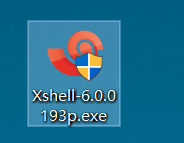
# 远程登录与文件上传

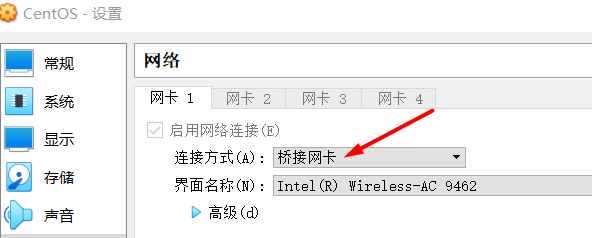
## 远程登录

在实际生产环境中，Linux操作系统并不是在程序员的电脑上的，而是在公司的机房或者在云上，而此时程序员如果想要操作Linux，就需要使用自己的电脑通过远程登录软件连接到Linux，然后进行操作，远程连接到Linux后，都是以命令行的方式对Linux进行操作的；

用于远程登录的软件有很多，用的最多的是XShell软件，XShell是目前最好的远程登录到Linux操作的软件，速度流畅并且完美解决中文乱码问题，是远程登录的首选软件；可以从该网址<https://xshell.en.softonic.com/>下载；



1. 如果Linux系统所在虚拟机软件与是安装在客户端（即需要使用远程连接Linux的电脑）上的，则设置网络为桥接模式，如下：



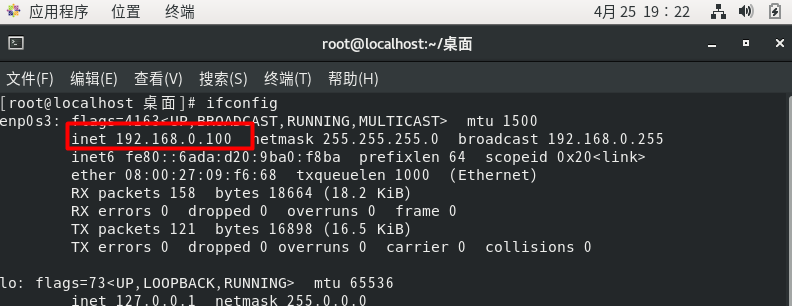
1. 启动Linux系统，在设置中打开网络（没有网络是无法远程登录的），如下：



并设置网络自动连接：



1. 打开终端，使用ifconfig指令查看ip地址



1. 在主机上打开XShell，创建连接：

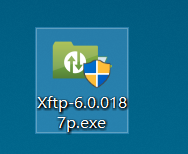


创建连接完成后，点击确定或连接即可根据IP连接到对应的Linux系统；

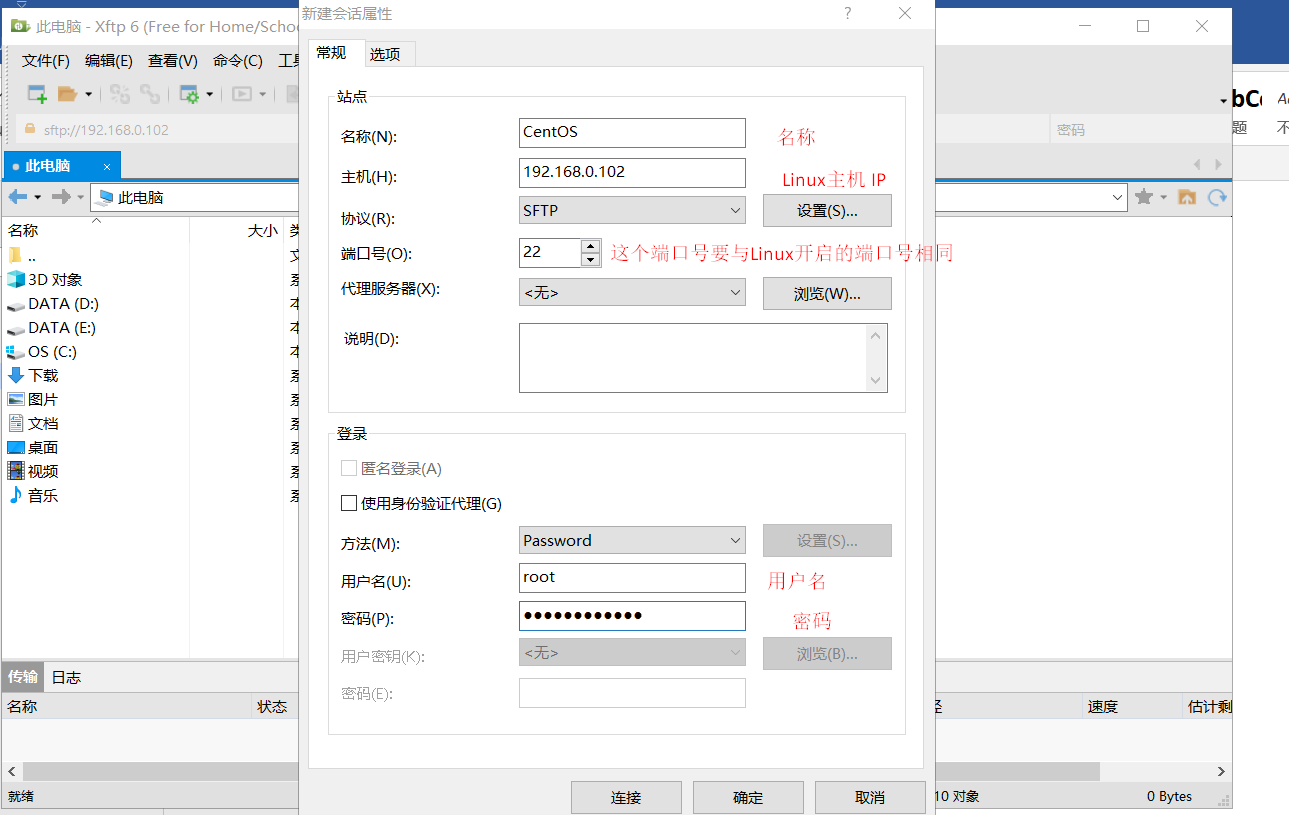
## 远程文件上传

在远程连接到Linux系统后，如果程序员需要将自己本机上的某个文件，如MySQL安装到Linux中，此时，就需要先将安装文件上传到Linux中，然后进行安装操作；

实现远程上传与下载文件可以使用Xftf软件，Xftp是一个基于window平台的强大的SFTP、FTP文件传输软件，通过它能够安全地在Linux和Windows PC之间传输文件，可以从<https://www.netsarang.com/zh/xftp/>下载，也可以在网上下载破解版；

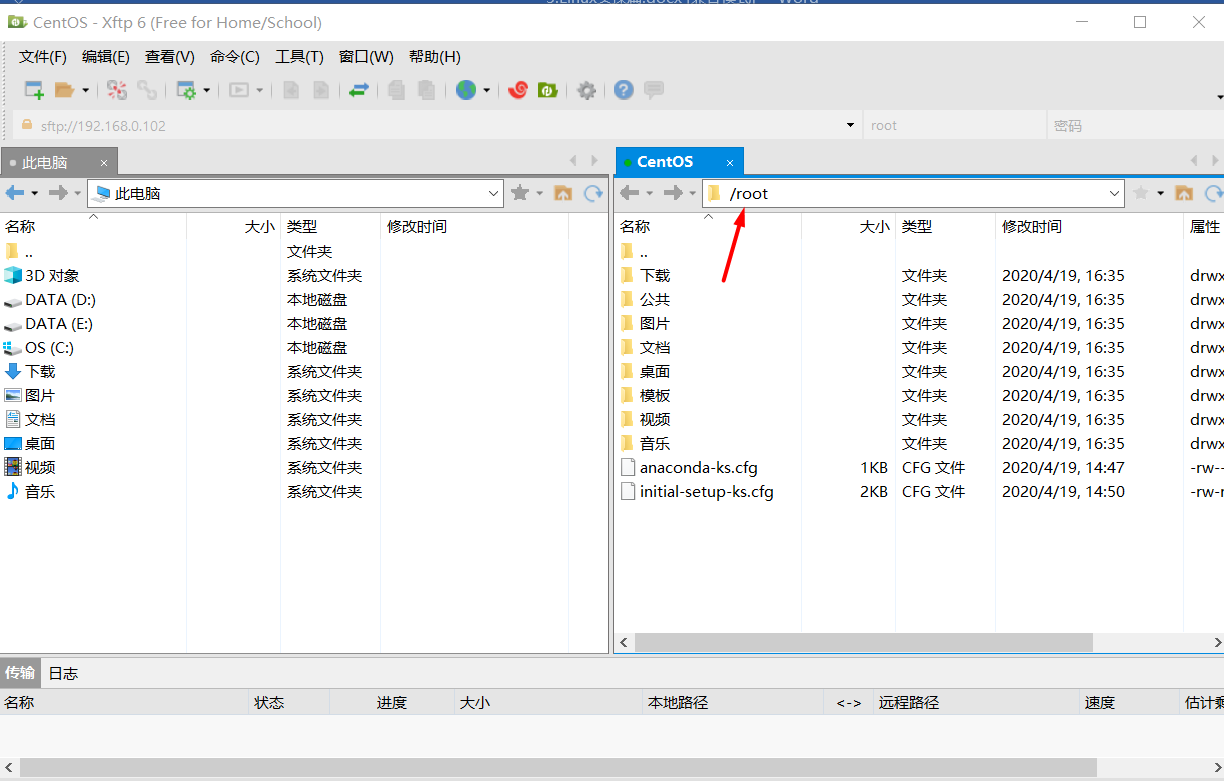


1. 安装该软件的过程很简单，只需下一步即可。安装完成后，运行它，如下：



点击【文件】🡪下的【新建】创建一个连接，填写完信息后，点击连接即可。

1. 连接完成后，如下：



此时，会在右边显示你所使用的用户登录上去的目录，比如使用root用户登录，则会显示root用户目录，你也可以返回到根目录；

自此，Windows和Linux就可以进行文件互传操作；

# Vi/Vim编辑器的使用

## 认识Vim编辑器

1. 所有的Linux系统都会有Vi文本编辑器，而Vim文本编辑器具有程序编译的能力，可以理解为Vi文本编辑器的增强版，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，编辑及错误跳转等方面编程的非常丰富，被程序员广泛使用；

