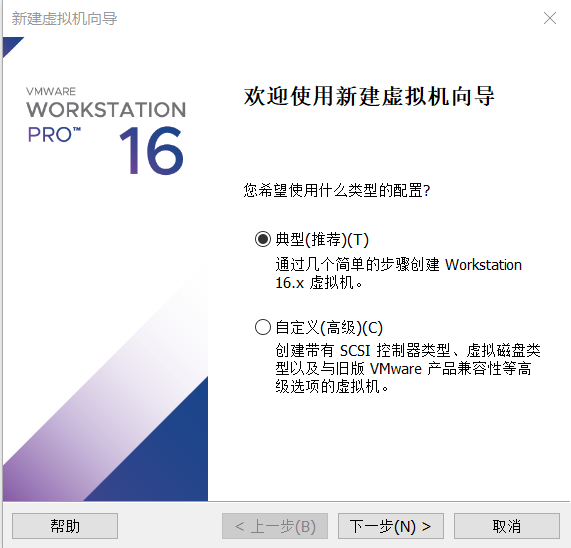
# Linux

学习网：[网站地图 (biancheng.net)](http://c.biancheng.net/sitemap/)

1. **Vm安装**

* **去BIOS里修改设置开启虚机化设备支持（f2，f10）**
* **以管理员身份打开Vm进行安装**

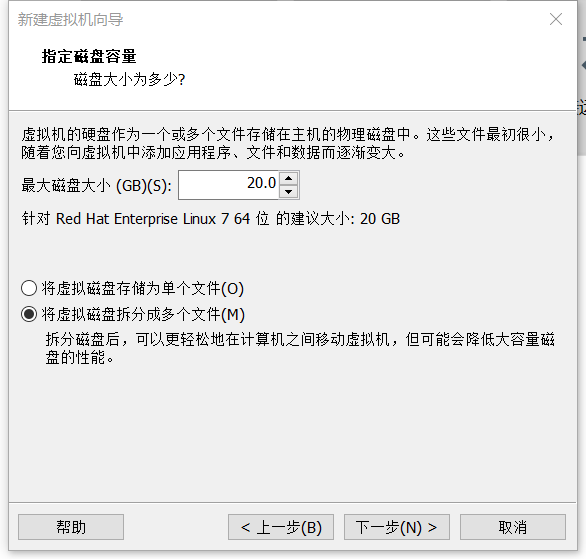
**二、创建虚拟机**

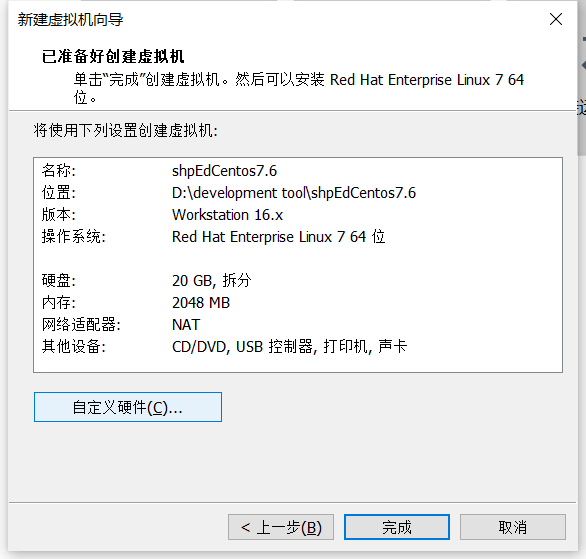
****

****

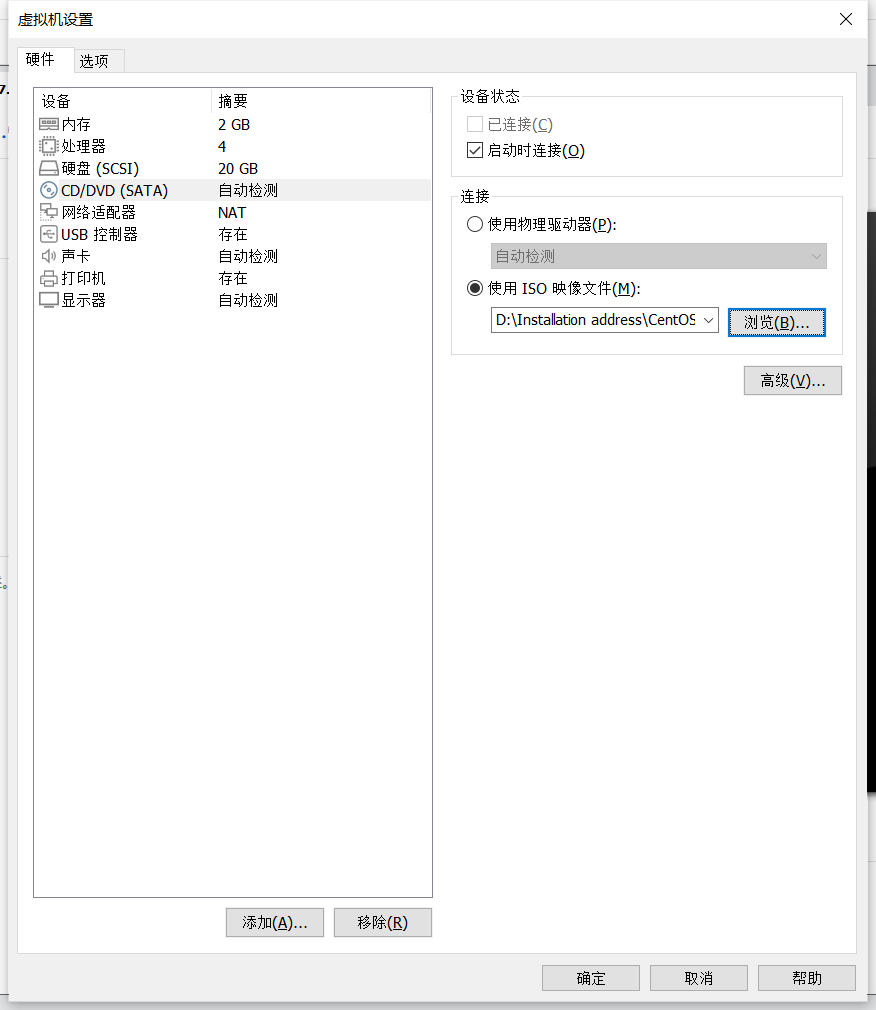
****

****

****



****



****

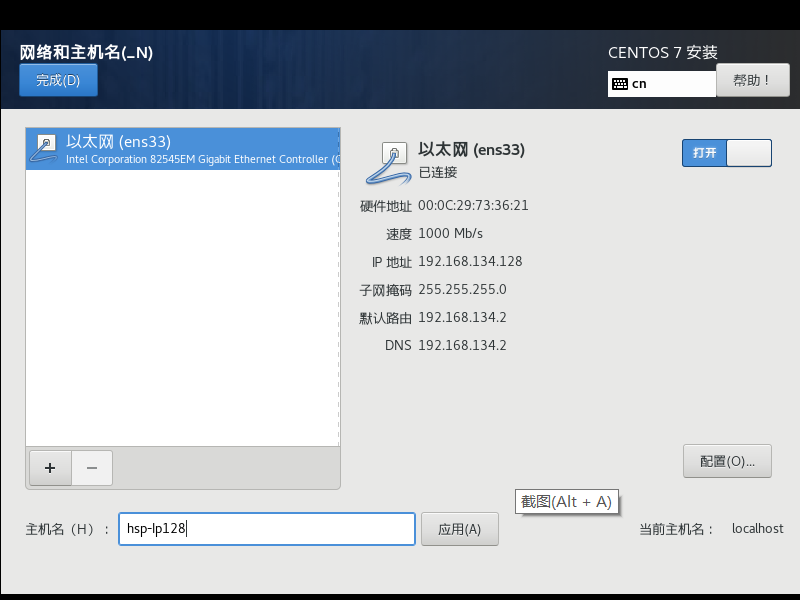
****

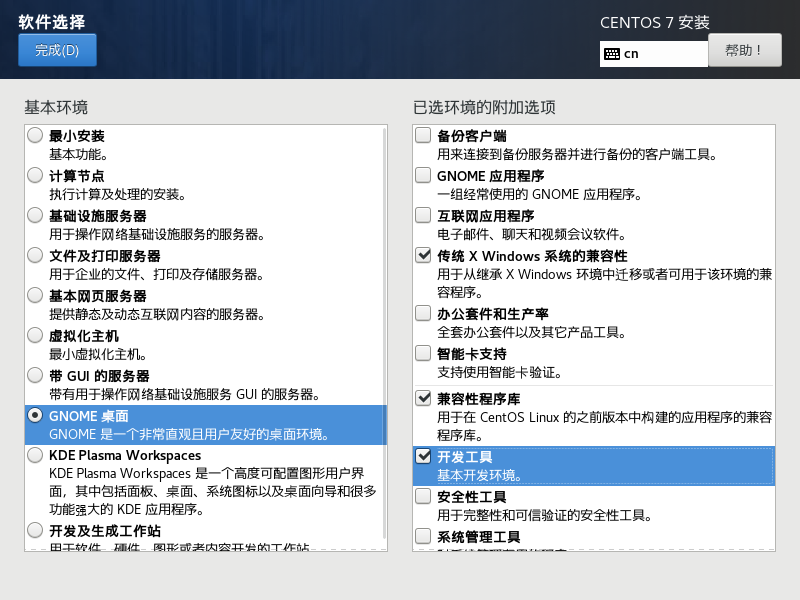
****

****

****

注意：实际工作中可开启，学习中可关闭不占内存



****

**三、网络连接的三种模式**

* **桥接模式：**直接连接物理网络，虚拟系统可以和外部系统通讯，但容易造成IP冲突
* **NAT模式：**用于共享主机的IP地址，网络地址转换模式，虚拟系统可以和外部系统通讯，不造成IP冲突
* **仅主机模式：**与主机共享的专有网络，独立的系统

**四、虚拟机克隆**

* **直接拷贝一份安装好的虚拟机文件**
* **用vmware的克隆操作（克隆时，需关闭linux系统）**

**五、虚拟机快照**

如果你在使用虚拟系统的时候（比如linux），你想回到原先的某一个状态，也就是说你担心可能有些误操作造成系统异常，需要回到原先某个正常运行的状态，vmware也提供了这样的功能，就叫快照管理。

**六、vmtools安装**

Vmtools安装后，可以让我们在windows下更好的管理vm虚拟机

可以设置windows和centos的共享文件夹

* 进行centos
* 点击vm菜单的->install vmware tools
* Centos会出现一个vw的安装包，xx.tar.gz
* 拷贝到/opt
* 使用解压命令tar，得到一个安装文件
* 进入该vm解压的目录，/opt目录下
* 安装./vwware-install.pl
* 全部使用默认设置即可，就可以安装成功
* 注意：安装vmtools需要有gcc

共享文件设置

* 菜单-> vm ->setting，设置选项为always enable，这样就可以读写了
* Window和centos可共享某目录下的文件读写了
* 共享文件在centos的/mnt/hgfs/下

注意：在实际开发中，文件的上传下载是需要使用 远程方式完成的

**七、linux目录结构**

1. /bin [常用] （/usr/bin、 /usr/local/bin）是Binary的缩写，这个目录存放着最经常使用的命令
2. /sbin （/usr/sbin、 /usr/local/sbin）s就是super user的意思，存放的是系统管理员使用的系统管理程序。
3. /home [常用] 存放普通用户的主目录，在linux中每个用户都有一个自己的目录，一般该目录名是以用户的账户命令
4. /root [常用] 该目录为系统管理员，也称作超级权限者的用户主目录
5. /bin 系统开机所需要最基本的动态连接共享库，其作用类似于windowns里的DLL文件，几乎所有的应用程序都需要用到这些共享库
6. /lost + found 这个目录一般情况下是空的，当系统非法关机后，这里就存放一些文件
7. /etc [常用] 所有的系统管理所需要的配置文件和子目录my.config
8. /usr [常用] 这是一个非常重要的目录，用户的很多应用程序和文件都放在这个目录下，类似与windowns下的program files目录
9. /boot [常用] 存放的是启动linux时使用的一些核心文件，包括一些连接文件以及镜像文件
10. /proc [不能动] 这个目录是一个虚拟的目录，它是系统内存的映射，访问这个目录来获取系统信息
11. /srv [不能动] service的缩写，该目录存放一些服务启动之后需要提取的数据
12. /sys [不能动] 这是linux2.6内核的一个很大的变化。该目录下安装了2.6内核中新出现的一个文件系统sysfs
13. /tmp 这个目录是用来存放一些临时文件的
14. /dev 类似于windows的设备管理器，把所有的硬件用文件的形式存储
15. /media [常用] linux系统会自动识别一些设备，例如U盘、光驱等等，当识别后，linux会把识别的设备挂载到这个目录下
16. /mnt [常用] 系统提供该目录是为了让用户临时挂载别的文件系统的，我们可以将
17. 外部的存储挂载在/mnt/上，然后进入该目录就可以查看里的内容了。d:/myshare
18. /opt 这是给主机额外安装软件所摆放的目录，如安装ORACLE数据库就可放到该目录下。默认为空
19. /usr/local [常用] 这是另一个主机额外安装软件所安装的目录。一般是通过编译源码方式安装的程序
20. /var [常用] 这个目录中存放着在不断扩充的东西，习惯将经常被修改的目录放在这个目录下，包括各种日志文件
21. selinux [security-enhacced linux] selinux是一种安全子系统，它能控制程序只能访问特定文件，有三种工作模式，可以自行设置

