，显得尤为强大。简单来说awk就是把文件逐行的读入，以空格为默认分隔符将每行切片，切开的部分再进行各种分析处理。

假设last -n 5的输出如下

[root@www ~]# last -n 5 ## 仅取出前五行

*root pts/1 192.168.1.100 Tue Feb 10 11:21 still logged in*

*root pts/1 192.168.1.100 Tue Feb 10 00:46 - 02:28 (01:41)*

*root pts/1 192.168.1.100 Mon Feb 9 11:41 - 18:30 (06:48)*

*dmtsai pts/1 192.168.1.100 Mon Feb 9 11:41 - 11:41 (00:00)*

*root tty1 Fri Sep 5 14:09 - 14:10 (00:01)*

如果只是显示最近登录的5个帐号

# last -n 5 | awk '{print $1}'

*root*

*root*

*root*

*dmtsai*

*root*

awk工作流程是这样的：读入有'\n'换行符分割的一条记录，然后将记录按指定的域分隔符划分域，填充域，$0则表示所有域,$1表示第一个域,$n表示第n个域。默认域分隔符是"空白键"或"[tab]键",所以$1表示登录用户，$3表示登录用户ip,以此类推。

如果只是显示/etc/passwd的账户

# cat /etc/passwd |awk -F ':' '{print $1}'

*root*

*daemon*

*bin*

*sys*

这种是awk+action的示例，每行都会执行action{print $1}。

-F指定域分隔符为':'

如果只是显示/etc/passwd的账户和账户对应的shell,而账户与shell之间以tab键分割

#cat /etc/passwd |awk -F ':' '{print $1"\t"$7}'

root /bin/bash

daemon /bin/sh

bin /bin/sh

sys /bin/sh

如果只是显示/etc/passwd的账户和账户对应的shell,而账户与shell之间以逗号分割,而且在所有行添加列名name,shell,在最后一行添加"blue,/bin/nosh"。

cat /etc/passwd |awk -F ':' 'BEGIN {print "**name,shell**"} {print $1","$7} END {print "**blue,/bin/nosh**"}'

***name,shell***

*root,/bin/bash*

*daemon,/bin/sh*

*bin,/bin/sh*

*sys,/bin/sh*

*....*

***blue,/bin/nosh***

awk工作流程是这样的：先执行BEGING，然后读取文件，读入有/n换行符分割的一条记录，然后将记录按指定的域分隔符划分域，填充域，$0则表示所有域,$1表示第一个域,$n表示第n个域,随后开始执行模式所对应的动作action。接着开始读入第二条记录······直到所有的记录都读完，最后执行END操作。

搜索/etc/passwd有root关键字的所有行

#awk -F: '/root/' /etc/passwd

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

这种是pattern的使用示例，匹配了pattern(这里是root)的行才会执行action(没有指定action，默认输出每行的内容)。

搜索支持正则，例如找root开头的: awk -F: '/^root/' /etc/passwd

搜索/etc/passwd有root关键字的所有行，并显示对应的shell

# awk -F':' '/root/{print $7}' /etc/passwd

/bin/bash

这里指定了action{print $7}

统计/etc/passwd:文件名，每行的行号，每行的列数，对应的完整行内容:

#awk -F ':' '{print "filename:" FILENAME ",linenumber:" NR ",columns:" NF ",linecontent:"$0}' */etc/passwd*

*filename:/etc/passwd,linenumber:1,columns:7,linecontent:root:x:0:0:root:/root:/bin/bash*

*filename:/etc/passwd,linenumber:2,columns:7,linecontent:daemon:x:1:1:daemon:/usr/sbin:/bin/sh*

*filename:/etc/passwd,linenumber:3,columns:7,linecontent:bin:x:2:2:bin:/bin:/bin/sh*

*filename:/etc/passwd,linenumber:4,columns:7,linecontent:sys:x:3:3:sys:/dev:/bin/sh*