2）要使用Vim编辑器需要经历三个模式，如下：

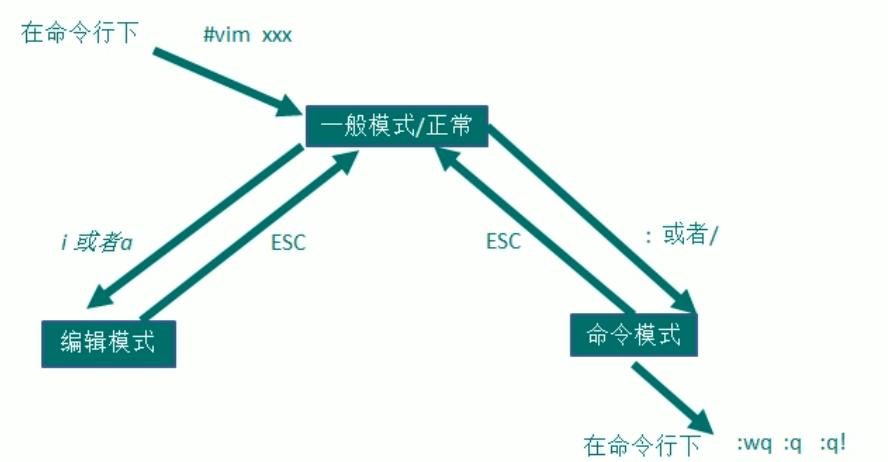
·正常模式：通过vim命令进入正常模式，进入正常模式后，你可以使用方向键

移动光标，或者完成复制粘贴、删除等操作，但不能进行写操作；

·编辑模式：进入正常模式后，可以按“i”键进入编辑模式，在插入模式中，相 比与正常模式，你还进行写操作；

·命令模式：在这个模式中，可以保存在编辑模式中进行的操作；

3）使用Vim编辑器势必需要在这3种模式之间进行切换。如下：



打开终端，默认就是命令模式，在命令模式下可以执行一些Linux命令，如果对某个文本文件执行vim命令或者使用vim命令创建文本文件则进入正常模式，在正常模式你可以对该文本文件执行一些删、复制粘贴等操作，然后使用:wq（保存文件并退出正常模式）、:q（退出不保存文件，如果对文件进行了修改则会执行失败）、:ql（强制退出，不会保存对文件的修改）等命令来决定是否需要保存此文件；你也可以按i键进入编辑模式，在编辑模式中，你可以执行写、删、复制粘贴操作，对文本文件执行这些操作后，如果需要对其进行保存或不保存，首先需要切换到命令模式，通过按esc键实现，在命令模式下，可以执行:wq命令保存该文件并退出、或执行:q命令不保存该文件并退出（如果做了修改则会执行失败）、或执行:q!命令强制不保存所做的修改，并退出；

1. 执行vim命令进入正常模式后，或者从编辑模式按Esc键进入命令模式后，可以执行一些命令，如下：

:wq：表示保存当前修改，并退出文件；

:q：表示退出当前文件，但如果对文件进行了修改，则会执行失败；

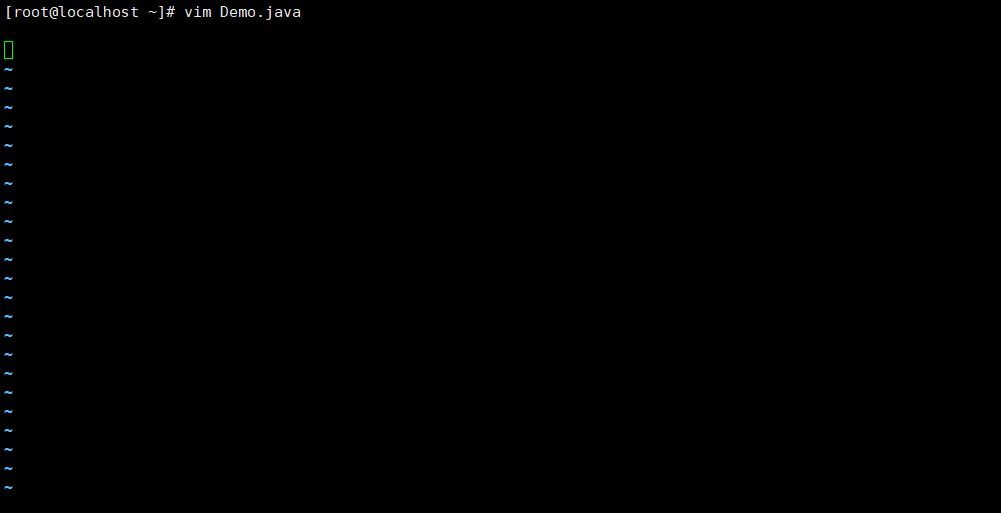
:q!：表示强制退出当前文件，如果对文件进行了修改，则不会保存（一般:q执行

失败后执行此命令）

当然，这只是常用的；

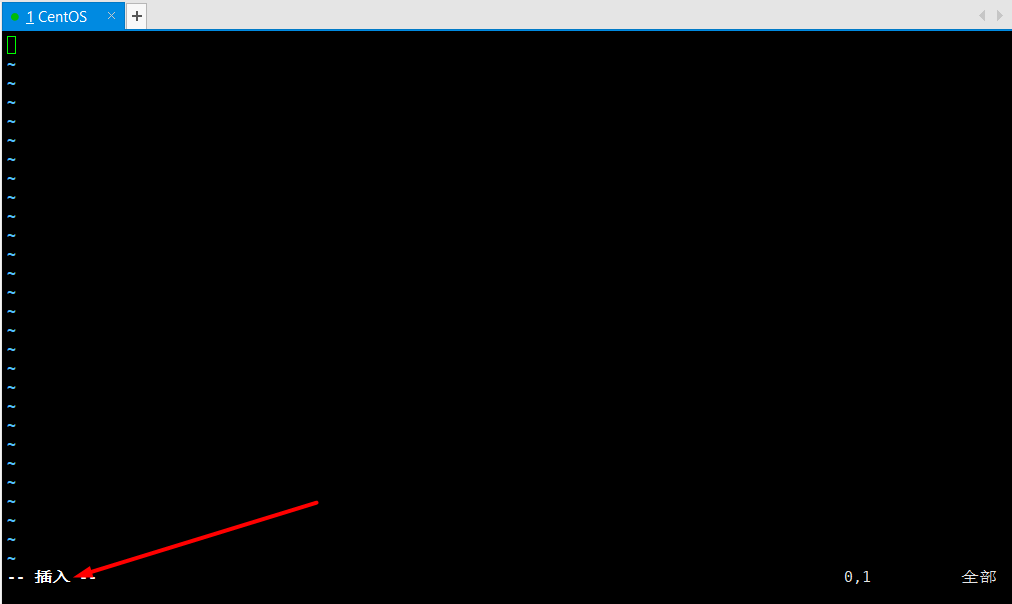
## 基本使用

1. 使用远程登录连接到Linux服务器，执行vim Demo.java命令，表示在当前目录下创建一个Demo.java文件（如果该文件已存在，则打开该文件），进入正常模式。如下：