**八、远程登录**

**为什么需要远程登录：**

* Linux服务器是开发小组共享
* 正式上线的项目是运行在公网

远程登录客户端工具：Xshell6、Xftp6等。

**Linux-Xshell6介绍**：

Xshell是目前最好的远程登录到linux操作的软件，流畅速度并且完美解决了中文乱码的问题，是目前首选的软件

Xshell是一个强大的安全终端模拟软件，它支持SSH1、SSH2以及Microsoft Windows平台的TELNET协议。

Xshell可以在windows界面下用来访问远端不同系统下的服务器，从而比较好的达到远程控制终端的目的

注意：Xshell远程登录必须知道公网ip以两个网络是否是通的，在linux系统中可输入ifconfig查看ip地址，windows下cmd输入**ping ip地址**进行查看两网络是否通

文件的上传下载可安装**Xftp6**

**九、vi和vim**

Linux系统会内置vi文本编辑器（类似windows记事本）

Vim具有程序编辑的能力，可以看做是vi的增强版，可以主动的以字体颜色辨别语法的正确性，方便程序设计。代码补完，编译及错误跳转等方便编程的功能特别丰富，在程序员中被广泛使用。

**vi和vim常用的三种模式：**

1. **正常模式：**

以vim打开一个档案就直接进入一般模式了（这是默认的模式）。在此模式中，你可以使用【上下左右】按键来移动光标，可以使用【删除字符】或【删除整行】来处理档案内容，也可以使用【复制、粘贴】来处理文件数据

1. **插入模式：**

按下i、I、o、O、a、A、r、R等任何一个字母之后才会进入编辑模式，一般来说按i即可

1. **命令行模式：**

在这个模式中，可以提供你相关的指令，完成读取、存盘、替换、离开vim、显示行号等的动作则是在此模式中达成的

**模式切换：**默认模式进入插入模式输入**i**或**a**，插入模式进入命令行模式先输入esc退出再输入**冒号**或**/**

**vi和vim快捷键：**

**命令模式下：**

1. :wq（保存退出）
2. :q（退出）
3. :q!（强制退出不保存）
4. 查找关键子（输入关键字回车，输入n下一个）
5. :set nu 和 :set nonu（是否显示行号）

**默认模式下:**

1. 拷贝当前行**yy**，拷贝当前行向下的5行 **5yy**，并粘贴（输入**p**）
2. 删除当前行**dd**，删除5行 **5dd**
3. **G**和**gg**（定位到末行和首行）
4. **5 + shift + g （**光标定位到第五行）
5. **u**（撤销，类似windows的ctrl + z）

**十、关机重启**

* 1. shutdown -h now（立即进行关机）
  2. shutdown -h 1（1分钟后关机）
  3. shutdown -r now（重启）
  4. halt （关机）
  5. reboot（重启）
  6. sync （把内存的数据同步到磁盘）

注意：不管重启还是关机，首先要运行sync

**十一、登录注销**

1. 登录时尽量少用**root**登录，因为它是系统管理员，最大的权限，避免操作失误。可以用普通用户登录，登录后再用“**su – 用户名**”命令来切换成系统管理员身份

权限高的用户切换到低的用户无需输入密码，反之需要输入密码

1. 在提示符下输入**logout**即可注销用户

**十二、用户管理**

1. **添加用户：**

* useradd 用户名
* useradd -d 指定目录 用户名，给新创建的用户指定家目录（默认在home目录下）

1. **修改用户密码：**

* passwd 用户名（如果不指定用户名默认是给root进行修改）
* pwd （显示当前用户所在的目录）

1. **删除用户：**

* **删除用户保留家目录：**userdel 用户名
* **删除用户不保留家目录：**userdel -r 用户名

1. **查询用户信息：**

* id root（查询管理员）

1. **查询当前用户/登录用户：**

who am i

1. **用户组:**

* **创建组：**groupadd 组名
* **删除组：**groupdel 组名
* **创建用户及所属组：**useradd -g 用户组 用户名
* **修改用户的组：**suermod -g 用户组 用户名

1. **查看所有用户**

**cat /etc/passwd | cut -f 1 -d :**

**十三、指定运行级别**

* **0：**关机
* **1**：单用户（找回丢失密码）
* **2**：多用户状态没有网络服务
* **3**：多用户状态有网络服务
* **4**：系统未使用保留给用户
* **5**：图形界面
* **6**：系统重启

**例子：**init 3

* **查看当前级别：**

systemctl get-default

* **设置默认为3级别：**

systemctl set-default multi-user.target

* **设置默认为5级别**：

systemctl set-default graphical.target

**十四、找回root密码（centos7以后）**

* 重启系统，3之内输入e
* 切换行到linux16行，在utf-8后面输入空格init=/bin/sh
* Ctrl + x进入单用户模式
* mount -o remount,rw /
* passwd (注意：有时需要指定用户名)
* 输入密码
* 确认密码
* touch / .autorelabel (注意：touch与/后面有一个空格)
* exec /sbin/init

****

**十五、帮助指令**

* **man (获得帮助信息)**
* **man ls：**查看ls的帮助信息
* **ls -a：**查看隐藏文件
* **ls -l：**单列输出
* **ls -lh：**单列输出并显示当前总文件大小
* **ls -la：**单列输出并输出隐藏文件（注意：l或不分前后顺序）
* **ls -la /home:** 单列输出home目录内容
* **q：**退出man ls
* **help指令**

**十六、文件目录指令**

* **cd ~**（回到家目录）
* **cd ..**  (回到上一级目录)
* **mkdir /home/test** (在home目录下创建一个test文件夹)
* **mkdir -p /home/test/test2** (在home目录下创建多级目录)
* **touch /home/test.txt** (创建一个空文件)

1. **删除指令：**

* **rmdir /home/test** (删除home下的test空目录，非空删除失败)
* **rm -rf /home/test** (强制删除整个非空目录)
* **rm test.java** （要删除的文件或目录， 会提示是否删除）
* **rm -f test.java** (删除文件不提示)

1. **copy指令：**

* **cp copy的文件 copy到的路径**

例子**：**cp test.txt bb/

* **cp -r 复制的文件 粘贴到的路径**（递归copy，copy整个目录）

例子：cp -r /home/test/(如当前在home目录下，可直接写test) /opt/

注意：如果在当前目录下已存在要copy的文件，会提示是否覆盖，可输入y，表示同意

* **\cp -r** **复制的文件 粘贴到的路径** （强制覆盖提示）

例子：\cp -r /home/test/(如当前在home目录下，可直接写test) /opt/

1. **、mv指令：**

* **mv /home/test.java test1.java** (重命名)

注意：如果test.java和test1.java在同一个目录就叫重命名

* **mv test1.java /root/** (把test1文件移动到root目录下)
* **mv test1.java /root/test.java** （移动并且重命名）
* **mv /home/test ../opt/**  (把home下的test目录整体移动到opt下)

1. **、cat查看指令（只能查看不能修改）：**

* **cat -n product.java**  (查看product文件，并显示行号)
* **cat -n product.java | more （查看文件）**
* **more -n product.java**

more：指令是一个基于vi编辑器的文本过滤器，它以全屏的方式按页显示文本文件内容。

输入向下箭头表示查看下一行；

空白键（**space**）向下翻一页；

**enter**代表向下翻一行；

**q**表示退出more；

**Ctrl + F** 表示向下滚动一屏；

**Ctrl + B** 表示返回上一屏；

**=** 输入当前行的行号；

**:f** 输入文件名和当前行的行号；

1. **、less指令：**

less指令用来分屏查了文件内容，它的功能与more类似，但比more更加强大，支持各种终端显示，less指令显示文件内容时，并不是将整个文件加载之后才显示，而是根据显示的需要加载内容，对于显示**大型的文件**具有较高的效率。

* 空白键：向下翻动一页
* pagedown：向下翻动一页
* pageup：向上翻动一页；
* /字符：向下搜索字符串；n：向下查找； N：向上查找；
* ?字符：向上搜索字符串；n：向下查找； N：向上查找；
* q: 退出less；

1. **、echo指令（输出内容到控制台）：**

* **echo $PATH:** 输出path环境变量；
* **echo $HOSTNAME:** 输出HOSTNAME环境变量；
* **echo “hello”**

1. **、head指令：**

**head**用于显示文件的开头部分内容，默认情况下显示文件的前10行内容

* **head 文件名**
* **head -n 5 文件名：**显示5行并显示行号

1. 、**tail指令：**

**tail**与**head**相反，用于输出文件的末尾内容，默认也显示10行；

* **tail -f product.java：**实时监控追踪文档的所有更新

1. **、箭头指令（>和>>）：**

* **> :** 输出重定向
* **>> :** 追加
* **ls -l > test2.java :** 把列表内容写入test2.java文件中（覆盖写）
* **ls -al >> test2.java:** 把列表的内容追加到test2.java文件中的末尾
* **cat test1.java > test2.java:** 把文件1的内容覆盖到文件2中
* **echo “测试” >> test2.java**

1. **、ln指令：**

软连接也称符号链接，类似于windows里的**快捷键**方式，主要存放了链接其他文件的路径。

* **ln -s /root/ /home/linkRoot:** 在home目录下生成一个root目录的快捷键，名字为linkRoot

1. **history指令：**

查看已经执行过的历史命令

* **history：**查看最近执行的所有命令
* **history** **10：**查看最近执行过的10个指令
* **!5:** 执行曾经执行过的第5个指令

**十七、date指令**

* **date:** 显示当前日期
* **date +%Y:** 显示当前年
* **date +%m:** 显示当前月
* **date +%d:** 显示当前日
* **date “+%Y-%m-%d %H:%M:%S”:** 显示年月日时分秒
* **date -s：**字符串时间(设置当前系统时间)

例子： date -s “2021-09-02 10:58:06”

* **cal:** 显示当前本月日历
* **cal 2021：**显示2021年的所有月日历

**十八、find查找指令**

**find**指令将从指定目录向下递归地遍历其各个子目录，将满足条件的文件或者目录显示在终端。

* **find /home -name test.java:** 在home目录下按名字查找test.java文件
* **find /opt -user nobody:** 按拥有者查找opt目录下，用户名称为nobody的文件
* **find / -size +200M：**查找整个linux目录下大于200M的文件

+n大于；-n小于；n等于

1. **locate指令：**

locate指令可以快速定位文件路径，locate指令利用事先建立的系统中所有文件名称及路径的locate数据库实现快速定位给定的文件。Locate指令无需遍历整个文件系统，查询速度较快。为了保证查询结果的准确度，管理员必须定期更新locate时刻。

**特别说明：由于locate指令基于数据库进行查询，所以第一次运行前，必须使用updatedb指令创建locate数据库。**

* **updated：**创建数据库
* **locate test1.java**

1. **which指令：**

* **which ls：**查看ls指令在哪个目录下

1. **grep指令和管道符号|：**

* **cat test1.java | grep -n “private”：**过滤查找private并显示所在行号
* **grep -n “private” test1.java：**过滤查找private并显示所在行号

-i：忽略字母大小写

**十九、压缩和解压指令**

1. **gzip压缩文件指令：**

* **gzip /home/test/product.java**

1. **gunzip解压文件指令：**

* **gunzip product.java.gz**

1. **zip压缩文件或目录：**

* **zip -r mytest.zip(压缩成) /home/test/**

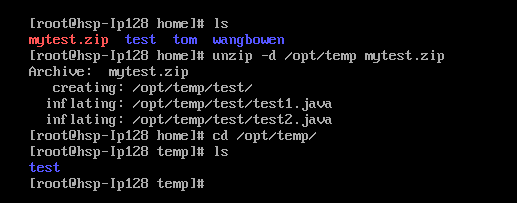
**-r：**递归压缩，即压缩目录



1. **unzip解压文件或目录：**

* **unzip -d 解压到目录 要解压的目录：**

**-d 目录：**指定解压后的文件要存放的目录

****

1. **tar指令：**

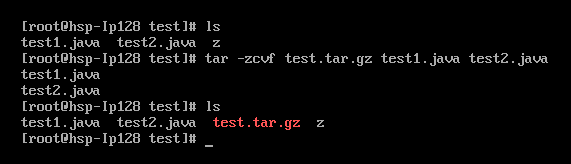
**tar**是打包指令，最后打包后的文件**.tar.gz**的文件。

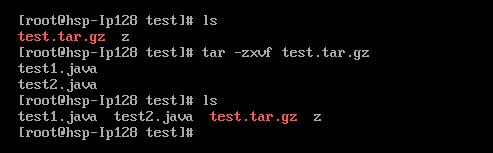
**语法：**

* **tar [option] xxx.tar.gz 打包的内容**
* **tar [option] xxx.tar.gz /home/**

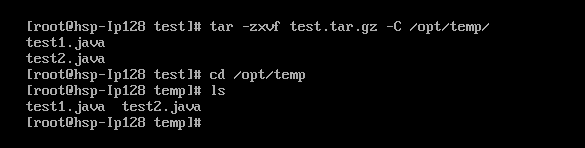
**选项：**

* **-c：** 产生.tar打包文件
* **-v：** 显示详细信息
* **-f：** 指定压缩后的文件名
* **-z：** 打包并压缩
* **-x：** 解压.tar文件