使用printf替代print,可以让代码更加简洁，易读

awk -F ':' '{printf("filename:%s,linenumber:%s,columns:%s,linecontent:%s\n",FILENAME,NR,NF,$0)}' /etc/passwd

### 1.3 crontab配置

#### 1、命令功能

通过crontab 命令，我们可以在固定的间隔时间执行指定的系统指令或 shell script脚本。时间间隔的单位可以是分钟、小时、日、月、周及以上的任意组合。这个命令非常适合周期性的日志分析或数据备份等工作。

#### 2、安装crontab

yum install crontabs

服务操作说明：

service crond start  ## 启动服务

service crond stop   ## 关闭服务

service crond restart ## 重启服务

service crond reload  ## 重新载入配置

## 查看crontab服务状态：

service crond status

## 手动启动crontab服务：

service crond start

## 查看crontab服务是否已设置为开机启动，执行命令：

chkconfig --list

## 加入开机自动启动：

chkconfig  --level 35 crond on

#### 3、命令格式

crontab [-u user] file

crontab [-u user] [ -e | -l | -r ]

参数说明：

-u user：用来设定某个用户的crontab服务，例如，“-u ixdba”表示设定ixdba用户的crontab服务，此参数一般有root用户来运行。

file：file是命令文件的名字,表示将file做为crontab的任务列表文件并载入crontab。

-e：编辑某个用户的crontab文件内容。如果不指定用户，则表示编辑当前用户的crontab文件。

-l：显示某个用户的crontab文件内容，如果不指定用户，则表示显示当前用户的crontab文件内容。

-r：删除定时任务配置，从/var/spool/cron目录中删除某个用户的crontab文件，如果不指定用户，则默认删除当前用户的crontab文件。

-i：在删除用户的crontab文件时给确认提示。

命令示例：

crontab file [-u user] ## 用指定的文件替代目前的crontab。

***必掌握：***

crontab -l [-u user] ## 列出用户目前的crontab.

crontab -e [-u user] ## 编辑用户目前的crontab.

#### 4、调度配置

##### 4.1 配置说明

基本格式 :

\*　　\*　　\*　　\*　　\*　　command

分　时　日　月　周　命令

第1列表示分钟1～59 每分钟用\*或者 \*/1表示

第2列表示小时0～23（0表示0点） 7-9表示：8点到10点之间

第3列表示日期1～31

第4列表示月份1～12

第5列标识号星期0～6（0表示星期天）

第6列要运行的命令

##### 4.2 配置示例

\*/1 \* \* \* \* date >> /root/date.txt

上面的例子表示每分钟执行一次date命令

30 21 \* \* \* /usr/local/etc/rc.d/httpd restart

上面的例子表示每晚的21:30重启apache。

45 4 1,10,22 \* \* /usr/local/etc/rc.d/httpd restart

上面的例子表示每月1、10、22日的4 : 45重启apache。

10 1 \* \* 6,0 /usr/local/etc/rc.d/httpd restart

上面的例子表示每周六、周日的1 : 10重启apache。

0,30 18-23 \* \* \* /usr/local/etc/rc.d/httpd restart

上面的例子表示在每天18 : 00至23 : 00之间每隔30分钟重启apache。

0 23 \* \* 6 /usr/local/etc/rc.d/httpd restart

上面的例子表示每星期六的11 : 00 pm重启apache。

\* \*/1 \* \* \* /usr/local/etc/rc.d/httpd restart

上面的例子每一小时重启apache

\* 23-7/1 \* \* \* /usr/local/etc/rc.d/httpd restart

上面的例子晚上11点到早上7点之间，每隔一小时重启apache

0 11 4 \* mon-wed /usr/local/etc/rc.d/httpd restart

上面的例子每月的4号与每周一到周三的11点重启apache

0 4 1 jan \* /usr/local/etc/rc.d/httpd restart

上面的例子一月一号的4点重启apache