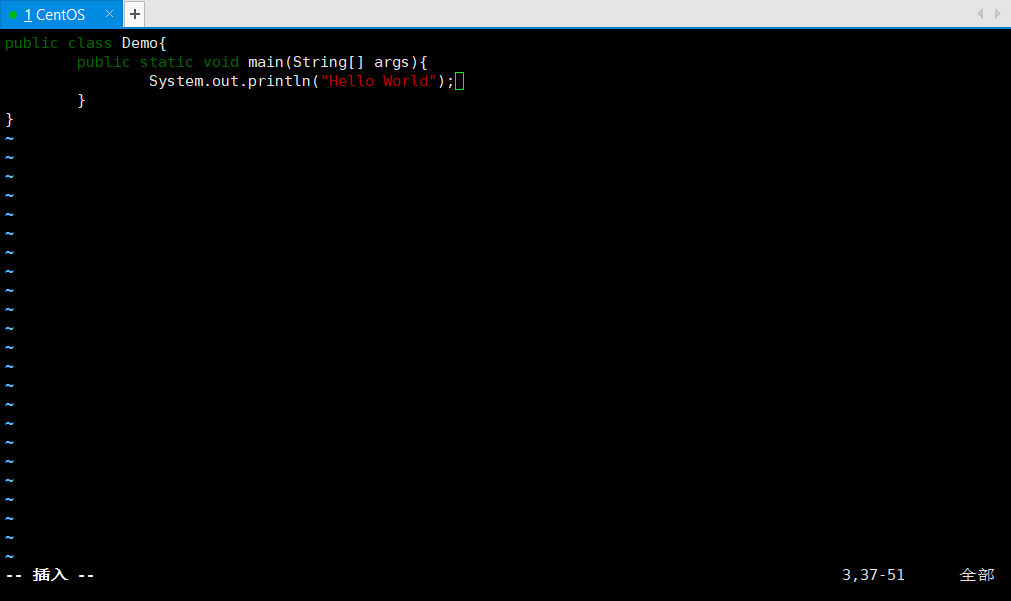


因为是一个新文件，所以里面什么东西也没写；

1. 按“i”键，进入编辑模式。如下：

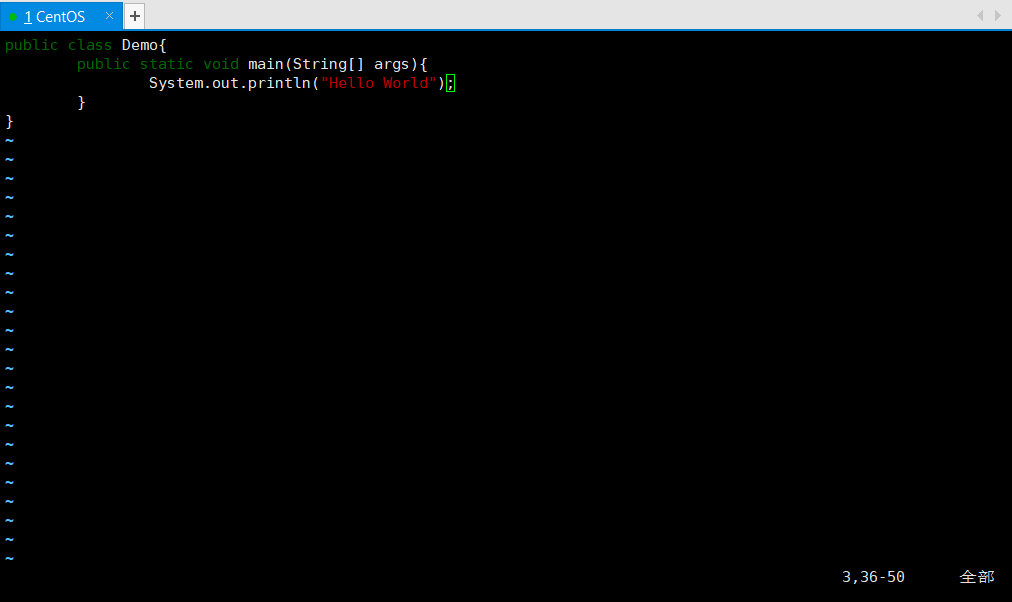


3）在编辑模式下，可以执行写操作，如下：

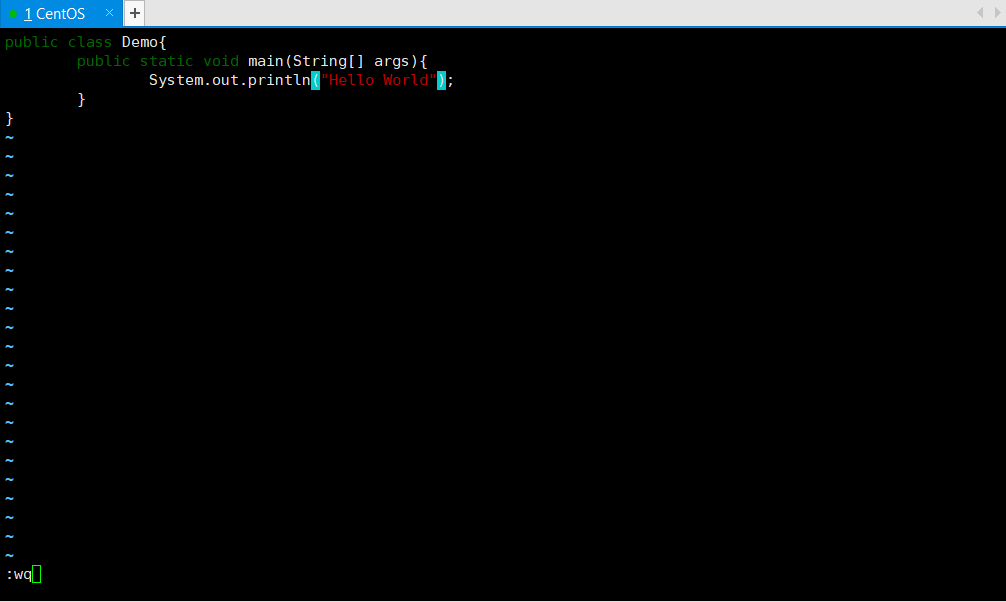


在这里写了一段Java代码，输出Hello World；

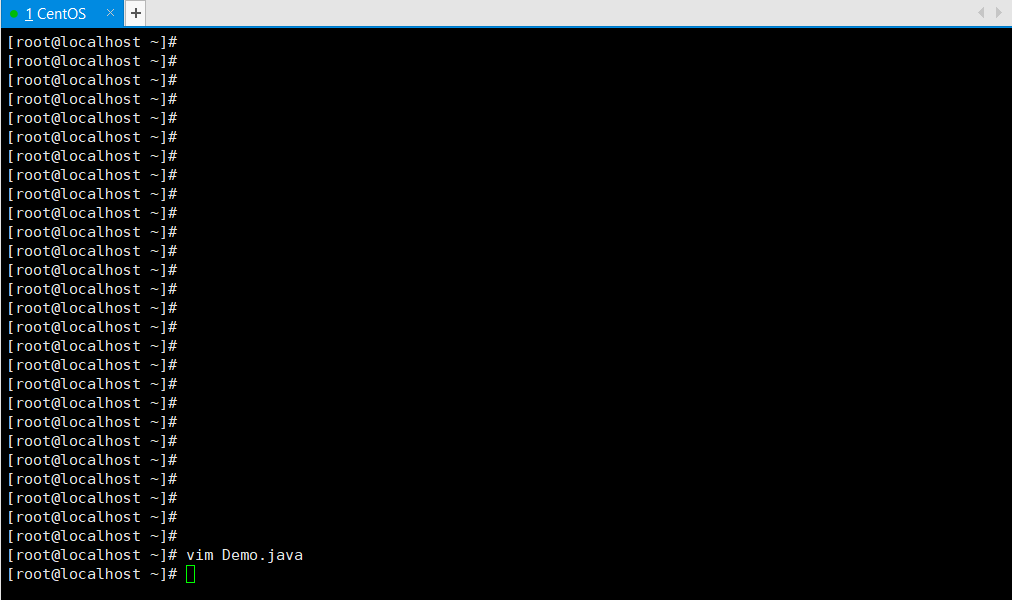
1. 在编辑模式下，如果需要对写的内容进行保存或者不保存，都需要按Esc键进入命令模式，如下：



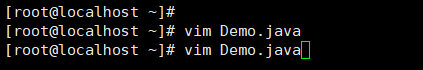
1. 从编辑模式进行命令模式后，如果需要保存修改，可以使用:wq指令执行保存，如下：



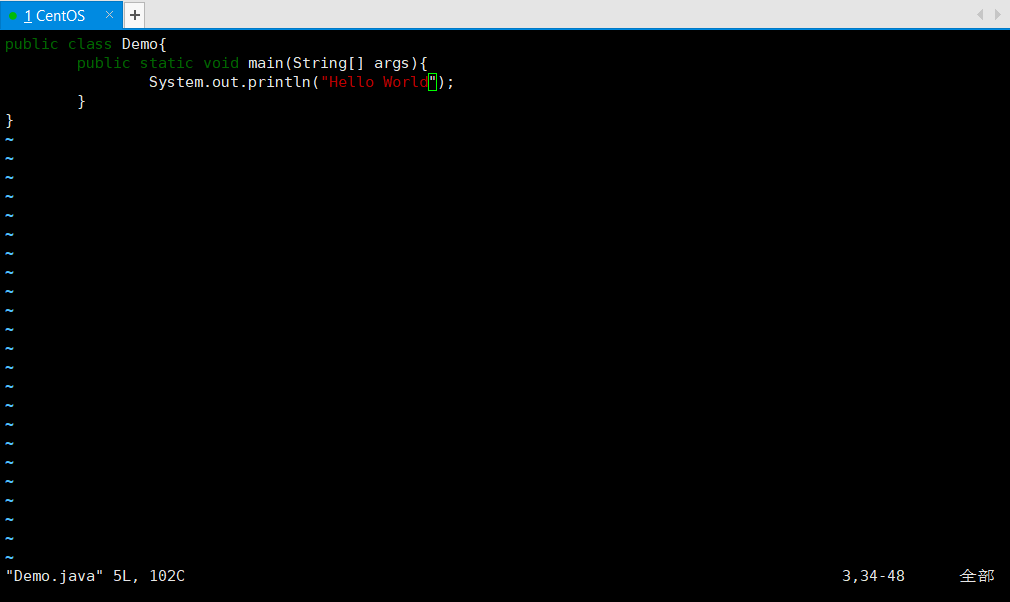
按回车键后，就会保存操作，如下：



此时再执行vim Demo.java命令打开Demo.java文件，如下：



可以发现，内容被保存，如下：



## 快捷键

这些快捷键都是在正常模式或命令模式下使用的，编辑（插入）模式下无法使用；

1）yy：

yy表示的是按y键两下，功能是复制当前光标所在行，复制后可以按p键进行粘贴，如果想复制当前光标下的n行，如5行，则按5键，然后双按y键，表示复制当前光标所在行的下面5行，复制后可以按p键进行粘贴；