****

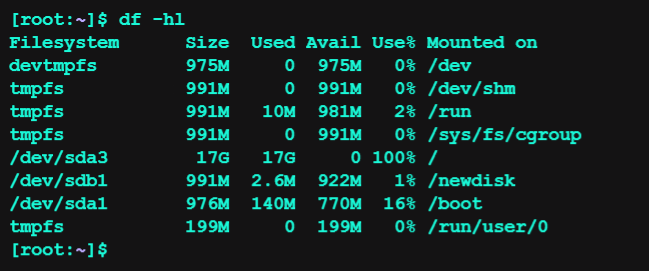
****

**解压到指定目录下：**

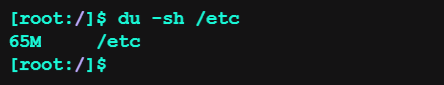
****

**二十、其它相关命令**

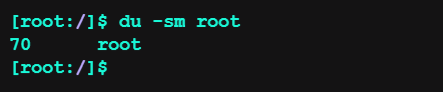
* **查看磁盘空间（df -hl）**

****

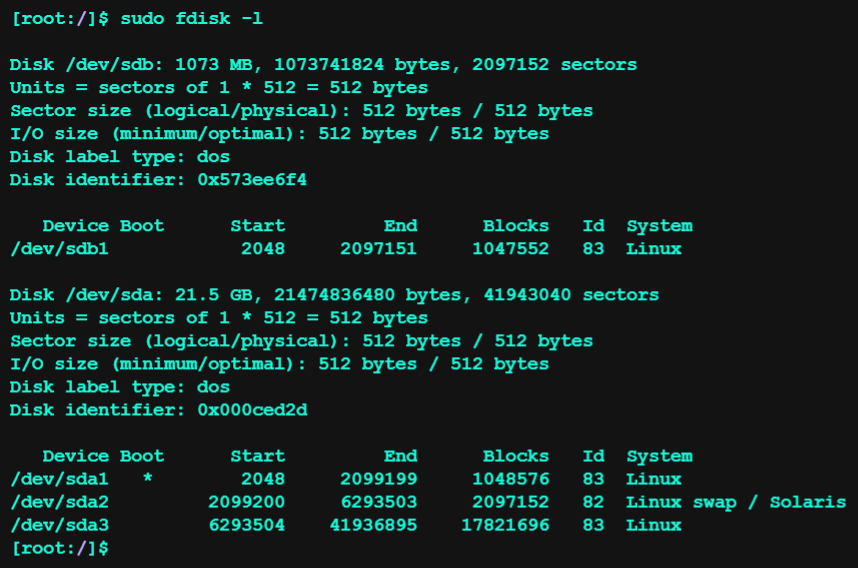
* **查看某个目录大小**

****

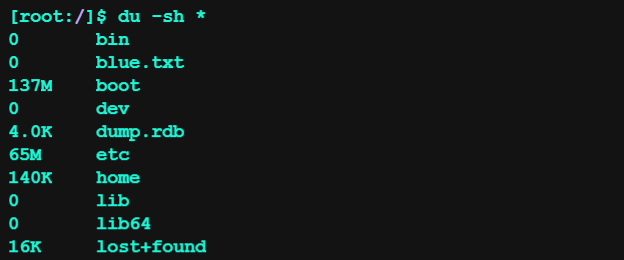
* **查看某个目录包含多少个文件和目录**

****

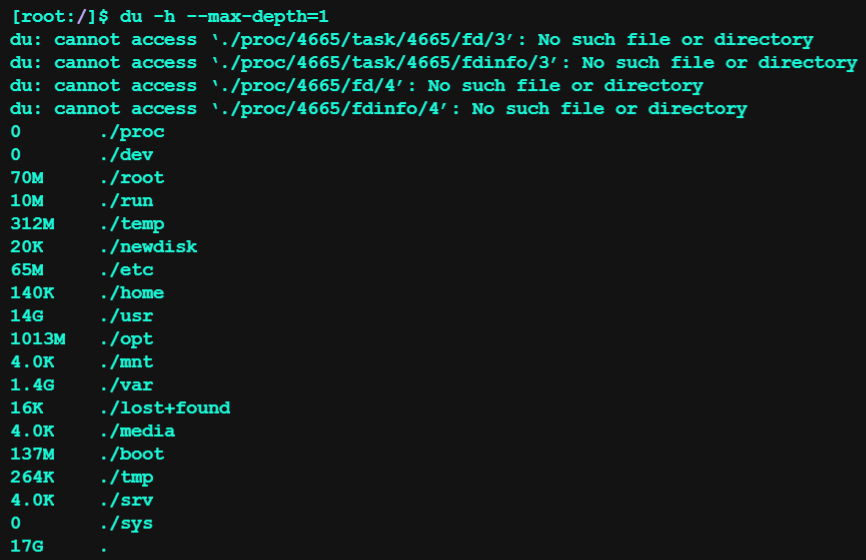
* **查看硬盘的分区**

****

* **当发现全是文件的时候，或者显示不出占用空间最多文件的时候，用**

****

* **查询当前目录下占用空间大的文件或者文件夹**

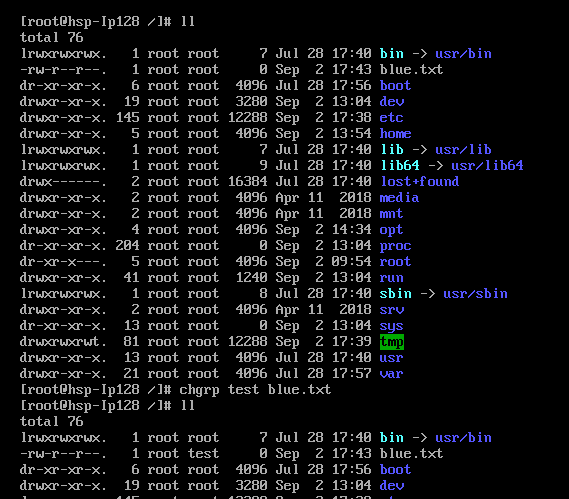


**二十一、文件或目录的所有者**

一般为文件的创建者，谁创建了该文件，就自然的成为该文件的所有者。

* **查看文件的所有者：ls -ahl**

* **修改文件的所有者：chown 用户名 test1.java**
* **修改文件所在组：chgrp 组名 文件名**

****

**二十二、权限**

**-rwxrw-r—**

**1:** 文件：硬链接数或目录: 子目录数

**root：** 用户

**root：**用户组

**1213：**文件大小

**Feb 2 09:11：**最后修改日期

**abc：**文件或目录

**1)、0~9位说明：**

1. **第0位确定文件类型（d、-、l、c、b）**

* **l：**是链接，类似于windows的快捷键方式
* **d：**是目录，类似windows的文件夹
* **c：**字符设备文件，鼠标、键盘
* **b：**是块设备，比如硬盘

1. **第1~3位确定所有者（该文件的所有者）拥有该文件的权限。--User**

* **r：**代表可读【**read**】

**作用到文件：**可以读写查看

**作用到目录：**可以读取，ls查看目录内容

* **w：**代表可写【**write**】，

**作用到文件：**可以修改，但不代表可以删除该文件，删除一个文件的前提条件是对该文件所在目录有写权限，才能删除该文件

**作用到目录：**可以修改，对目录内创建 + 删除 + 重命名目录

* **x：**代表可执行【**execute**】，

**作用到文件：**可以被执行

**作用到目录：**可以进入该目录；cd

1. **第4~6位确定所属组（同用户组的）拥有该文件的权限。--Group**
2. **第7~9位确定其他用户拥有该文件的权限。--Other**

**2）、修改权限：**

**chmod指令:** 可以修改文件或目录的权限

* **+、-、=变更权限**

**语法：**chmod u=rwx，g=rx，o=x 文件或目录

**说明：**

* **u：**所有者
* **g：**所有组
* **o：**其他人
* **a：**所有人（u、g、o的总合）

**案例：**

* **给test.java文件的所有者读写执行的权限，给所在组读执行权限，给其它组读执行权限**

chmod u=rwx,g=rx,o=x test.java

* **给 test.java文件的所有者除去执行权限，增加组写的权限**

chmod u-x,g+w test.java

* **给 test.java文件的所有用户添加读的权限**

chmod u+r test.java

* **通过数字变更权限：**

r = 4、w = 2、x = 1； rwx = 4 + 2 + 1 = 7；

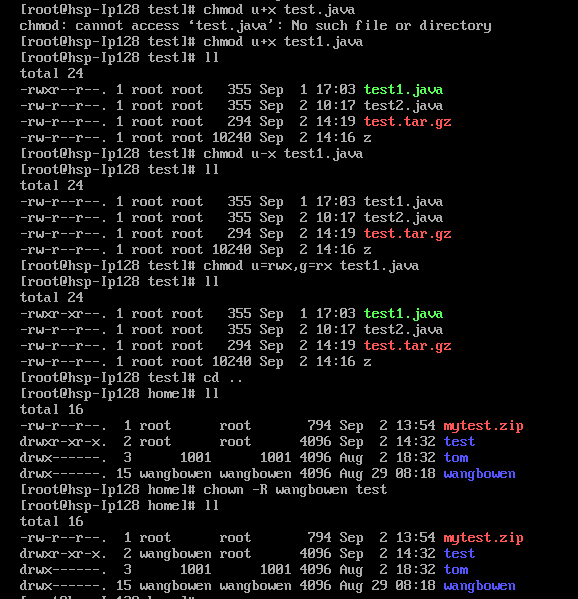
**语法：**

**chmod u=rwx，g=rx，o=x 文件目录名 = chmod 751 文件目录名**

* **修改所有者：**

**文件的所有者：**chown 用户 文件

**目录及目录下所有者：**chown - R 用户名 目录



* **修改文件或目录所在组：**

chgrp 新组名 文件/目录（递归修改加**-R**）；

**二十三、crond任务调度**

**crontab进行定时任务的设置**

任务调度是指系统在某个时间执行的特定的命令或程序。

**任务调度分类：**

* **系统工作：**有些重要的工作必须周而复始地执行。如病毒扫描等
* **个人用户工作：**个人用户可能希望执行某些程序，如mysql数据库的备份。

**语法：**crontab [选项]

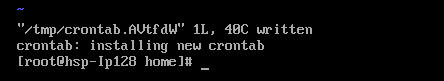
**常用选项：**

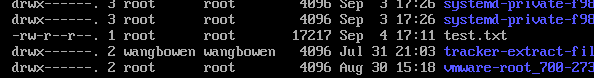
* **-e：**编辑crontab定时任务
* **-l：**查询crontab任务
* **-r：**删除当前用户所有crontab任务

**设置定时任务：**

* **crontab -e**
* **\*/1 \* \* \* \* ls -l /etc/ > /tmp/test.txt**

****

****

****

**说明：**每隔一分钟单列输出etc目录并重定向到tmp目录下的test.txt中

**参数数明：**

* **第1个\*：**

一个小时当中的第几分钟，范围：0~59

* **第2而\*：**

一天当中第几个小时，范围：0~23

* **第3个\*：**

一个月当中的第几天，范围：1~31

* **第4个\*：**
* 一年当中的第几个月, 范围：1~12
* **第5个\*：**

一周当中的星期几，范围：0~7（0和7都代表星期日）

**特殊符号说明：**

* **\*：代表任何时间**
* **，：代表不连续时间**

**例子：**0 8,12,16 \* \* \* 命令

表示每天的8点0分，12点0分，16点0分都执行一次

* **-：代表连续的时间范围**

**例子：**0 5 \* \* 1-6 命令

表示周一到周六的凌晨5点0分执行命令

* **\*/n：代表每隔多久执行一次**

**例子：**\*/10 \* \* \* \* 命令

表示每隔10分钟执行一次

注意：星期几和几号最好不要同时出现，因为它们定义的都是天，非常容易混乱。

**案例：**

* **每隔1分钟将日期和日历都追加到/home/mycal文件中**

**步骤1：**vim /home/mycal.sh

写入date >> /home/mycal.sh和cal >> /home/mycal.sh

**步骤2：**给mycal.sh增加执行权限；chmod u+x mycal.sh

**步骤3：**crontab -e

\*/1 \* \* \* \* /home/mycal.sh

* **每天凌晨2点将mysql数据库testdb，备份到文件中**

**步骤1：**crontab -e;

**步骤2：**0 2 \* \* \* mysqldump -u 用户名 -p 密码 数据库 > /home/db.bak;

**crond其它相关指令：**

* **crontab -r：**终止任务调度
* **crontab -l：**列出当前有哪些任务调度
* **service crond restart：**重启任务调度

**二十四、at定时任务**

**基本介绍：**

* at命令是一次性定时任务，at的守护进程atd会以后台模式运行，检查作业队列来运行。
* 默认情况下，atd守护进程每60秒检查作业队列，有作业时，会检查作业运行时间，如果时间于当前时间匹配，则运行此作业。
* At命令是一次性计划任务，执行完一个任务后不再执行此任务了。
* 在使用at命令的时候，一定要保证atd进程的启动，可以使用相关指令来查看

检测atd是否在运行：**ps -ef | grep atd**

**语法：**at [选项] [时间]；ctrl + d 结束at命令的输入

**选项：**

* **-m：**当指定的任务完成后，将给用户发送邮件，即使没有标准输出
* **-I：**atq的别名
* **-d：**atrm的别名
* **-v：**显示任务将被执行的时间
* **-c：**打印任务的内容到标准输出
* **-V：**显示版本号
* **-q<队列>：**使用指定的队列
* **-f<文件>：**从指定文件读入任务而不是从标准输入读入
* **-t<时间参数>：**以时间参数的形式提交要运行的任务