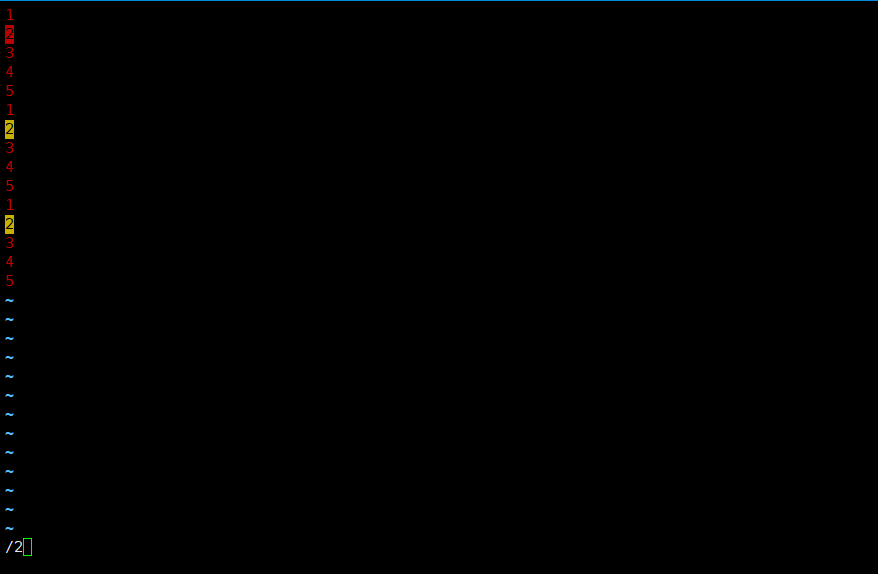
2）dd：dd表示按d键两下，功能是删除光标所在行；

如果想删除光标下面的n行，如5行，按5键，然后双按d键，表示删除光标所

在行的下面5行；

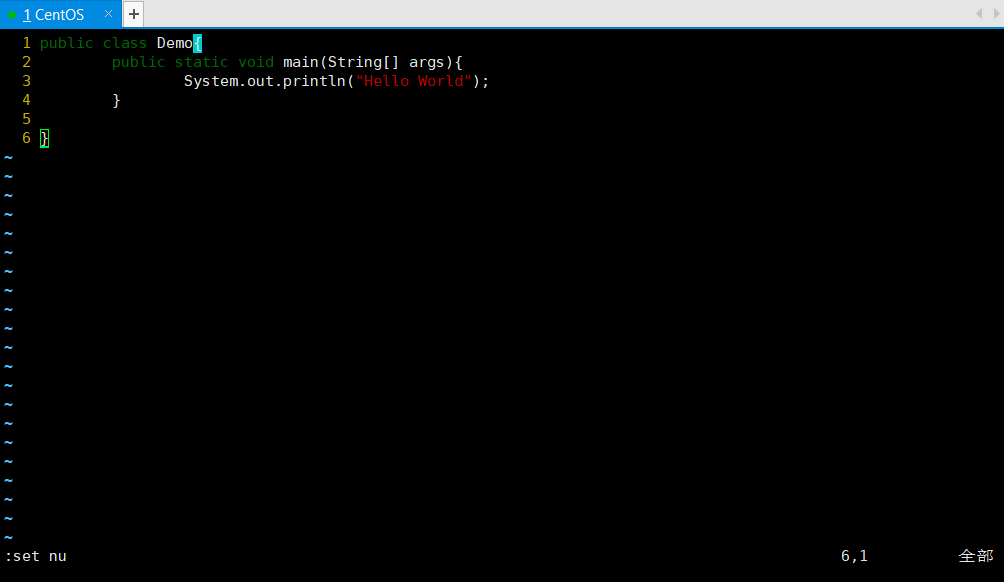
3）/：/表示按一下/键，功能是查找字符串，如查

找2；

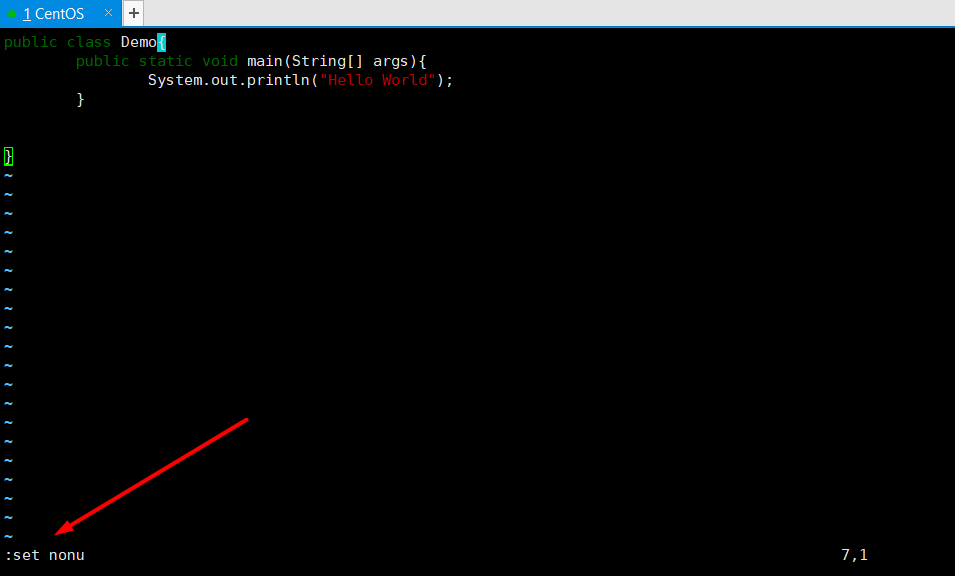


查找结果后，按小n表示向下查找；按大n表示向下查找；

4）:set nu：表示显示行号，如下：



5）:set nonu：表示不显示行号，如下：



G：表示按大G键，功能是将光标移动到文件末尾；

gg：表示按小g键两下，功能是将光标移动到文件首部；

6）u：表示按一下u键，功能是撤销；

7）shift+g：表示同时按shift键和小g键，功能是将光标移动到指定行；比如，按5，

然后同时按shift键和g键，此时会发现光标移动到了文件的第5行；

# 关机重启指令

## 关机指令

立即关机：**shutdown now**

一分钟后关机：**shutdown -h 1**

## 重启指令

立即重启：**shutdown -r now**

立即重启：**reboot**

## 同步指令

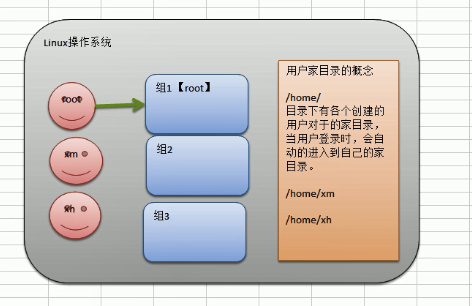
把内存的数据同步到磁盘上：**sync**

建议在重启和关机前执行一次，但看不到效果，它的作用可以这样描述：在关机前执行该命令后，如果内存中的某些数据还没有保存，此时内存中的数据会写入磁盘，防止数据丢失；

# 用户管理

Linux系统是一个多用户多任务的操作系统，任何一个要使用系统资源的用户，都必须首先向系统管理员申请一个账号，然后以这个账号的身份进入系统；

在实际开发中，Linux管理员不可能直接把root权限给程序员，此时就需要使用到用户管理。



在Linux中，每一个用户至少属于一个组，并且每一个用户都会在/home下存在着相应的目录（是用户的家目录）；

## 创建用户

创建用户的基本语法为：

**useradd [选项] 用户名**

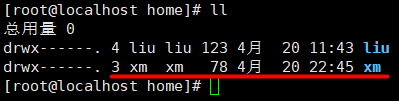
为用户设置密码的语法为：

**passwd 用户名**

1. 创建一个xm用户，如下：



执行该命令后，会创建一个用户名为xm的用户，并且在/home下创建一个名称为xm的目录表示是xm用户的家目录；

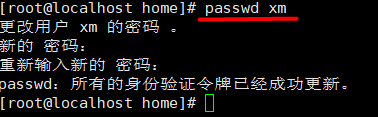


1. 如果不想使用默认生成的家目录作为用户的家目录名，则可以使用以下语法创建用户：

**useradd -d /home/mydir 用户名**

表示将home目录下的mydir目录作为该用户的家目录，如果不存在则自动创建；（但不能把其他用户的家目录给别的用户）

1. 为创建的xm用户设置密码（也可用于修改密码，如果不指定用户名，则表示为root用户修改密码），如下



执行完passwd命令后，会提示设置密码，但输入的密码时看不到的；

此时，使用xm用户就可以登录操作Linux了。

## 删除用户

删除用户分为两种，一种是删除用户但保留其家目录，另一种是删除用户并删除其家目录；

1）删除用户但保留家目录，语法如下：

**userdel 用户名**

如删除xm用户，但保留他的家目录，如下：



1. 删除用户并删除其家目录，语法如下：

**userdel -r 用户名**

如删除xm用户并删除其家目录，如下：



在实际开发中，删除用户时一般都会保留用户的家目录；

## 查看用户信息

查看用户信息的语法为：

**id 用户名**

比如查看xm用户的信息，如下：



在显示结果中，uid表示用户id、gid表示用户所在组的id、组表示用户所在组；uid=1003（xm）表示xm用户是本系统的创建的第4个用户，从1000开始计，gid=1003（xm）表示xm组是本系统创建的第4个组，

## 切换用户

登录Linux后，可以随时切换用户，语法为：

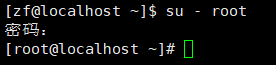
**su – 用户名**

比如，我是使用的root用户登录的，现在我要切换到zf用户，如下：

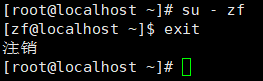


切换成功，并且不需要输入密码，这是因为高权限用户切换到低权限用户是不需要输入密码的；

如果现在要切换到高权限用户，就需要输入密码了，如下：



使用exit指令可以回到切换前的用户，如下：



## 用户组管理

将用户分组，把具有共性的用户分为一个组，方便管理。

### 增加组

增加组的语法为：

**groupadd 组名**

如创建一个wudang组：



### 删除组

删除组的语法为：

**groupdel 组名**

如删除创建的wudang组：



### 为用户指定组

在创建用户时，如果没有为用户指定组，那么在创建用户后，会默认创建一个与用户名相同的组，用户就属于该组，所以在创建用户时，往往需要为用户指定组，语法为：

**useradd -g 用户组 用户名**

如创建一个zwj用户并指定为wudang组组员，如下：



先创建了一个wudang组，然后创建zwj用户并指定为wudang组组员；

可以使用id指令查看zwj用户的信息，看其是否是wudang组组员；



### 修改用户组

某用户为A组组员，而此时想把该用户换到其他组中去，语法如下：

**usermod -g 用户组 用户名**

如存在zwj用户，为wudang组组员，此时存在一个shaolin组，将zwj移动到shaolin组中，成为shaolin组组员：



### 用户和组的相关文件

在Linux中与用户和用户组有关的文件有3个比较重要的文件，它们分别是：

/etc/passwd

该文件是用户的配置文件，记录了用户的各种信息；它每行的含义表示如下：  
 **用户名:加密后的密码:用户ID:组ID :注释型描述:家目录:登录:Shell**

/etc/shadow

该文件是用户密码的配置文件，存放了用户的密码，但是是加密的，它每行的含义表示如下：

**用户名:加密后的密码:最后一次修改时间:最小时间间隔:最大时间间隔:警告活动:不活动时间:失效时间:标志**

/etc/group

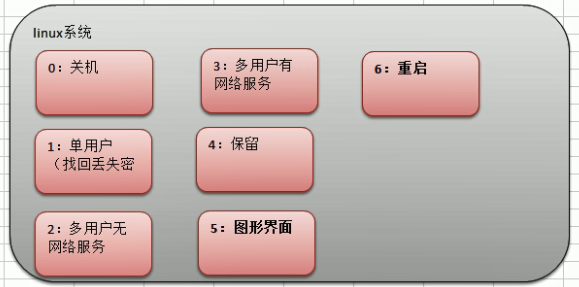
该文件是组的配置文件，记录Linux中组的信息，它每行的含义如下：

**组名:口令:标识符:组的用户列表**

# 运行级别

## 认识运行级别

Linux系统有7种运行级别：



常用的级别是3和5；

## 修改运行级别

如果要修改Linux系统的运行级别，可以修改/etc/inittab文件，也可以通过init指令完成，语法为：

**init 0|1|2|3|4|5|6**

如切换到3号运行级别：



执行该命令后，Linux系统将进入到3号运行级别，3号级别无图形界面；

切换回图形界面，如下：



## 查看运行级别

使用runlevel命令可以查看当前系统的运行级别，如下：



表示运行级别为5；

## 找回root密码

root用户作为最高管理权限，如果将密码忘记了可以进入单用户运行级别重新设置root密码。

# 帮助命令

当我们对某个命令不熟悉的时候，我们可以使用Linux提供的帮助指令来了解这个指令的使用方法。

## man

man命令可以查看某个命令的用法，语法为：

**man [命令或配置文件]**

比如，查看ls命令的用法：



## help

help命令可以查看某个命令的用法，语法为：

**help 命令**

比如，查看cd命令的用法：



## 如何学习一个命令

使用帮助命令因为其显示的是英文，也没有案例，所以，学习一个命令，使用百度更方便；

# 文件目录命令

## pwd

pwd命令用于查看当前所处的目录。



## ls

ls指令用于显示当前目录下的所有文件或目录，语法为：

**ls [选项] [目录|文件]**

常用选项：

-a：显示当前目录所有的文件和目录，包括隐藏的，Linux中隐藏文件以.开头；

-l：以列表的形式显示信息；

-h：显示计量单位；

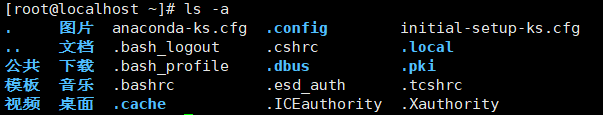
1）基本使用，查看当前目录所有文件和目录，不包括隐藏的：



2）以列表的形式查看当前目录所有文件和目录，不包括隐藏的：



3）查看当前目录下的所有目录和文件，包括隐藏的：



## cd

cd命令用于进入指定的目录，语法如：

**cd [绝对路径|相对路径]**

1. 在任意目录中以绝对路径的方式进入bin目录



1. 返回上一级目录



1. 在根目录下以相对路径的方式进入home目录



1. 在任意目录中回到家目录



## mkdir

mkdir指令用于创建目录，语法为：

**mkdir [选项] 目录名**

常用选项

-p：创建多级目录

1. 在home目录下创建一个a目录



1. 在当前目录中创建一个a目录



1. 创建多级目录，在当前目录下创建b目录，并在b目录中创建c目录,



## rmdir

rmdir指令用于删除空目录，语法为：

**rmdir [选项] 目录名**

删除home目录下的a目录（a目录为空）



这样只能删除空目录，目录非空则会删除失败，如果要删除非空目录，如

删除root目录下的a目录（a目录非空）



## touch

touch用于创建文件，语法为：

**touch 文件名称**

1. 在当前目录下创建一个hello.txt文件



1. 在当前目录创建一次性创建多个文件，1.txt、2.txt



## cp

cp指令用于实现文件（包括目录）的复制粘贴（拷贝），语法为：

**cp [选项] source dest**

source表示要复制的文件，可以是相对路径或绝对路径

dest表示要粘贴到的位置，可以是相对路径或绝对路径

常用选项：

-r：表示复制粘贴整个目录

1. 将当前目录下的1.txt文件复制粘贴到根目录下



1. 将当前目录下的1.txt文件复制粘贴到根目录下并更名为2.txt



1. 将当前目录下的1.txt文件复制粘贴到当前目录下的a目录中去（a目录必须存在）



1. 将当前目录下的a目录复制粘贴到根目录下



1. 如果根目录下已存在1.txt文件，将当前目录下的1.txt文件复制粘贴到根目录下，此时会提示是否覆盖：



如果拷贝的是一个目录，目录中存在几十或几百个文件，就需要输入几十个或几百个y来一次次覆盖，这样很累，我们可以使用强制覆盖：



使用强制覆盖后，将不再会提示是否覆盖。

## rm

rm指令用于删除文件或目录，语法为：  
 **rm [选项] 要删除的文件或目录**

常用选项：

-r：删除目录；

-f：强制删除，不提示

1. 删除当前目录中的1.txt文件



1. 强制删除当前目录下的a目录



如果不强制，则你需要一个个的确定；

## mv

mv指令用于为文件或目录重命名，它还可以实现剪切粘贴（移动）的功能，语法为：

**mv oldFileName newFileName**

**mv souce dest**

1. 将当前目录下的1.txt更名为2.txt
2. 将当前目录下的a目录更名为b



1. 将当前目录下的a目录剪切并粘贴到home目录下

1. 将当前目录下的1.txt文件剪切并粘贴到home目录下



## cat

cat指令用于以只读的方式查看文件内容，不能修改文件，语法为：

**cat [选项] 文件名**

常用选项：

-n：显示行号

1. 查看当前目录下的anaconda-ks.cfg文件内容



1. 查看当前目录下的anaconda-ks.cfg文件内容，并显示行号



1. 如果文件内容过多，则可以分页显示，如查看当前目录下的anaconda-ks.cfg文件内容，显示行号，并分页



分页显示后，按空格键进行换页；

## more

more指令是为了更方便的查看文件内容，即分页查看，与其他命令配合使用，如上面的cat指令，more指令的语法为：

**more 文件名**

1. 查看当前目录下的anaconda-ks.cfg文件

如果该文件内容够长，则会分页显示，如下：



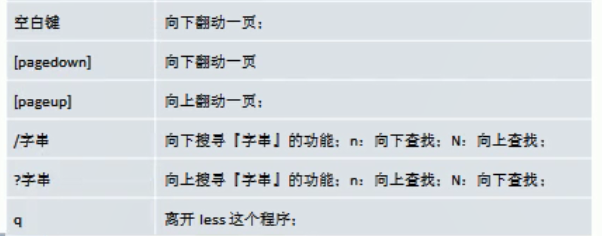
表示已显示57%的内容，此时你可以使用more指令的一些快捷键来进行剩余内容的浏览

## less

less指令与more指令功能类似，都是为了更好的浏览文件内容，但它比more指令更强大，使用more指令会一次性把文件内容列出，然后通过more指令的快捷键进行浏览，而less指令并不是一次性的查看文件的所有内容，而是一页一页的查看，对于大的文件（如大文件的日志），则推荐使用less指令查看，效率会比more指令更高，它的语法为：

**less 文件名**

它的快捷键如下：



推荐使用less执行查看大型文件，如日志文件，这样效率比较高；

## >

> 命令用于将执行其他命令后显示的内容保存为一个指定名称的文件，语法为：

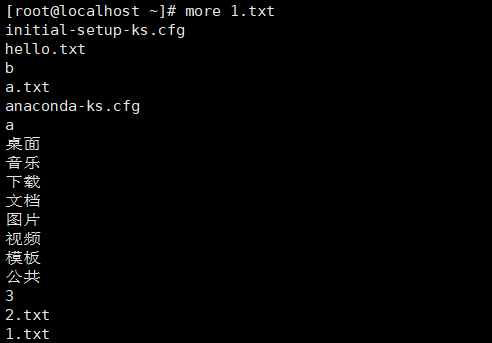
**其他指令 >文件名**

如果指定名称的文件已存在，则覆盖；

1）将执行ls指令后要显示的内容保存为1.txt文件



此时查看1.txt文件，可以发现将ls指令显示的内容被保存起来了



1. 使用cat指令查看anaconda-ks.cfg文件，将它的文件内容保存为2.txt文件



1. 将指定的内容保存为1.txt文件，如hello world



## >>

>>命令用于将执行其他命令执行后显示的内容追加到指定的某个文件中，语法为：

**其他命令 >>文件名**

1）将ls指令显示的内容追加到1.txt文件中



2）通常cat查看hello.txt文件，将其显示的内容追加到1.txt文件末尾



1. 将字符串Hello World最加到1.txt文件末尾



追加后，可以通过more指令查看1.txt文件，观察是否追加成功；

## echo

echo指令用于向控制台输出内容或者查看环境变量，语法为：

**echo [选项] [输出内容]**

也常常用于查看环境变量；

1. 使用echo指令输出PATH环境变量



1. 向控制台输出hello world



## head

head指令用于查看文件的前n行内容，如不指定n，则默认查看文件的前10行内容，语法为：

**head [选项] [n] 文件名**

常用选项：

-n：表示自定义要查看的行数；

1. 查看当前目录下anaconda-ks.cfg文件的前10行内容



1. 查看当前目录下anaconda-ks.cfg文件的前25行内容



## tail

tail指令与head指令作用相反，它用于查看文件的后n行内容，如不指定n，则默认查看文件的后10行内容，处理查看文件的后n行内容外，它还有一个重要的作用，就是实时监控功能。语法为：

**tail [选项] [n] 文件名**

常用选项：

-n：表示自定义要查看的行数

-f：实时监控文件，查看其是否有有变化

1. 查看当前目录下anaconda-ks.cfg文件的后10行内容



1. 查看当前目录下anaconda-ks.cfg文件的后20行内容



3）实时监控当前目录下的hello.txt文件



监控hello.txt文件后，该终端将进入阻塞状态，不同执行其他命令，此时，如果有其他用户修改了此文件，就会在该终端打印出变化；如果想退出监控，请Ctrl+C；

## ln

ln指令用于生成软链接（快捷方式），软链接中存放了链接其他文件的路径，语法为：

**ln -s [原文件或目录] [软链接名]**

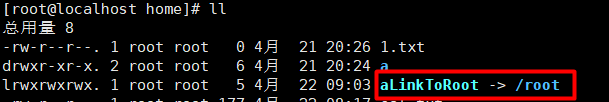
选项：

-s：表示允许软链接指向的是一个目录

如：在当前目录下创建一个软链接，链接到/root目录，软链接名称为aLinkToRoot



此时使用ll命令查看



通过cd指令使用该软链接，就会进入root目录



虽然使用的是root目录，但仍然处于软链接所在目录



如果要删除软链接：



## history

history指令可以查看被执行过的历史指令，它还可以执行历史指令，语法为：

**history [n]**

1. 查看所有执行过的指令



可以发现每条历史指令都有一个编号；

1. 查看最近执行过的n个指令，如要查看最近执行过的10个执行



3）执行编号为5的历史指令



# 时间日期命令

## date

date指令用于查看时间日期或设置时间日期

1. 查看当前时间和日期



1. 查看当前年月日



对于中间的连接符“-”可随意；

1. 查看当前年月日时分秒



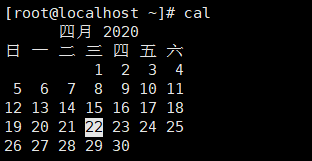
1. 设置日期时间



设置完成后，返回系统当前时间；

## cal

cal指令用于查看日历信息，：



查看2020年所有月份的信息：



# 搜索查找命令

## find

find命令用于查找某个文件或目录是否存在，如果存在，则列出它的路径，语法为：

**find [搜索范围] [选项]**

搜索范围指的是要搜索的目录，如不指定，则默认从当前目录中查找；

常用选项：

-name：表示按照指定的文件名或通配模式查找文件；

-user：表示查找属于指定用户的文件；

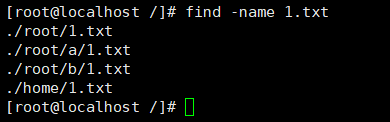
-size：表示查找指定大小的文件；

1. 查找home目录（包括所有下级目录）的1.txt文件



文件名也可以是包含通配符；

1. 在当前目录下（包括所有下级目录）查找1.txt文件



文件名也可以包含通配符；

1. 查看当前目录中属于某用户的文件，如查看属于liu用户的文件



1. 查看当前目录下大于2 0M的文件



1. 查看root目录下小于20M的文件



1. 查看当前目录下等于20M的文件



1. 查找当前目录下等于20k的文件



## locate

locate指令可以快速的定位文件，因为locate指令利用事先建立的系统中所有文件名称及路径的locate数据库快速定位给定的文件，它不需遍历整个文件系统，查询速度快，为了保证查询结果的准确度，管理员必须定期更新locate时刻。语法为：

**locate 文件**

由于locate是基于数据库的查询，所以第一次使用它时必须使用updatedb指令创建locate数据库；



查找当前目录下所有的1.txt文件



文件名也可以是通配符；

locate查找的范围是整个Linux系统；

## grep

grep指令用于在文件中过滤查找某个内容，通常与管道符“|”搭配使用，管道符的作用先执行管道符“|”前面的指令，然后将其执行结果交给管道符后面的指令进行处理；grep指令的语法为：

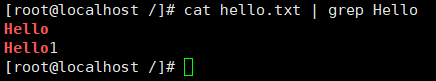
**其他命令|grep [选项] [查找内容] 源文件**

常用选项：

-n：显示匹配号及行号；

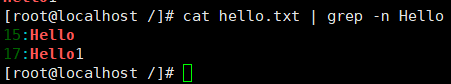
-i：忽略字母大小写

1）查看hello.txt文件中是否存在Hello内容，如果存在，则列出

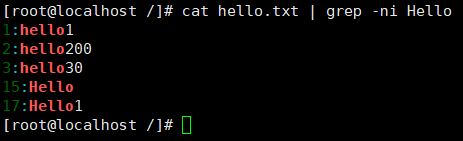


以上命令的执行流程：先执行cat命令查看hello.txt文件的内容，然后将hello.txt文件内容交给grep指令，grep指令查看文件内容中是否存在Hello内容，如果存在，则列出；