**时间：**

* 接受在当天**hh:mm**(**小时:分钟**)式的时间指定。假如该时间已过去，那么就放在第二天执行。例：**04:00**
* 使用**midnight(深夜)**、**noon(中午)**、**teatime(饮茶时间，一般下午4点)**
* 采用**12小时**计时制，即在时间后面加上**AM(上午)**、**PM(上午**)；例：12pm
* 指定命令执行具体日期，格式**为month day(月 日)**或**mm/dd/yy(月/日/年)**或**dd.mm.yy(日.月.年)**，指定的日期必须跟在时间的后面。例：04:00 2021-09-04
* 使用相对计时法。格式为**now + count time-units；**

**now**就是当前时间、**time-units**时时间单位

这里能够是**minutes**(分钟)、**hours**(小时)、**days**(天)、**weeks**(星期)。**count**是时间数量，几天，几小时。例：now + 5 minutes

* 直接使用**today**(今天)、**tomorrow**(明天)来指定完成命令的时间。

**案例：**

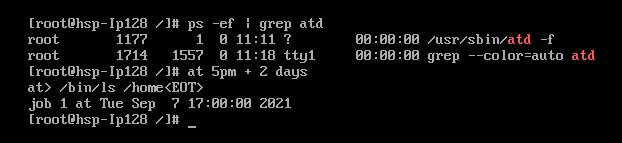
* **2天后的下午5点执行/bin/ls /home**

**第1步：**at 5pm + 2 days

**第2步：**按回车(enter)

**第3步：**输入/bin/ls /home

**第4步：**按两次ctrl + d退出at



* **明天17点钟，输出时间到指定文件中去**

**第1步：**at 5pm tomorrow

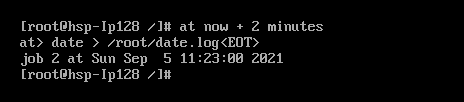
**第2部：**按回车(enter)

**第3部：**输入date > /root/date.log

**第4步：**按两次ctrl + d退出at

* 2分钟后，输入时间到指定文件中去

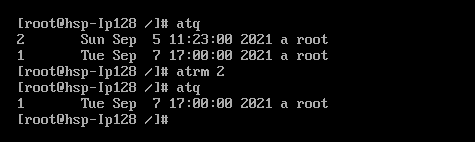
**第一步：**at now + 2 minutes；后面操作同上



* **删除指定编号的任务**

**查看任务队列：**atq

**删除队列：**atrm 2



**二十五、磁盘分区**

**原理介绍：**

* Linux无论有几个分区，分给哪一个目录使用，它归根结底就只有一个根目录。一个独立且唯一的文件结构，linux中每个分区都是用来组成整个文件系统的一部分。
* Linux采用了一种叫“载入”的处理方法，它的整个文件系统中包含了一整套的文件和目录，且将一个分区和一个目录联系起来。这时要载入的一个分区将使它的存储空间在一个目录下获得。

**硬盘说明：**

* Linux硬盘分**IDE**硬盘和**SCSI**硬盘，目前基本上是**SCSI**硬盘
* **IDE**硬盘，驱动器标识符为“**hdx~**”，其中“**hd**”表明分区所在的设备类型，这里是指**IDE**硬盘了。“**x**”为盘号（**a**为基本盘、**b**为基本从属盘、**c**为辅助主盘、**d**为辅助从属盘），“**~**”代表分区，前4个分区用数字1到4表示，它们是主分区或扩展分区，从**5**开始就是逻辑分区。