2）查看hello.txt文件中是否存在Hello内容，如存在，则列出，并显示其所在行



1. 使用grep过滤内容是默认区分大小写的，如果不想区分大小写则可以使用-i参数，如查看当前目录下的hello.txt文件中是否存在hello内容，且不分大小写，如果存在，则列出并显示行号



# 压缩与解压缩命令

在Linux中，压缩与解压缩的命令一共有3种：

1. gzip和gunzip
2. zip和unzip
3. tar

## gzip

gzip命令用于压缩文件（它只能压缩文件不能压缩目录），语法为：

**gzip 文件名**

压缩后会将原先的文件删除掉；

如将当前目录下的hello.txt文件进行压缩



使用ll指令查看压缩后产生的文件：



## gunzip

gunzip命令用于将压缩后的文件进行解压缩操作，语法为：

**gunzip 文件名**

解压缩后，压缩文件将被删除；

如将当前目录下的hello.txt.gz文件进行解压缩



使用ll指令查看解压缩后的产生的文件：



## zip

zip指令用于压缩文件或目录，语法为：

**zip [选项] 文件或目录**

压缩后原文件不会被被删除；

常用选项：

-r：表示压缩目录；

如：将home目录进行压缩



-r表示对目录进行压缩；

home.zip表示压缩包的名称；

/home表示要压缩的目录；

压缩包默认保存在当前目录中；

使用ll指令查看当前目录是否存在home.zip，如下：



## unzip

unzip指令用于对zip类型的压缩包进行解压缩，语法为：

**unzip [选项] 压缩包名称**

解压缩后压缩包不会被删除；

常用选项：

-d：表示将压缩包解压到指定目录下，默认解压缩在本目录下；

如：将当前目录下的home.zip目录解压缩到root目录下



## tar

tar指令用于压缩和解压缩，压缩包文件名是以.tar.gz结尾的文件，语法为：

**tar [选项] xxx.tar.gz 需要打包的文件后目录**

常用选项：

-c：执行压缩操作；

-x：执行解压缩操作；

-v：表示显示命令的执行过程；

-f：指定压缩包的文件名；

-z：表示通过gzip或ungzip指令处理执行打包或解压缩；

1）将当前目录下的1.txt、home.txt文件压缩为 a.tar.gz 压缩包



以上命令使用了-zcvf参数，表示使用了-z -c -v -f这4个常用选项；

使用ll指令查看是否存在a.tar.gz，验证是否压缩成功：



2）将home目录进行压缩，压缩包的名称为myhome.tar.gz



1. 将当前目录下的a.tar.gz压缩包解压缩到当前目录下



5）将当前目录下的myhome.tar.gz压缩包解压缩到指定目录下，如/root目录下



指定的目录必须存在，否则报错；

# 权限管理

Linux中的权限管理指的是对文件或目录的权限。

## 文件所有者和所在组和其他组

### 查看文件的所有者和所在组

在Linux中，每个文件（包括目录）具有所有者、所在组等属性，所有者即文件是哪个用户的，所在组即文件是属于哪个组的。一般来讲，谁创建了这个文件，该用户就是这个文件的所有者；同时，该文件所在组也是该用户的所在组，可以通过ls指令查看文件的所有者和所在组，语法为：

**ls -ahl**

-ahl即-a、-h、-l选项，-a表示显示隐藏文件或目录，-h表示显示文件大小的单位，-l表示以列表的形式列出文件和目录；

案例：

1. 创建一个组，名为police，创建一个用户tom，指定为police组组员，设置tom用户的密码

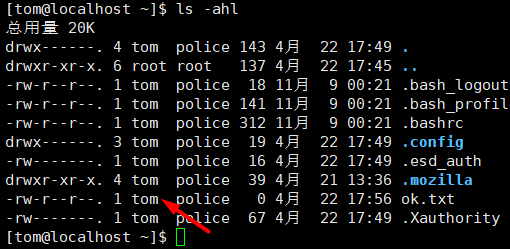




1. 再开一个连接，使用tom用户登录终端，进入tom用户的家目录，创建一个ok.txt文件



1. 使用ls -ahl指令查看ok.txt文件所有者



可以观察到这个文件ok.txt文件的所有者是tom，所在组是police

### 修改文件所有者

默认情况下，文件的所有者是创建这个文件的用户，可以使用chown修改文件的所有者，语法为：

**chown 用户名 文件名**

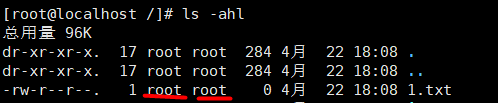
案例：

使用root用户创建一个1.txt文件，将该文件的所有者更改为tom用户

1. 创建1.txt文件



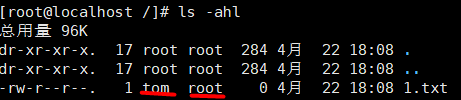
使用ls指令查看1.txt文件，其所有者是root用户，所在组是root组



1. 使用chown指令修改1.txt的所有者



使用ls指令查看1.txt文件，可以发现所有者从root变为tom，但是所在组并没有改变



因此，修改文件所有者，并不能修改文件的所在组；

### 修改文件所在组

使用chown指令可以修改文件所有者，但不能修改文件所在组，可以使用chgrp指令修改文件所在组，语法为：

**chgrp 组名 文件名**

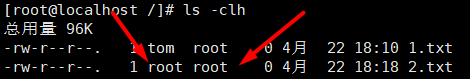
案例：

使用root用户创建2.txt文件，查看该文件的所在组，之后将该文件的所在组修改为police组

创建2.txt文件



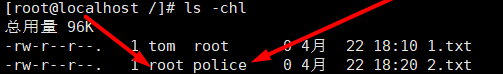
使用ls指令查看该文件的所在组



可以发现，文件所在的组是root组，所有者是root用户，此时修改文件所在组



此时，在查看2.txt文件的所在组



所在组被成功修改为police，但所有者没变；

因此得知，修改文件所在组并不能修改文件所有者；

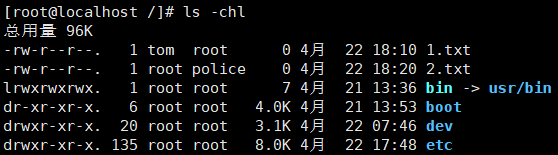
### 其他组

其他组指的是文件所在组之外的组，比如，存在a组、b组、c组，存在1.txt，该属于a组，则b、c组为其他组；

## 认识权限

### 权限介绍

在使用ls或ll指令查看当前目录的所有文件时，同时也列出了这些文件某些属性。



以列出的1.txt文件为例



第0位表示该文件的类型，这里为-，表示该文件是个普通文件，如果是d，则表示该文件是一个目录；如果为l，则表示该文件是一个软链接；如果是c，则表示这是一个字符设备【键盘、鼠标】，前面已经讲过，在Linux中一切皆为文件，包括设备；如果是b，则表示这是一个块文件，如硬盘；

第1~3位表示文件所有者对该文件的权限，这里为rw-，表示该文件的所有者对它具有读（r）和写（w）的权限；

第4~6位代表该文件所在组中的用户对该文件的权限，这里为r，表示该文件所在组中的用户对该文件只具有读（r）权限；

第7~9位表示其他组中的用户对该文件的权限，这里为r- -，表示其他组中的用户对该文件只具有读（r）权限；

1所在位置的含义需要根据文件的类型而定，如果该文件是一个目录，则表示该目录中子目录的个数；如果是文件，表示硬链接的数；这里的1.txt是一个文件，所以这里的1表示硬链接数是1，（因为1.txt文件只有它本身可以被点击，它不是一个目录）

tom表示该文件的所有者；

root表示该文件的所在组；

0表示该文件的大小，如果该文件时目录，该处值为4096，表示目录的大小，但并不是目录中内容的大小，而是目录本身的大小，因为目录是一个特殊的文件；

4月 22 18:10表示该文件的最后一次修改时间；

1.txt表示该文件的文件名；

### 权限详解

在上一节权限介绍中，说明了文件的权限，它分为3种，即文件所有者对文件的权限、文件所在组用户对文件的权限、其他组用户对文件的权限；而权限具体可以分为3类，即r、w、x；

根据文件类型的不同，r、w、x所代表的含义又不同，

1）当rwx作用于文件时：

r：表示可读（read）；

w：表示可写（write），但是不代表可以删除，删除这个文件的前提是对该文件所

在的目录有写权限，才能删除该文件；

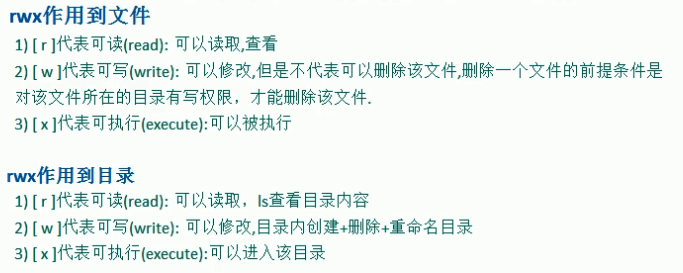
x：表示可执行（execute）

2）当rwx作用于目录是：

r：表示可以查看该目录中的内容，如使用ls命令；

w：表示可以修改该目录，在该目录内创建、删除、重命名目录或文件；

x：表示可以进入该目录；



## 权限管理

### 修改权限

chmod指令用于修改文件或目录的权限，在修改权限时，u表示文件所有者，g表示文件所在组，o表示其他组；

案例演示：

1. 修改当前目录下的abc文件的权限，为该文件的所有者赋予读写执行权限，为该文件所在组用户赋予读执行权限，为其他组用户赋予读权限



u表示文件所有者，g表示文件所在组，o表示其他组；abc表示要修改权限的文件；

修改后，abc文件的权限情况如下：



同时发现abc文件名颜色变绿了，因为在Linux中，绿色文件表示可执行文件；

1. 修改当前目录下的abc文件的权限，为该文件的所有者除去执行权限，为所在组用户增加写权限



修改后，abc文件的权限情况如下：



1. 修改当前目录下的abc文件的权限，为所有用户添加读权限



a表示所有用户；修改后abc的权限情况如下：



### 修改文件所有者

虽然前面已经讲过修改文件所有者，在这里再次进行补充，使用chown指令修改文件所有者：

1. **chown 用户名 文件名**：用于修改指定文件的所有者；

如将当前目录下1.txt文件的所有者修改为tom用户



2）**chown 用户名:用户组 文件名**：用于同时修改文件的所有者和所在组；

如将当前目录下的1.txt文件的所有者修改为tom用户，所在组修改为police组



3）**chown -R 用户名 目录名**：用于修改指定目录下的所有目录以及文件的所有者；

如将当前目录下的a目录下的所有目录及文件的所有者修改为tom



### 修改文件所在组

虽然前面已经讲过如何修改文件所在组，在这里再次进行补充，修改文件所在组可以使用chgrp指令实现。

1）**chgrp 组名 文件**：修改文件或目录的所在组；

如：修改a.txt文件的所在组为police



2）**chgrp -R 组名 目录名**：修改指定目录下所有文件的所在组；

如：修改kkk目录下所有文件的所在组为police



## 最佳实践

### 练习1

这是一个警察土匪之间的游戏；

1）存在police、bandit组，police组组员有jack、jetty，bandit组组员有xh、xq











记得为他们设置密码；

1. jack用户创建一个文件，他自己可以读写，本组人可以读，其他组对该文件没有任何权限



创建文件后，查看文件的权限



发现并没有满足要求，此时需要修改文件权限，



再次查看文件权限，如下：



1. 情况有变，jack用户刚刚创建的文件又需要进行修改，jack用户需要将该文件的权限修改为本组人可以读写，其他组可以读；



查看文件权限，如下：



修改成功；

1. 此时，作为bandit组的xh想要查看jack用户刚刚创建的文件，它是有权限读的，因为jack刚刚为文件设置权限时，设置了其他组对该文件具有读权限；但是jack创建的文件是在他自己的家目录中创建的，xm用户在进入他的家目录时，发现进不去



通过查看权限发现，xh对该目录没有任何权限

此时xh想要进入该目录，则至少要用户对该目录的x权限，同时还需要一个r的权限，才可以读该目录，他可以向jack用户索取或root用户索取rx权限，xh想了想，干脆投靠到jack用户所在的组，如下：



虽然xh投靠了police组，但是xh还是进不去jack用户的家目录，因为jack的家目录并没有向他的所在组进行开放，此时jack用户向所在组进行开放，如下：



观察到已成功开放



此时，xh需要注销重新登录一下，即可进入jack用户的家目录：



 就可以对jack家目录中的jack01.txt文件进行读写操作了，

因为xh是police组组员，所以对该文件具有读写权限



1. 而作为bandit组的xq对该文件没有任何权限；

### 练习2

先把练习1做完再来做练习2；



# 定时任务调度

## 基本语法

通过定时调度可以让Linux系统在指定的时间执行某个脚本或某块代码来完成某项工作，比如数据备份、病毒扫描等，crontab命令用于实现定时调度，语法为：

**crontab [选项]**

常用选项：

-e：增加定时任务

-l：查询定时任务；

-r：删除当前用户所有的定时任务；

如果只是简单的定时任务，可以不用写脚本，直接在执行crontab指令后加入任务即可，而对于一些复杂的任务，需要编写脚本，然后在使用crontab命令时引用它；

## 创建定时任务

### 案例1

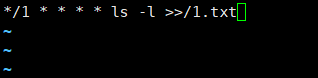
编写定时任务，每隔一分钟执行一次“ls -l >/1.txt”指令，表示 将当前目录下所有的文件列出来，然后将其列出的内容追加到根目录下的1.txt文件中；

1）执行crontab指令，增加定时任务：



该命令执行后，会进入vim编辑器；

2）在vim编辑器中，写入你要执行的定时任务：



3）保存后，会提示任务已安装



4）一分钟后查看根目录下的1.txt文件中是否每隔一分钟更新内容；



### 案例2

通过编写shell脚本的形式，创建一个定时任务，任务的目的是每隔1分钟将当前时间信息追加到/home目录下的dateinfo.txt文件中；

1）创建一个简单的shell脚本文件



在shell脚本中写入要做的任务，记得保存（必须是绝对路径）



脚本文件保存后，通过ll指令查看，发现它不是可执行文件



此时给需要使用该shell脚本的用户添加可执行权限



再次查看，mytask1.sh可执行（绿色代表可执行）



2）执行corntab指令，追加定时任务





/home/mytask1.sh表示要执行的shell脚本的所在路径，必须是绝对路径；上面的定时任务是上一案例添加的；

1. 添加定时任务后，保存文件，一分钟后查看/home/dateinfo.txt文件中是否每隔一分钟就向该文件中追加时间信息；

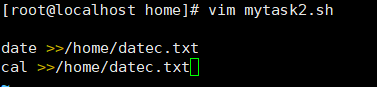


成功。

### 案例3

通过shell脚本的形式，每隔一分钟，将当前时间和日历追加到home目录下的datec.txt文件中；

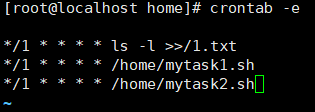
1. 创建shell脚本并编写任务



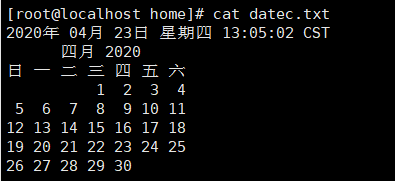
保存后，给需要使用该shell脚本的用户添加可执行权限



1. 执行corntab指令，引用shell脚本，追加定时任务



1. 保存后，一分钟后观察/home/datec.txt文件中是否每隔一分钟追加了时间和日历信息；



创建定时任务成功；

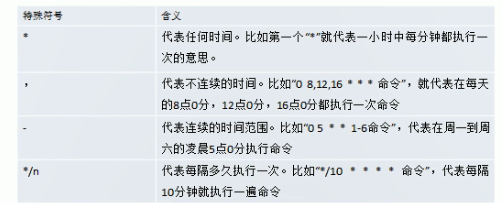
### 案例4

每天00:00定时备份Mysql中的数据库；（由于尚未安装MySQL），所以等后面学习了如果安装MySQL后，再进行操作；

## 任务参数说明

执行corntab -e指令后，进入vim编辑器编写具体的任务，通过5个\*号来指定该任务何时执行，这5个\*号的含义如下：







## 查看重启删除定时任务

**crontab -l：**查看当前用户所有的定时任务：

**crontab -r**：删除属于当前用户所有的定时任务：

**service crond restart：**重启任务调度：

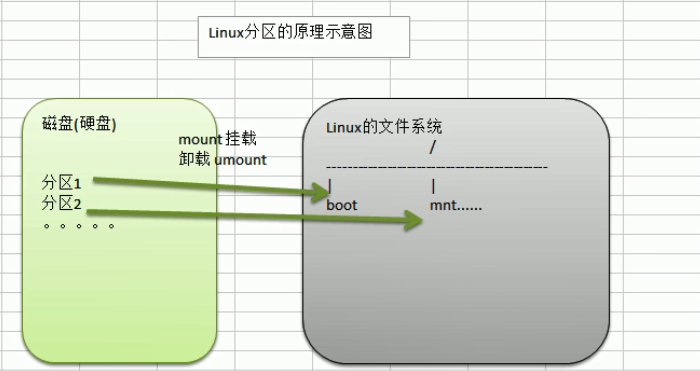
注意：Linux中的定时任务调度是分用户的，tom用户创建的定时任务，只有tom用户才能查看、删除、重启该定时任务，其他用户不能做到；

# 磁盘管理

## 介绍

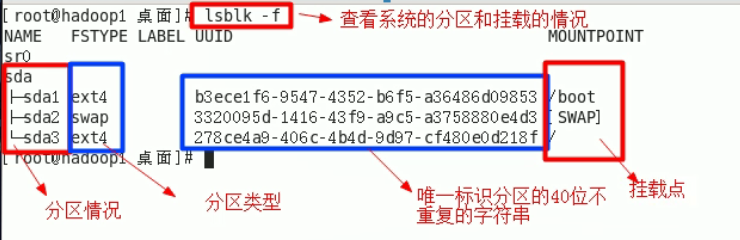
在Linux中，无论有多少个分区，它归根到底就只有一个根目录，Linux中每个分区都是用来组成整个文件系统的一部分；

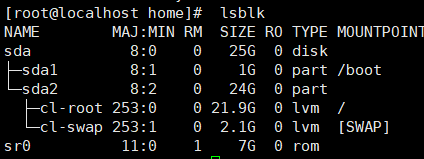
Linux采用一种叫“挂载”的处理方法，它的整个文件系统中包含了一整套的文件和目录，且将一个分区和一个目录联系起来，这时要载入的一个分区将使它的存储空间在一个目录下获得；