例如：**had3**表示为第一个IDE硬盘上的第三个主分区或扩展分区，**hdb2**表示为第二个IDE硬盘上的第二个主分区或扩展分区。

* **SCSI**硬盘则标识为“**sdx~**”，**SCSI**硬盘是用“**sd**”来表示分区所在设备的类型的，其余则和IDE硬盘的表示方法一样。

**查看所有设备挂载情况：**

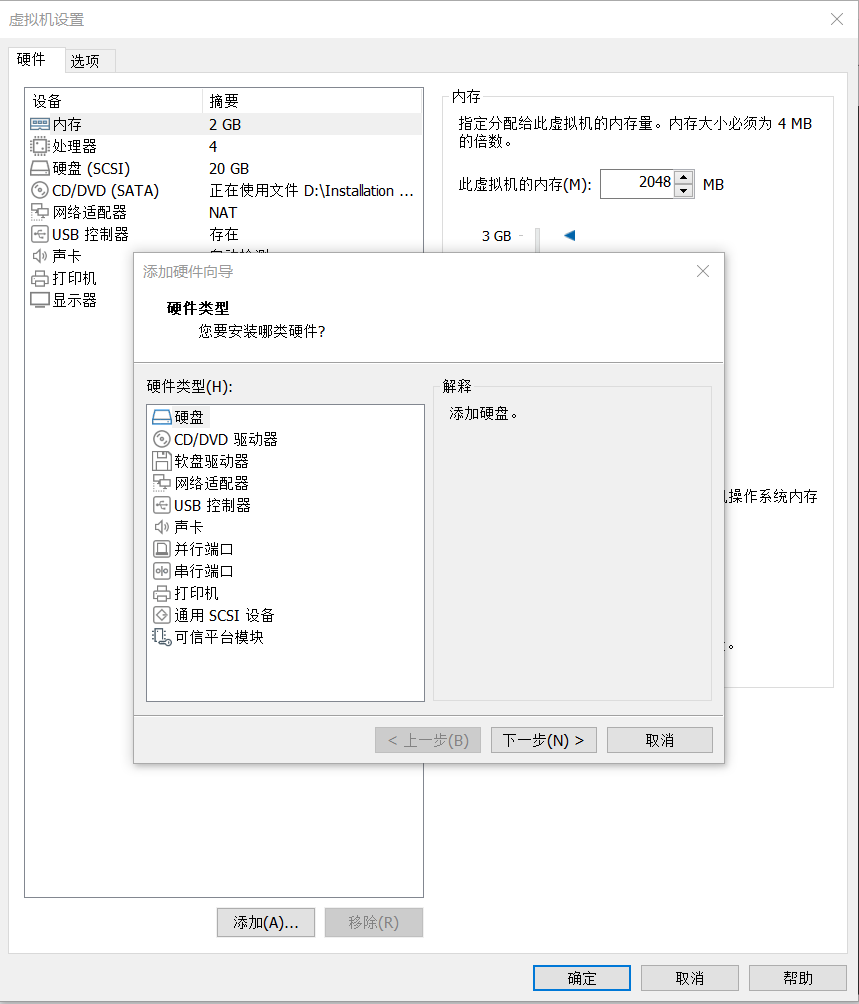
**命令：lsblk或lsblk -f**

****

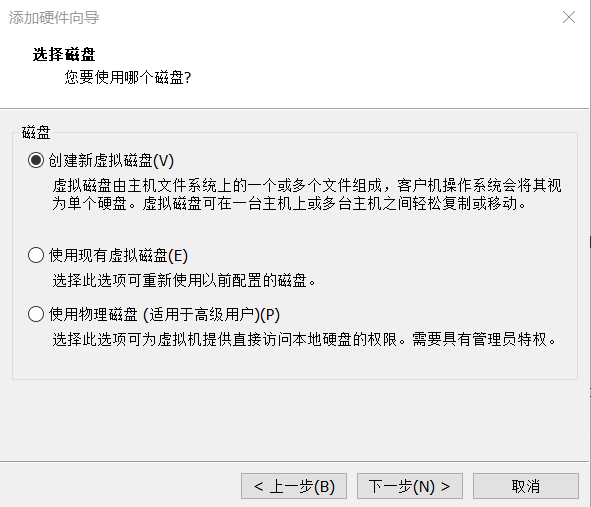
**新增硬盘：**

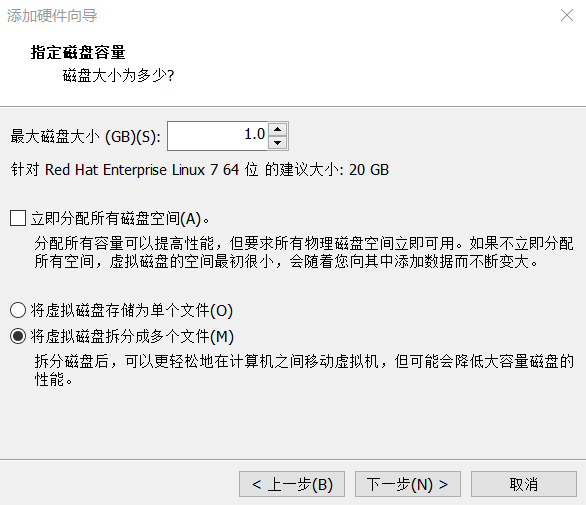
* **虚拟机添加硬盘**

**在【虚拟机】菜单中，选择【设置】**

****

****

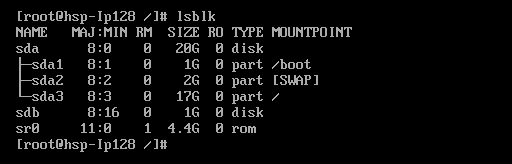
****

****

****

****

注意：需要重启，才能查看到新建的硬盘

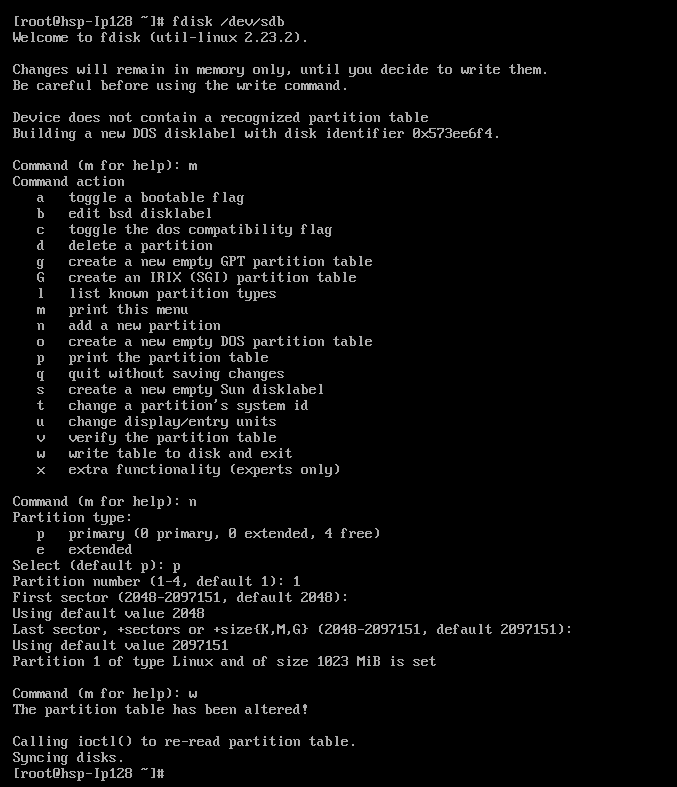


* **分区**

**命令：**fdisk /dev/sdb

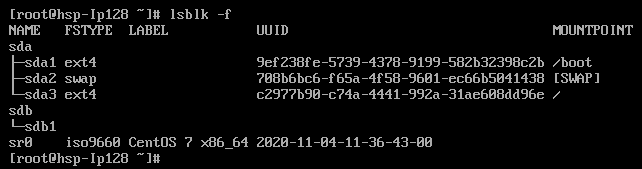
**步骤说明：**

开始分区后输入**n**，新增分区，然后选择**p**，分区类型为主分区。两次回车默认剩余全部空间。最后输入**w**写入分区并退出，若不保存退出输入**p**



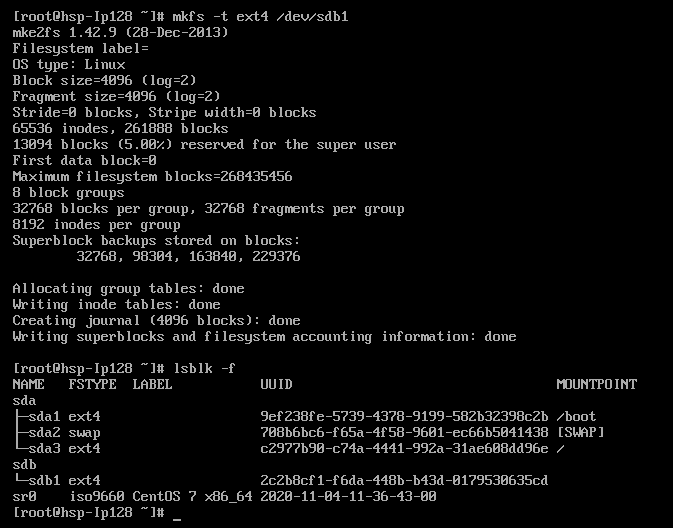
**参数说明：**

* **m：**显示命令列表
* **p：**显示磁盘分区，同fdisk -l
* **n：**新增分区
* **d：**删除分区
* **w：**写入并退出

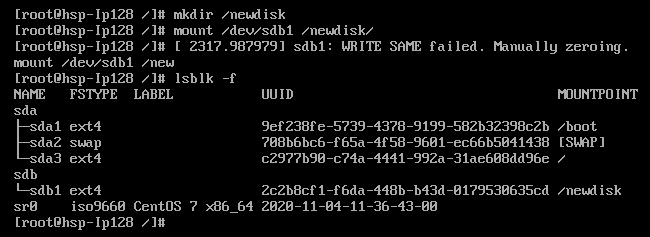


* **格式化磁盘**

**命令：**mkfs -t ext4 /dev/sdb1; 其中**ext4**是分区类型

****

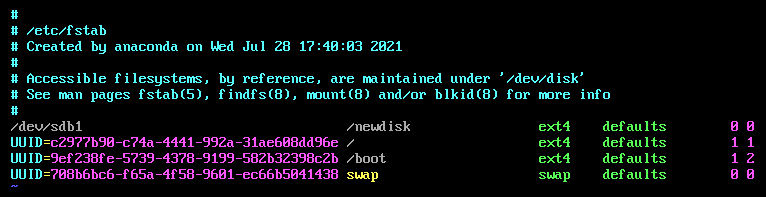
* **挂载**

****

注意：此挂载当重启后挂载点就失效了

* **设置自动永久挂载**

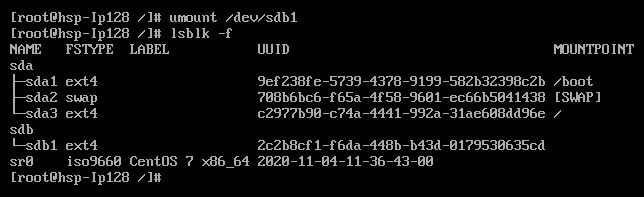
**第一步：**修改**/etc/fstab**实现挂载



**第二步：**执行**mount -a**即刻生效

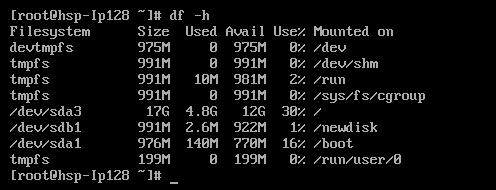
注意：修改完成后记的重启

* **卸载挂载点**

****

**查看磁盘情况：**

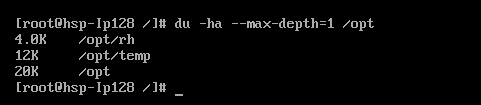
* **查询系统整体磁盘使用情况：**df -h



* **查询指定目录的磁盘占用情况：**du -h 目录

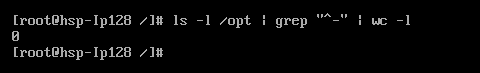
**参数选项：**

* **-s：**指定目录占用大小汇总
* **-h：**带计量单位
* **-a：**含文件
* **--max-depth=1:** 子目录深度
* **-c：**列出明细的同时，增加汇总值



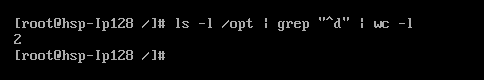
**磁盘使用指令案例：**

* **统计/opt目录下文件的个数**

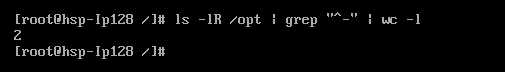
****

**grep**过滤管道； **wc** 统计；

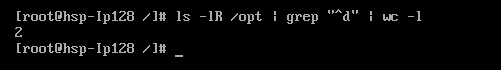
* **统计/opt目录下的目录的个数**

****

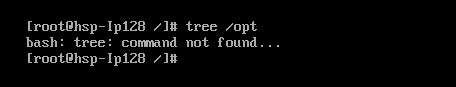
* **统计/opt目录下的文件的个数，包括子文件夹下的**

****

* **统计/opt目录下的目录的个数，包括子文件夹下的**

****

* **以树状模式显示目录结构**

****

注意：如果没有**tree**，则使用**yum install tree**安装即可

**二十六、NAT网络**

* **网络环境配置**

**方式1（自动获取）：**

**说明：**登录后，通过页面设置自动获取ip；

**特点：**linux启动后会自动获取ip，缺点是每次自动获取的ip地址可能不同

**方式2（指定ip）：**

**直接修改配置文件来指定，并可以连接到外网（程序员推荐）**

* **编辑：**vim /etc/sysconfig/network-scripts/ifcfg-ens33

**要求：**将ip地址配置静态的，比如：ip地址为192.168.200.130

**Ifcfg-ens33文件说明：**

* **DEVICE=eth0**

接口名（设备，网关）

* **HWADDR=00:0C:2x:6x:0x:xx**

MAC地址

* **TYPE=Ethernet**

网络类型（通常是Ethernet）

* **UUID=………**

随机id

* **ONBOOT=yes**

系统启动的时候网络接口是否有效（yes/no）

* **BOOTRROTO=static**

Ip的配置方法（**none**-引导时不使用协议、**static**-静态分配ip、**bootp**-BOOTP协议、**dhcp**-DHCP协议）

* **IPADDR=192.168.200.130**

Ip地址

* **GATEWAY=192.168.200.2**

网关

* **DASI=192.168.200.2**

域名解析器





* 保存后，点击编辑找到**VMnet8**，修改ip与以上保存的ip一致；





* **重启网络服务或者重启系统生效**

service network restart 、reboot

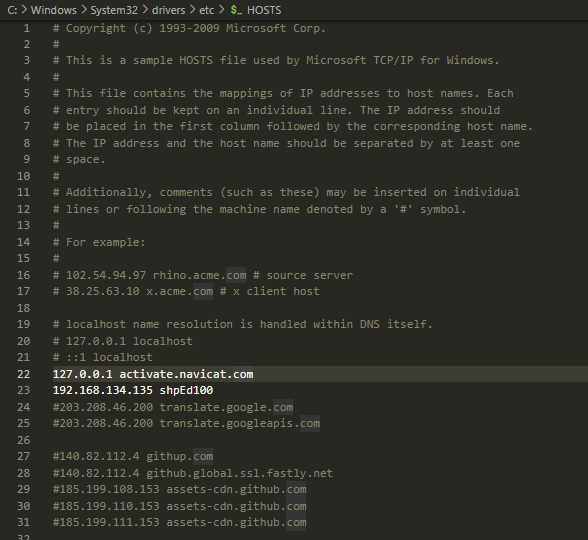
* **主机名和hosts映射**
* **修改主机名：**

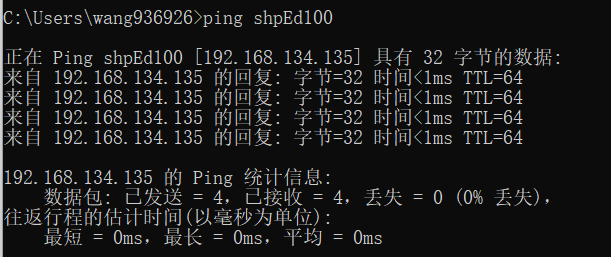
vim /etc/hostname；重启（reboot）

* **设置host映射：**

**Windows下：**

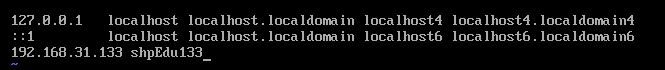
**修改C:\Windows\System32\drivers\etc\HOSTS文件**

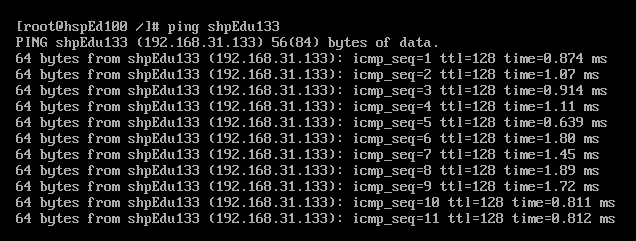
****

****

**Linux下：**

**修改文件：**vim /etc/hosts

****



**主机名解析机制分析(Hosts、DNS)：**

* 当用户在浏览器输入一个域名时，浏览器首先会检查浏览器缓存中有没有该域名解析的**ip**地址，有就先调用这个**IP**完成解析；如果没有，就检查操作系统的**DNS**解析器缓存，如果有直接返回IP完成解析。这两个缓存，可以理解为本地解析器缓存。
* 如果本地解析器缓存没有找到对应的映射，就检查系统中**hosts**文件中有没有配置对应的域名**ip**映射，有则完成解析并返回。
* 如果本地**DNS**解析器缓存和hosts文件中均没有找到对应的**IP**，则到域名服务**DNS**进行解析域。

**DNS域名解析缓存查看：**ipconfig /displaydns

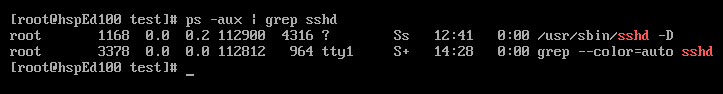
**手动清除DNS缓存：**ipconfig /flushdns

**二十七·、进程**

* **显示系统执行的进程**

**语法：**

* ps 【可选参数】；
* ps -aux | grep sshd（过滤查询）

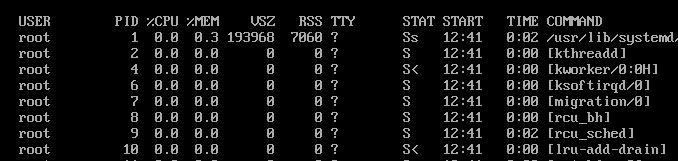


**参数：**

* **-u：**以用户格式显示进程信息
* **-a：**显示当前终端的所有进程信息
* **-x：**显示后台进程运行的参数
* **-ef：**全格式显示当前所有的进程
* **-e：**所有
* **-f：**全格式

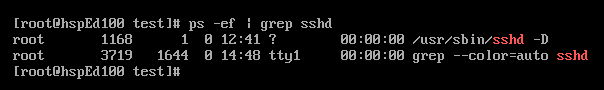
**列项说明：**

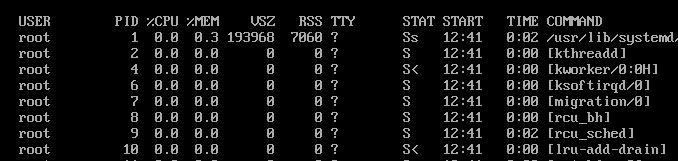
* **PID：**进程号
* **MEM：**占用物理内存的百分比
* **VSZ：**进程占用虚拟内存大小
* **RSS：**进程占用物理内存大小
* **TTY：**终端
* **STAT：**运行状态
* **S：**休眠；
* **s：**表示该进程是会话的先导进程
* **r：**运行中
* **N：**表示进程拥有比普通优先级更底的优先级
* **D：**短期等待
* **Z：**僵死进程
* **T：**被跟踪或被停止
* **START：**进程的启动时间
* **TIME：**占用CPU的总时间
* **COMMAND：**启动进程所用的命令和参数，如果过长会被截断显示

****

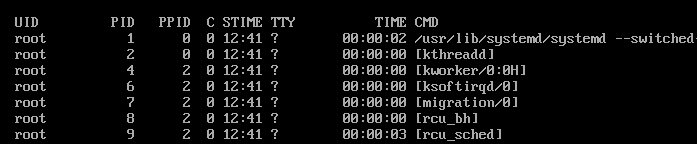
* **父子进程**

**查看父进程：**

****

****

**编号为1168的进程的父进程为1**

****

**编号为1的进程的父进程为0（表示无）**

* **终止进程**

**语法：**

* **kill 【可选项】进程号**

通过进程号杀死或终止进程

* **killall 进程名称**

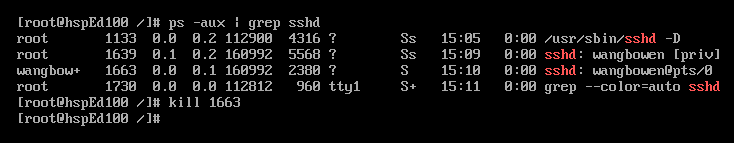
通过进程名称杀死进程，也支持通配符，这在系统因负载过大而变得很慢是很有用

**可选参数：**

* **-9：**强迫进程立即停止

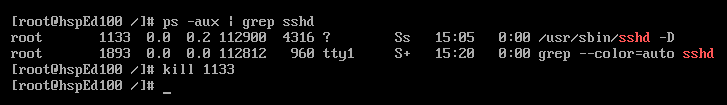
**案例：**

* **终止用户为1663的进程**

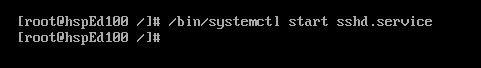
****

* **终止远程登录服务sshd，在适当的时候再次重启sshd服务**

**终止：**

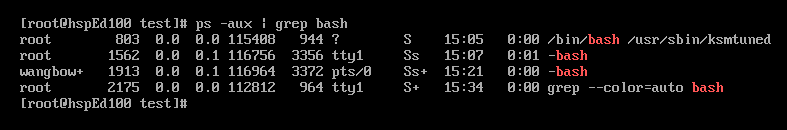
****

**重启：**

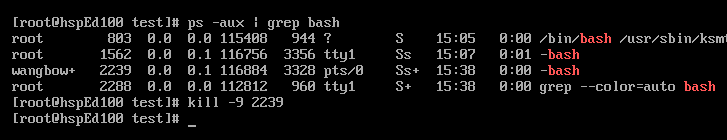
****

* **强制停止某个终端**

**查看启用的终端：**

****

**强制终止：**

****

* **查看进程数pstree**

**语法：**

**pstree 【可选项】**

**参数选项：**

* **-p：**显示进程编号
* **-u：**显示进程的所属用户

* **服务管理**

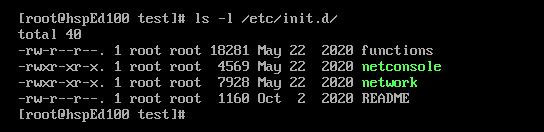
服务本质就是进程，但是是运行在后台的，通常都会监听某个端口，等待其它程序的请求，比如mysql、sshd、防火墙等。因此我们又称为守护进程

**指令：**

* **service 服务名 [start | stop | restart | reload | status]**

注意：centos7.0后很多服务不再使用**service**，而是**systemctl**

**查看当前可用的service：**

****

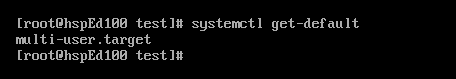
**查看所有服务：**setup



可按**tab**键切换按钮操作项

说明：\*表示当linux启动时，此服务也自动启动。如果要取消此操作可切换到此项按空格键

**查看当前运行级别：**

****

**设置当前运行级别：**

systemctl set-default 级别名

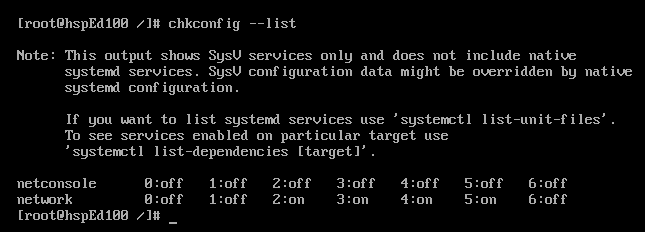
* **chkconfig指令**

通过**chkconfig**命令可以给服务的各个运行级别设置**自启动**或**关闭**

**chkconfig**管理的服务可在**/etc/init.d**中查看

注意：centos7.0后，很多服务使用**systemctl**管理

* **查看chkconfig管理的服务**



* **设置当前服务在某个级别下是否自启动**

chkconfig –level 5（级别）服务名 on/off

* **systemctl指令**

**systemctl**指令管理的服务在**/usr/lib/system/system** 查看

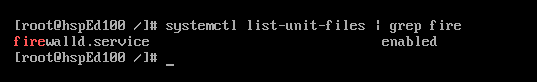
**基础语法：**

systemctl 【start | stop | restart | status】服务名

**设置服务的自启动状态：**

1. **查看服务开机启动状态，grep可进行过滤**

systemctl list-unit-files | grep 服务名



1. **设置服务开机启动**

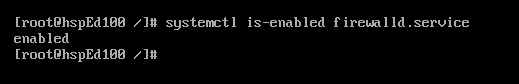
systemctl enable 服务名

1. **关闭服务开机启动**

systemctl disable 服务名

1. **查询某个服务是否是自启动状态**

systemctl is-enabled 服务名



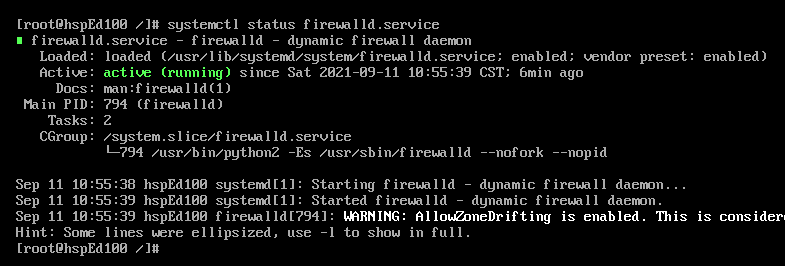
**案例：**

**查看当前防火墙状态，关闭和开启防火墙-firewalld.service**

systemctl status fire;

systemctl stop firewalld;

systemctl start fire;



注意：stop和start设置服务只是临时起效，如要永久生效可使用

systemctl 【enable | disable 】服务名

* **firewall指令**

在真正的生产环境，往需要将防火墙打开，这样外部请求数据包就不能跟服务器监听的端口通讯。这时需要打开指定端口。

* **打开端口**

firewall-cmd –permanent –add-port=端口号/协议

* **关闭端口**

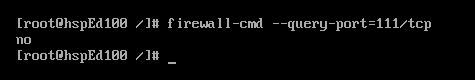
firewall-cmd --permanent -remove-port=端口号/协议

* **重新载入才能生效open/close设置**

firewall-cmd –reload

* **查询端口是否开发**

firewall-cmd --query-port=端口号/协议



* **查看协议**

netstat -anp | more



* **测试111端口是否可以连接上**

telnet 192.168.134.135 111

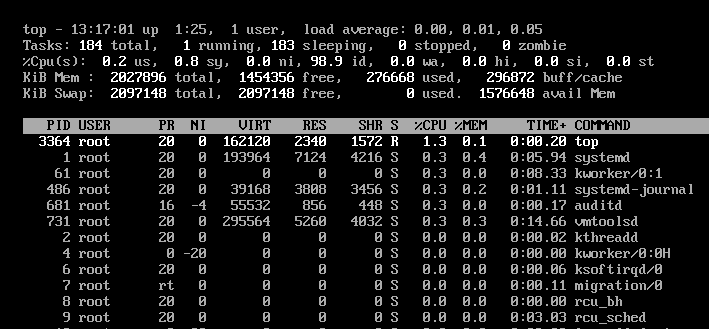
* **动态监控进程**

**top**与**ps**命令很相似，都是用来显示正在执行的进程。不同之处在于**top**在执行一段时间可以更新正在运行的进程。

**基本语法：**top [选项]

**选项说明：**

* **-d：**指定每隔几秒更新（默认3秒）
* **-i：**使top不显示任何闲置或僵死进程
* **-p：**通过指定监控进程ID来仅仅监控某个进程的状态

****

**交互操作说明：**

* **P：**以cpu从大到小排序显示
* **M：**以占用内存从大到小排序显示
* **N：**以进程ID进行排序显示
* **q：**退出top

**案例：**

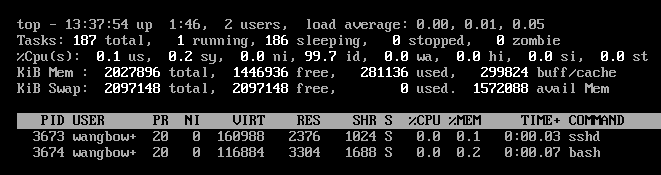
* **指定每隔10秒更新**

top -d 10

* **监控tom用户**

**第一步：**输入top回车

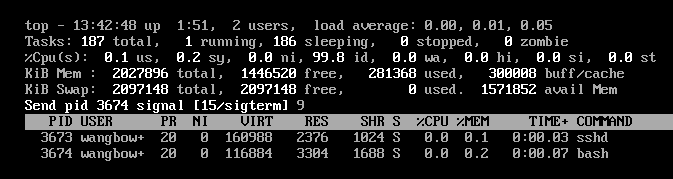
**第二步：**输入**u**，然后再输入用户名回车



* **终止指定进程**

**第一步：**输入**top**回车

**第二不：**输入**k**，在输入进程**ID**回车；输入**9**回车强制终止

****

* **监控网络状态**

**基本语法：netstat 【选项】**

**选项说明：**

* **-an：**按一定顺序排列输出
* **-p：**显示哪个进程在调用

**案例：**

* **查看服务名为sshd的服务信息**

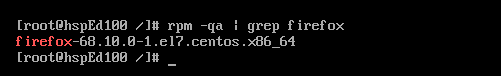
netstat -anp | grep sshd



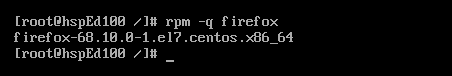
**二十八、rpm包的管理**

**Rpm用于互联网下载包及安装工具·，它包含在某些linux分发版本中。它生成具有.rpm扩展名文件。Rpm是RedHat Package Manager（RedHat软件包管理工具·）的缩写，类似windows的setup.exe**

* **查询已安装的rpm列表**

****

* **查询软件包是否安装**

****

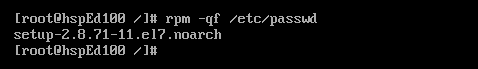
* **查询软件包信息**

****

* **查询软件包包含了哪些文件**

rpm -ql 包名

* **查看某个文件归属于哪个安装包**

****

* **卸载rpm包**

**基础语法：**rpm -e 包名

**强制删除：**rpm -e –nodeeps 包名

* **安装rpm包**

**基本语法：**rpm -ivh rpm包全路径名称

**参数说明：**

* **i**=install 安装
* **v**=verbose 提示
* h=hash 进度条

**二十九、yum**

**Yum**是一个**shell**前端软件包管理器。基于**rpm**包管理，能够从指定的服务器自动下载**rpm**包并且安装，可以自动处理依赖关系，并且一次安装所有依赖的软件包

* **查询yum服务器是否有需要的安装的软件**

yum list | grep 包名

* **安装软件包**

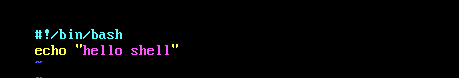
yum install 包名

# Shell脚本

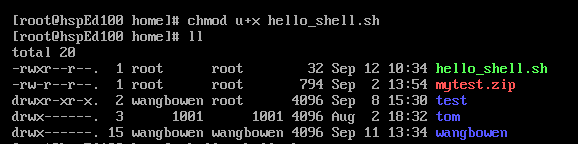
**(一)、快速入门**

**Shell**是一个命令行解释器，它为用户提供了一个向l**inux**内核发送请求以便运行程序的界面系统程序，用户可以用shell来启动、挂起、停止甚至是编写一些程序。

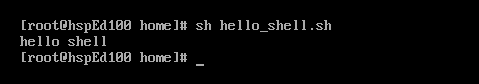
* **脚本格式要求：**
* 以**#!/bin/bash**开头
* 脚本需要有可执行权限



* **脚本执行方式：**
* 赋予脚本**+x**可执行权限



* 不用赋予可执行权限，执行： **sh 脚本名**

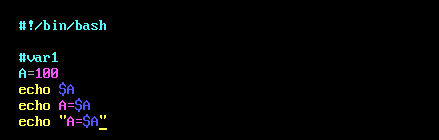
****

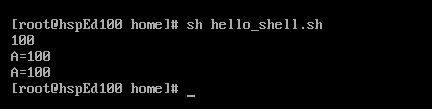
**（二）、变量**

* **系统变量**
* $HOME、$PATH、$PWD等 echo $PWD
* 查看shell所有系统变量：**set**
* **自定义变量**
* **定义规则：**

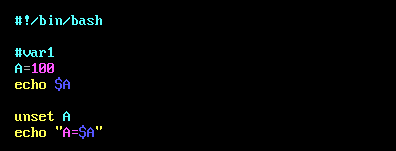
变量名称由字母、数字、下划线组成。但不能以数字开头，等号两侧不能有空格，变量名称一般习惯为大写

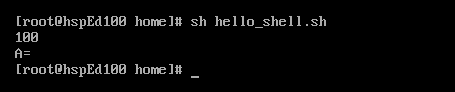
* **定义并输出变量**

****

****

* **撤销变量**

****

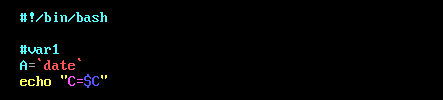
****

* **静态变量**

readonly B=200

注意：静态变量不能unset

* **将指令返回的结果赋值给变量**

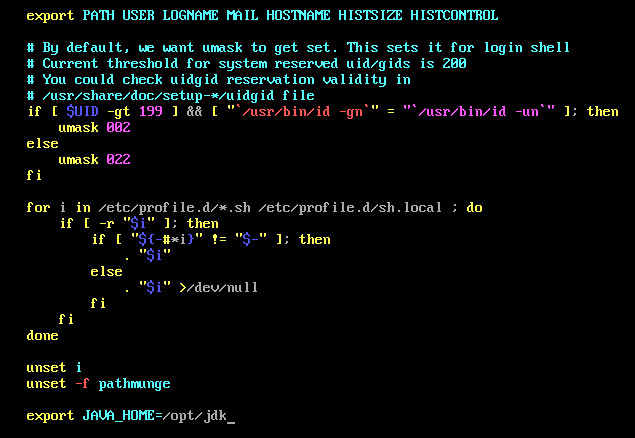




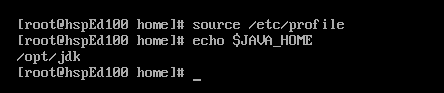
**(三)、设置环境（全局）变量**

* **基础语法**
* **将shell变量输出为环境变量或全局变量**

**在/etc/profile编辑：** export 变量名=变量值



* **让修改后的文件立即生效**

****

source 配置文件

* **shell多行注释**

：<<! 内容 !

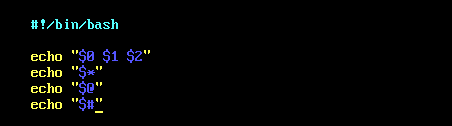
**(四)、位置参数变量**

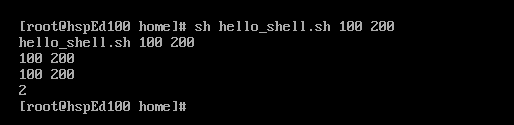
当执行一个**shell**脚本时，如果需要获取命令行中的参数信息，就可以使用位置参数变量比如：**./myshell.sh 100 200,**这就是一个执行shell的命令行，可以在myshell脚本中获取到参数信息

* **基本语法**
* **$n：**

n为数字，**$0**代表命令本身，**$0-$9**代表第一到第九个参数，十以上的参数需要用到括号包含，如**${10}**

* **$\*：**代表命令行中所有的参数，把所有参数看成一个整体
* **$@：**代表命令行中所有的参数，与**$\***不同在于**$@**把每个参数区分对待
* **$#：**参数的个数





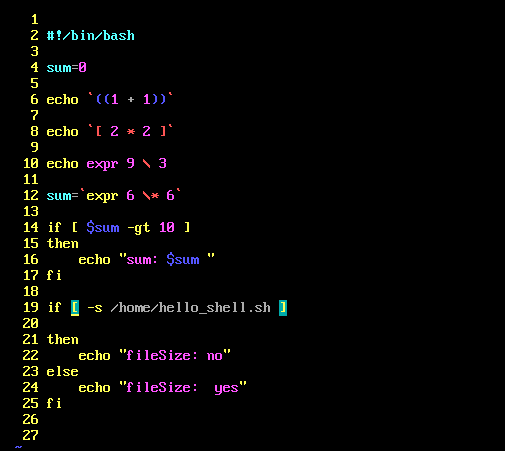
**(五)、预定义变量**

预定义变量就是shell设计者事先已经定义好的变量，可直接在shell脚本中使用

* **$$：**当前进程的**ID**号
* **$!：**后台运行的最后一个进程的进程号（**PID**）
* **$?：**最后一次执行的命令返回状态。如果这个变量的值为**0**，则证明上一个命令正确执行，如果非0，则证明上一个命令执行不正确了

**（六）、运算符**

* **算术运算符**
* **+：**相加
* **-：**相减
* **\*：**乘
* **/：**除
* **%：**取余
* **==：**相等。用于比较数字，相同则返回true
* **!=：**不相等。用于比较数字，不同则返回true
* **关系运算符（只支持数字）**
* **-eq：**相等
* **-ne：**不相等
* **-gt：**大于
* **-lt：**小于
* **-ge：**大于等于
* **-le：**小于等于
* **布尔运算符**
* **!：**非运算，表达式为true则返回false，否则返回true
* **-o：**或运算，有一个表达式为才返回true
* **-a：**与运算，满足所有表达式为才返回true
* **逻辑运算符**
* **&&：**满足所有
* **||：**至少满足一个
* **字符串运算符**
* **=：**赋值
* **!=：**不等于
* **-z：**检测字符串长度是否为0，为0返回true
* **-n：**检测字符串长度是否不为0，不为0则返回true
* **$：**检测字符串是否为空，不为空返回true
* **文件测试运算符**
* **-r：**检测文件是否为可读
* **-w：**检测文件是否为可写
* **-x：**检测文件是否为可执行
* **-b：**检测文件是否是块设备文件，是则返回true
* **-c：**检测文件是否是字符设备文件，是则返回true
* **-e：**检测文件（包括目录）是否存在，存在则返回true
* **-f：**检测文件是否是普通文件（不是目录也不是设备文件），是则返回true
* **-s：**检测文件是否为空（文件大小是否大于0），不为空返回true
* **基本语法**
* **方式一：**$((运算式))
* **方式二：**$[运算式]
* **方式三：·** `expr 运算式`
* **条件判断**



注意：**expr**运算符间要有空格，使用\*符是要在前面加**\(\\*)** 如果希望将某个值赋给某个变量则使用**``**

**（七）、字符串**

字符串可以由单引号' '包围，也可以由双引号" "包围，也可以不用引号。



* **三种形式的区别**

**1) 由单引号' '包围的字符串：**

* 任何字符都会原样输出，在其中使用变量是无效的。
* 字符串中不能出现单引号，即使对单引号进行转义也不行。

**2) 由双引号" "包围的字符串：**

* 如果其中包含了某个变量，那么该变量会被解析（得到该变量的值），而不是原样输出。
* 字符串中可以出现双引号，只要它被转义了就行。

**3) 不被引号包围的字符串：**

* 不被引号包围的字符串中出现变量时也会被解析，这一点和双引号" "包围的字符串一样。
* 字符串中不能出现空格，否则空格后边的字符串会作为其他变量或者命令解析。
* **获取字符串长度**



* **字符串拼接**
* str1=$name$url #中间不能有空格
* str2="$name $url" #如果被双引号包围，那么中间可以有空格
* str3=$name": "$url #中间可以出现别的字符串
* str4="$name: $url" #这样写也可以
* str5="${name}Script: ${url}index.html" #这个时候需要给变量名加上大括号
* **字符串截取**
* **从左边开始截取**

****

* **从右边开始截取**

这里需要强调两点：

* 从左边开始计数时，起始数字是 0（这符合程序员思维）；从右边开始计数时，起始数字是 1（这符合常人思维）。计数方向不同，起始数字也不同。
* 不管从哪边开始计数，截取方向都是从左到右。



* **从指定位置开始截取**

这种截取方式无法指定字符串长度，只能从指定字符（子字符串）截取到字符串末尾。Shell 可以截取指定字符（子字符串）右边的所有字符，也可以截取左边的所有字符。

#### 使用 # 号截取右边字符



#### 2) 使用 % 截取左边字符



注意：\*的位置，因为要截取 chars 左边的字符，而忽略 chars 右边的字符，所以\*应该位于 chars 的右侧。其他方面%和#的用法相同，这里不再赘述，仅举例说明：



* **汇总**

|  |  |
| --- | --- |
| **${string:start :length}** | 从 string 字符串的左边第 **start** 个字符开始，向右截取 **length** 个字符。 |
| **${string: start}** | 从 string 字符串的左边第 **start** 个字符开始截取，直到最后。 |
| **${string: 0-start :length}** | 从 string 字符串的右边第 **start** 个字符开始，向右截取 **length** 个字符。 |
| **${string: 0-start}** | 从 string 字符串的右边第 **start** 个字符开始截取，直到最后。 |
| **${string#\*chars}** | 从 string 字符串第一次出现 **\*chars** 的位置开始，截取 **\*chars** 右边的所有字符。 |
| **${string##\*chars}** | 从 string 字符串最后一次出现 **\*chars** 的位置开始，截取 **\*chars** 右边的所有字符。 |
| **${string%\*chars}** | 从 string 字符串第一次出现 **\*chars** 的位置开始，截取 **\*chars** 左边的所有字符。 |
| **${string%%\*chars}** | 从 string 字符串最后一次出现 **\*chars** 的位置开始，截取 **\*chars** 左边的所有字符。 |

**（八）、数组**

Shell 数组用括号来表示，元素用"空格"符号分割开，语法格式如下：



* **读取数组**

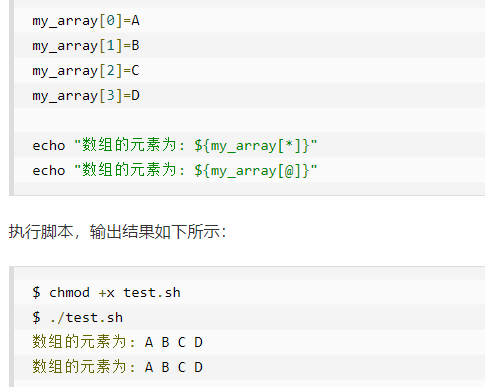
****

* **使用下标定义数组**

****

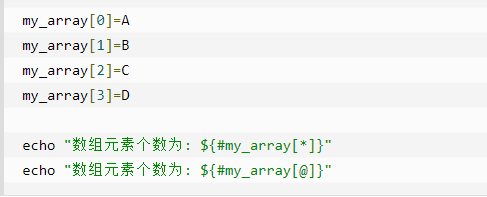
* **获取数组中所有元素**

使用**@** 或 **\*** 可以获取数组中的所有元素，例如：



* **获取数组长度**

获取数组长度的方法与获取字符串长度的方法相同，例如：

****

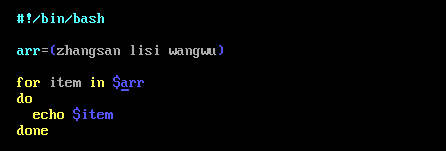
****

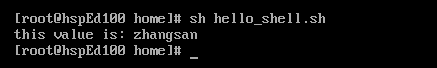
**(九)、数学计算命令**

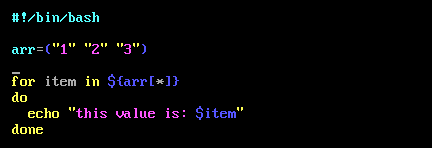
|  |  |
| --- | --- |
| [(( ))](http://c.biancheng.net/view/2480.html) | 用于整数运算，效率很高，**推荐使用**。 |
| [let](http://c.biancheng.net/view/2504.html) | 用于整数运算，和 (()) 类似。 |
| [$[]](http://c.biancheng.net/view/vip_3235.html) | 用于整数运算，不如 (()) 灵活。 |
| [expr](http://c.biancheng.net/view/vip_3236.html) | 可用于整数运算，也可以处理字符串。比较麻烦，需要注意各种细节，不推荐使用。 |
| [bc](http://c.biancheng.net/view/vip_3237.html) | Linux下的一个计算器程序，可以处理整数和小数。Shell 本身只支持整数运算，想计算小数就得使用 bc 这个外部的计算器。 |
| [declare -i](http://c.biancheng.net/view/vip_3238.html) | 将变量定义为整数，然后再进行数学运算时就不会被当做字符串了。功能有限，仅支持最基本的数学运算（加减乘除和取余），不支持逻辑运算、自增自减等，所以在实际开发中很少使用。 |

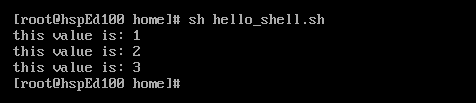
**（十）、循环**

* **For循环**

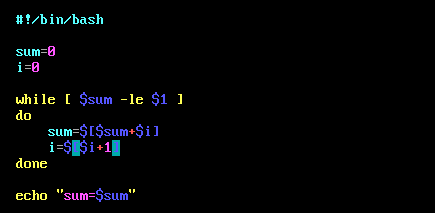
****

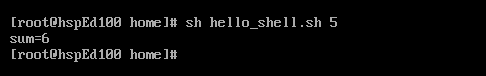
****

****

****

* **While循环**

****

****

**（十一）、read读取控制台输入**

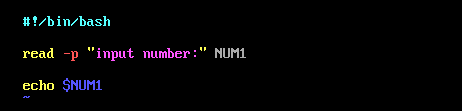
* + - **基本语法**

read 【选项】 【参数】

* + - **选项**
* **-p：**指定读取值时的提示符
* **-t：**指定读取值时的等待时间（秒），如果没有在指定时间输入，就不再等待
  + - **参数**

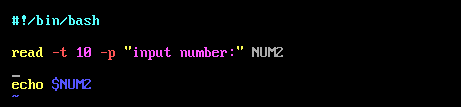
指定读取的变量名

* **案例**
* **读取控制台输入的NUM1值**

****

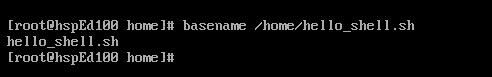
****

* **读取控制台输入的NUM2值，在10秒内输入**

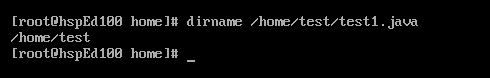
****

**（十二）、函数**

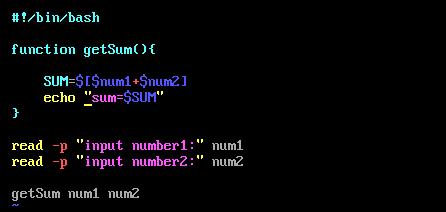
* **系统函数**
* **basename（返回完整路径最后/的部分，常用于获取文件名）：**

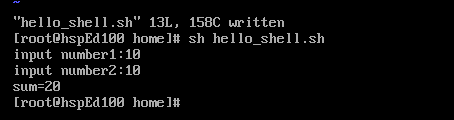
****

* **dirname（返回完整路径最后/的前部分，常用于返回路径部分）：**

****

* **自定义函数**

****

****

# Linux高级篇

**一、系统常用日志**

* /va/log/boot.log：

系统启动日志

* /var/log/cron:

记录与系统定时任务相关的日志

* /var/log/cups/

记录打印信息的日志

* /var/log/dmesg/
* /var/log/btmp

记录错误的登录日志，这个文件是二进制文件，不能使用**vim**查看，而要使用**lastb**命令查看，命令如下：【root@localhost log】#lastb

* /var/log/lasllog:

记录系统中所有用户最后一次登录时间日志。这个文件是二进制文件，要使用**lastlog**命令查看

* /var/log/mailog:

记录邮件信息日志

* /var/log/message:

记录系统重要消息日志，如果系统出现问题，首先要检查的应该就是这个日志文件

* /var/log/secure:

记录验证和授权方面的信息，只要涉及账户和密码的程序都会记录，如：系统登录、**ssh**登录、**su**切换用户、**sudo**授权、甚至添加用户或修改用户密码都会记录在这个文件中

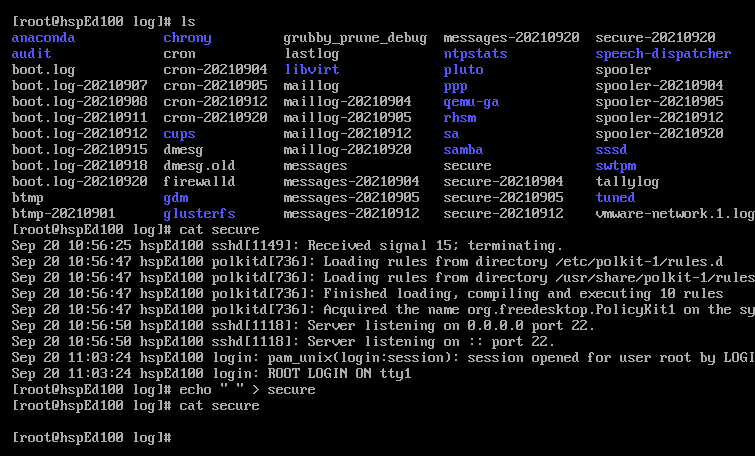
* /var/log/wtmp

永久记录所有用户的登录、注销信息，同时记录系统的启动、重启、关机事件。此文件是二进制文件，需使用**last**命令查看

* /var/log/ulmp:

记录当前已经登录的用户信息，这个文件会随着用户的登录注销而不断变化，只记录当前登录的用户信息。这个文件不能用vim查看，而要使用**w**、**who**、**users**等命令查看

* **echo “ ”> secure：**清空日志



**二、日志管理服务rsyslogd**

* + **查询rsyslogd服务是否启动**

ps aux | grep “rsyslogd”| grep -v “grep”

* + **查询rsyslogd服务的自启动状态**

systemctl list-unit-files | grep rsyslog

* + **配置文件（/etc/rsyslogd.conf）**
* **日志类型**

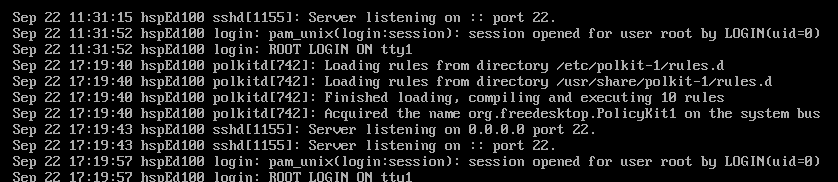
**编辑文件的格式为：\*.\***

**说明：**第一个**\***代表日志类型；第二**\***代表日志级别

* **auth：**pam产生的日志
* **authpriv：**ssh、ftp等登录信息的验证信息
* **corn：**时间任务相关
* **kern：**内核
* **lpr：**打印
* **mail：**邮件
* **mark（syslog）-rsyslog：**服务内部信息，时间标识
* **news：**新闻组
* **user：**用户程序产生的相关信息
* **uucp：**unix to nuix copy主机之间相关的通信
* **local 1-7：**自定义日志设备
* **日志级别**
* **debug：**有调试的信息，日志通信最多
* **info：**一般信息日志，最常用
* **notice：**最具有重要性的普通条件信息
* **warning：**警告级别
* **err：**错误级别，组织某个功能或模块不能正常工作的信息
* **crit：**严重级别，阻止整个系统或整个软件不能正常工作的信息
* **alert：**需要立刻修改的信息
* **emerg：**内核崩溃等重要信息
* **none：**什么都不记录

注意：从上到下，级别从低到高，记录信息越来越少

* **日志文件格式包含**
* 事件产生的事件
* 产生事件的服务器的主机名
* 产生事件的服务名或程序名
* 事件的具体信息

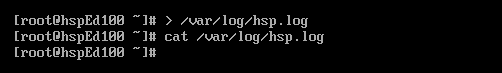


* + **自定义服务**

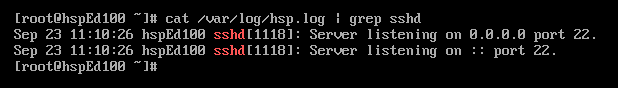
在/etc/rsyslog.conf中添加一个日志文件/var/log/hsp.log,当有事件发送时，该文件会接受到信息并保存



创建hsp.log文件；默认也会自己创建



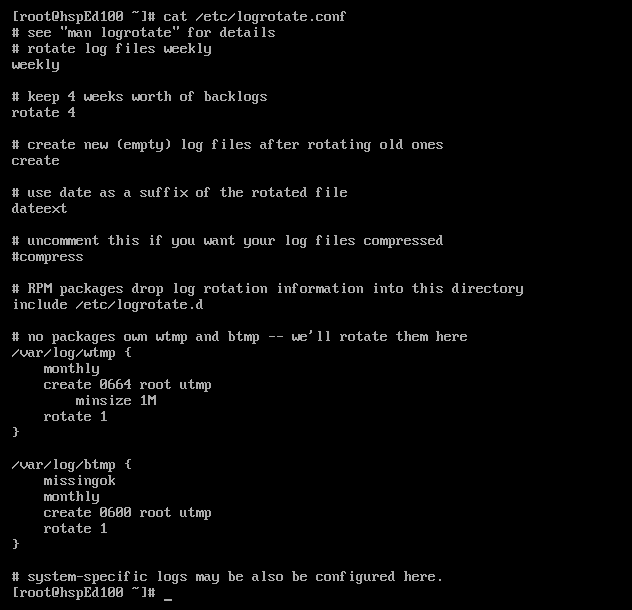
重启再次查看



* + **日志轮替**

日志轮替就是把旧的日志文件移动并改名，同时建立新的空日志文件，当旧日志文件超出保存的范围之后，就会进行删除。

* **日志轮替文件命名**
* Centos7使用**logrotate**进行日志轮替管理，要想改变日志轮替文件名字，通过**/ect/logrotate.conf**配置文件中“**dateext**”参数
* 如果配置文件中**dateext**参数，那么日志会用**日期**来作为日志文件的后缀，这样日志文件名不会重叠，也就不需要更改日志文件名，只需要指定保存日志个数，删除多余的日志文件即可
* 如果配置文件中没有dateext参数，日志文件就需要进行改名。当第一次进行日志轮替时，当前的“**secure**”日志会自动改名为“**secure1**”，然后新建“**secure**”日志，用来保存新的日志。当第二次进行轮替时，“**secure1**”会自动改名为“**secure2**”，当前“**secure**”日志会自动改名为“**secure1**”，然后也会新建“**secure**”日志，用来保存新的日志，以此类推。



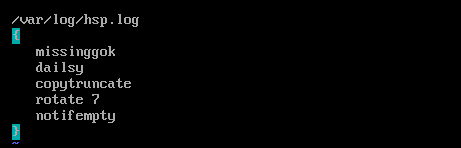
* **Logrotate.conf文件说明**
* **Weekly：**每周对日志进行一次轮替
* **Rotate 4：**共保存4份日志文件，当建立新的日志文件时，旧的将会删除
* **Dateext：**使用日期作为日志轮替文件的后缀
* **Include /etc/logrotate.d：**日志文件是否压缩，如果取消注释，则日志会在转储的同时进行压缩。

**包含/etc/logrotate.d目录中所有的子配置文件。也就是说会把这个目录的所有子配置文件读取出来，**

* **/var/log/wtmp{}：**自定义配置

说明：也可以把某个日志文件的轮替规则，写到/etc/logrotate.d目录下，

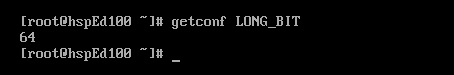
* **参数说明**
* **daily：**轮替周期时每天
* **weekly：**轮替周期是每周
* **monthly：**轮替周期每月
* **rotate** 数字：保留的文件个数，0指没有备份
* **compress：**轮替时，旧的日志进行压缩
* **create mode owner group：**建立新日志，同时指定新日志的权限与所有者和所属组
* **mail address：**轮替时，输出内容通过邮件发送到指定的邮件地址
* **missingok：**如果日志不存在，则忽略该日志的警告信息
* **notifempty：**如果日子为空文件，则不进行日志轮替
* **minsize** 大小：日志轮替的最小值，也就时日志一定要达到这个最小值才是轮替，否则就是时间到达也不轮替
* **size 大小：**日志只有大于指定大小才进行日志轮替，而不是按照时间轮替
* **dateext：**使用日期作为轮替文件的后缀
* **sharedscripts：**在此关键字之后脚本只执行一次
* **prerotate/endscript：**在此关键字之后脚本只执行一次
* **postrotate/endscript：**在日志轮替之后执行脚本命令
* **把自定义日志加入到轮替**

****

* + **内存日志**

****

**查看系统是多少位**

****

**查看系统版本**

**cat /etc/redhat-release**

**查看防火墙状态**

**systemctl status firewalld**

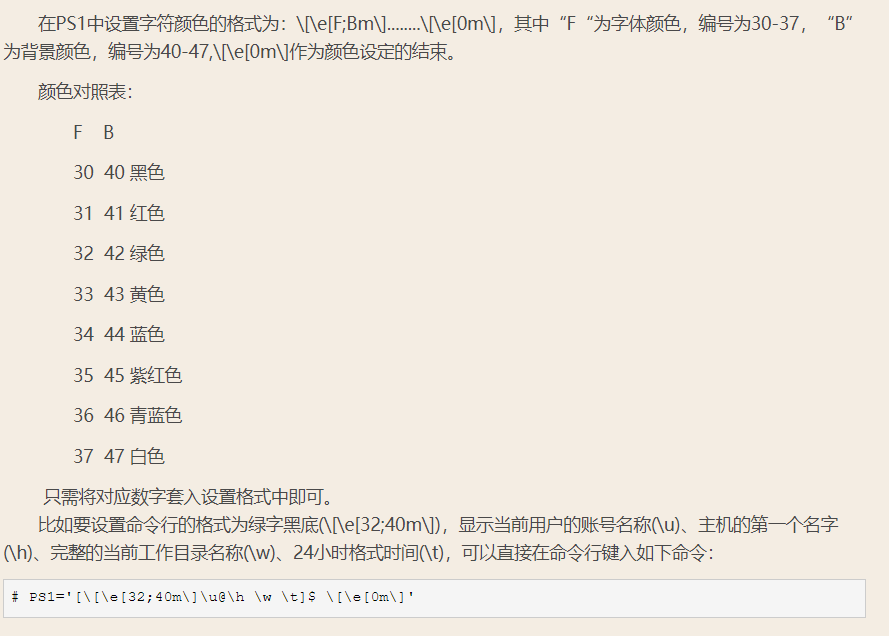
**关闭防火墙**

**systemctl stop firewalld**

**检查是否安装某个软件**

rpm -qa | grep java

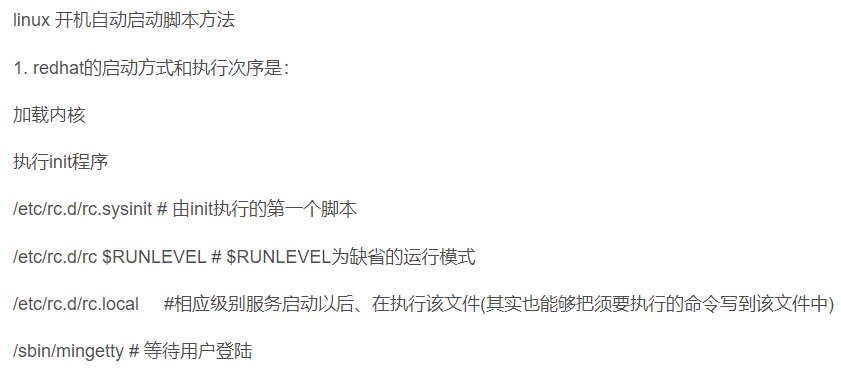
1. **修改linux字体颜色及其它配置**

****

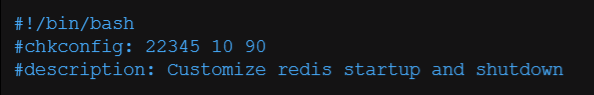
* **cd**
* **ls -la**
* **vim .bashrc**
* **保存并刷新**

source .bashrc

**自定义服务并设置开机自启动**

****

* **方式一（/etc/init.d）**
* 启动时间长，init是串行启动，只有前一个进程启动完，才会启动下一个进程
* 启动脚本复杂，Init进程只是执行启动脚本，不管其他事情，脚本需要自己处理各种情况，这往往使得脚本变得很长
* 由Linux内核加载运行，位于 /sbin/init   ,是系统中第一个进程，PID永远为1
* **新建文件并在文件开头设置以下标签，否则无法添加到chkconfig中**



* **执行：chkconfig –add 服务名**
* **查看是否添加成功：chkconfig –list | grep 服务名**
* **重启测验**

发现并没有自动启动脚本

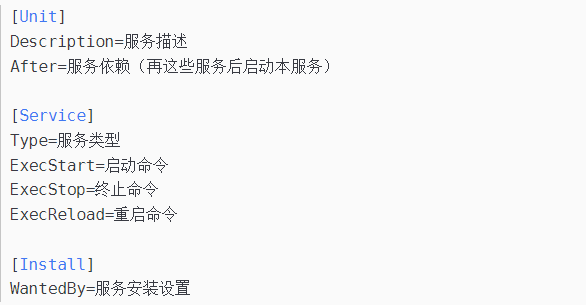
* **方式二（/etc/rc.local）**

**rc.local**在相应级别服务启动以后、才执行该文件；

* 需要在**/etc/rc.local**文件中追加需要执行的脚本文件路径，然后设置**rc.local**具有可执行权限

****

* **重启测试成功**
* **方式三 (/etc/system/system)**
* 按需启动服务，减少系统资源消耗。
* 尽可能并行启动进程，减少系统启动等待时间
* 由Linx内核加载运行，位于 /usr/lib/systemd/systemd  ，是系统中第一个进程，PID永远为1
* **新建service文件，并声明以下参数**

****

****