## 查看系统分区与挂载情况

可以使用lsblk查看系统分区与挂载情况，或者使用lsblk -f；



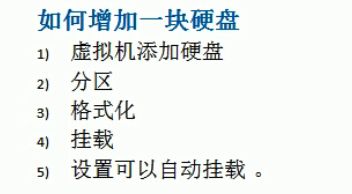


## 添加新硬盘

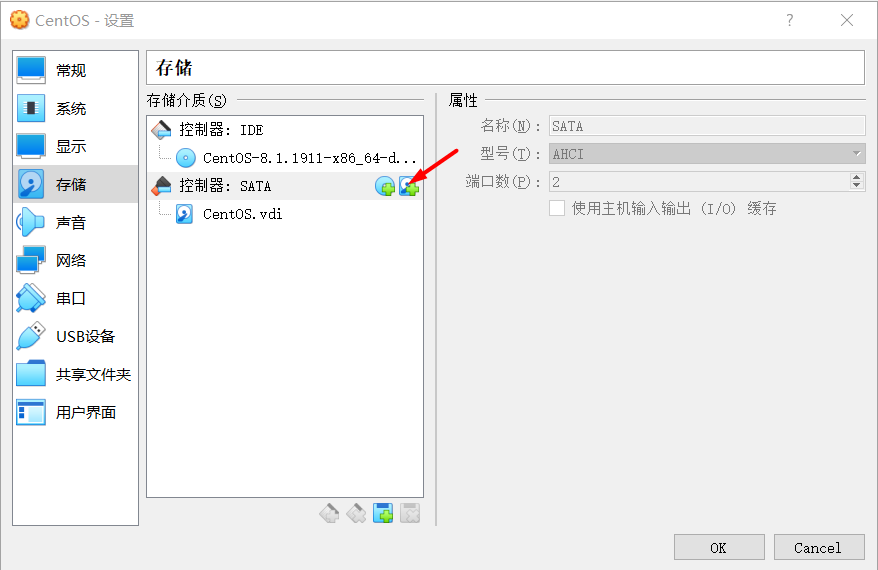
需求：假设Linux系统中硬盘的容量不够了，此时添加一个新硬盘，将其挂载到

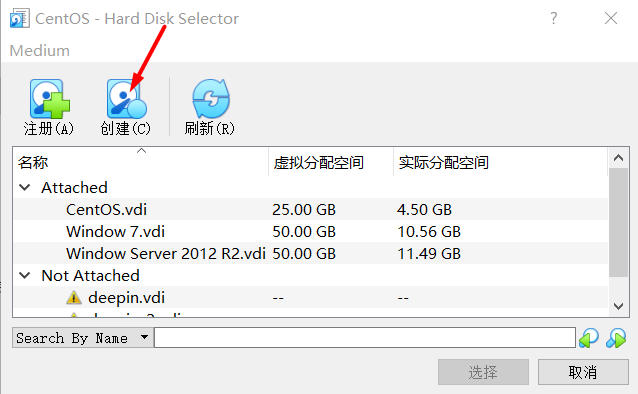
/homt/newdisk目录；

添加一块新磁盘的步骤如下：

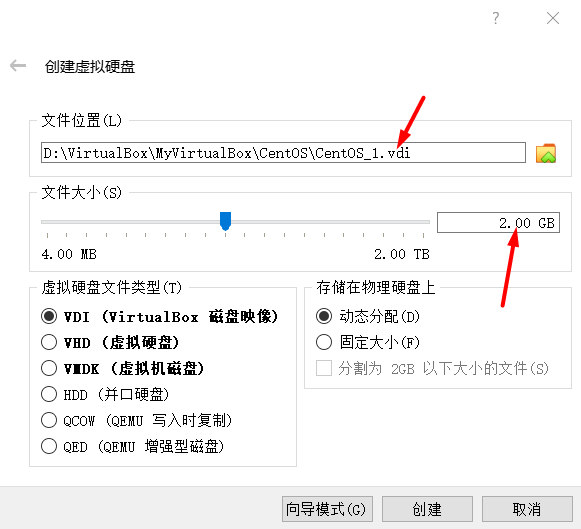


1. 使用虚拟机为该Linux系统新增一块硬盘

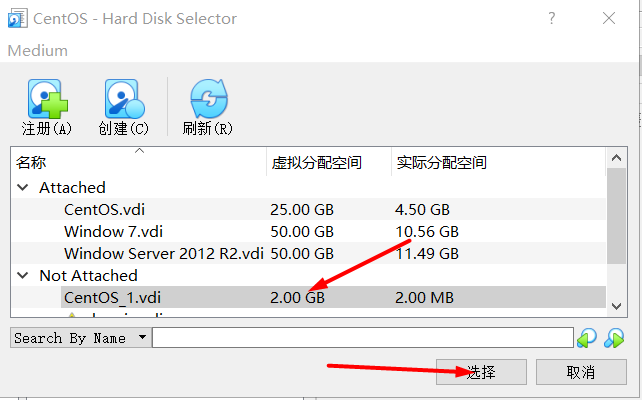




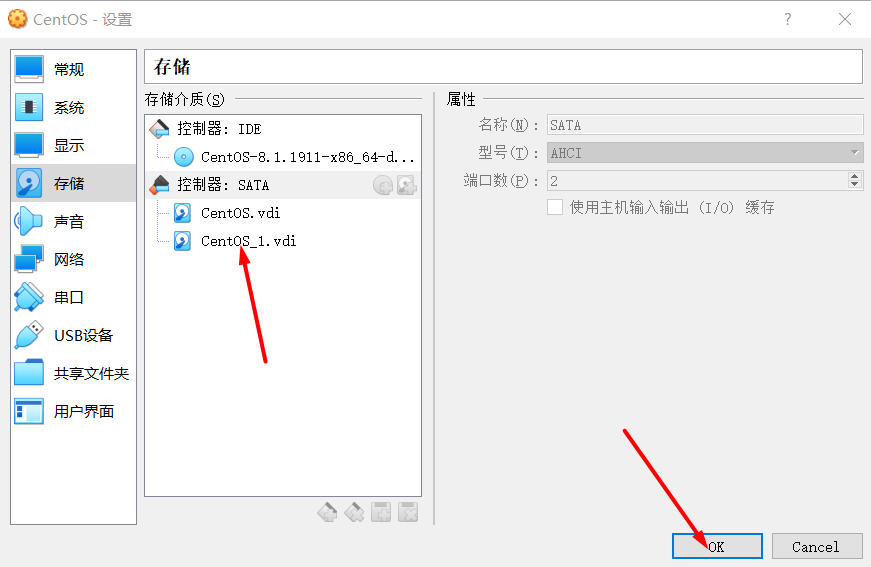
因为只是演示，所以只给了2GB



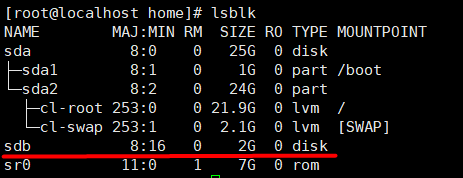
创建后，然后选择创建的磁盘



可以发现成功添加，点击OK



在终端通过lsblk指令查看是否添加成功



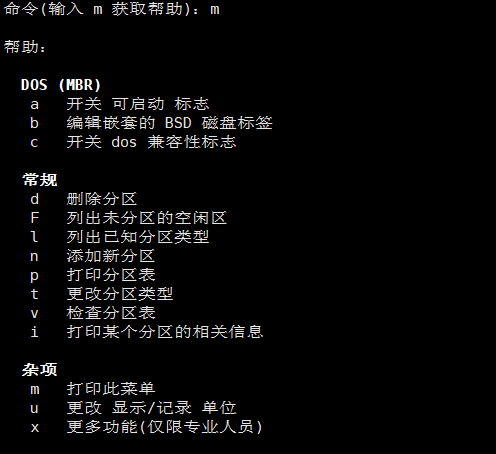
已添加成功（如果没有没有看到，则将Linux系统重启一下之后再执行lsblk执行查看）；

2）虽然添加了硬盘，但该硬盘没有进行分区，此时需要为硬盘进行分区

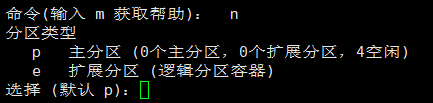
执行fdisk /dev/sdb指令



根据需要决定是否要输入m获取帮助，此时它会将命令列出



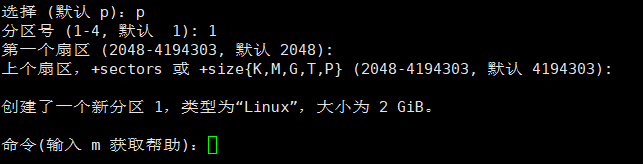
其中存在一个n命令，如上图中所示，n命令用于添加新分区，输入n命令执行



在这里有两种类型，根据需要进行选择，这里选择p

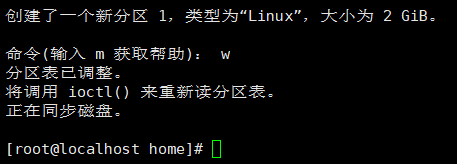


系统会让你输入分区号，根据提示输入即可，之后一路回车



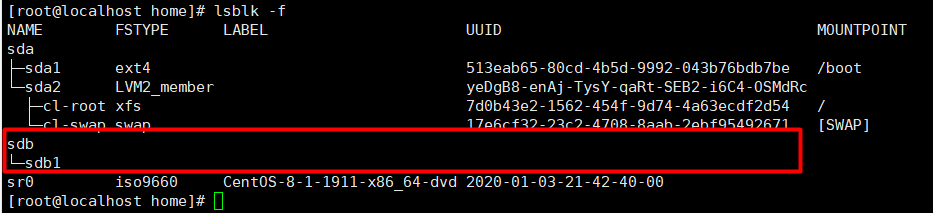
 分区创建完成后，它又会提示你输入命令，此时输入一个w命令，它的作用如下：

可以输入m查看w命令的作用，执行w命令



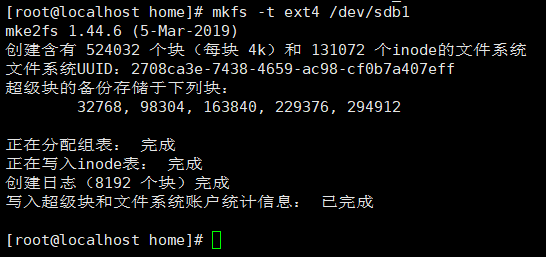
分区完成。

使用lsblk -f指令查看

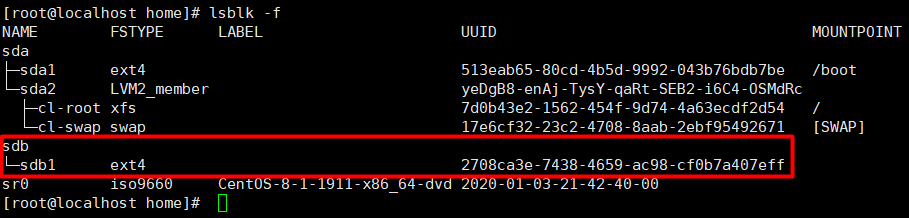


可以看到，虽然分区成功了，但是并没有像其他一样显示UUID等信息，这是因为该要对其进行格式化；

1. 分区完成后，还需要对其进行格式化，



再次使用lsblk -f指令查看



发现与其他分区一样了。

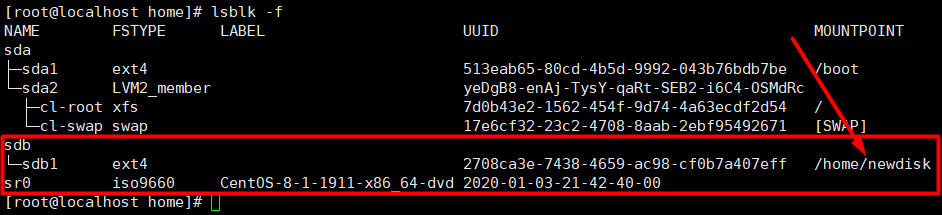
1. 进行挂载，将其挂载到/home/newdisk目录，在挂载之前，newdisk目录必须存在，所以先在home目录下创建newdisk目录



通过mount指令进行挂载操作

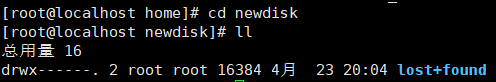


查看是否挂载成功：



挂载成功。

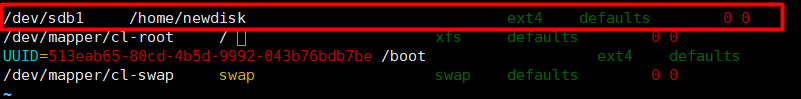
此时，进入/home/newdisk目录中，也可以发现挂载成功



1. 虽然上一步将硬盘分区挂载在/home/newdisk目录，但那属于临时挂载，如果系统重启了，硬盘分区与/home/newdisk目录之间挂载关系就会丢失；对此，我们可以设置永久挂载，在etc目录下有一个fstab文件，将其打开



这个文件记录了系统分区以及挂载点的情况；

在这个文件中添加一行

表示将sdb1挂载到/home/newdisk目录；

保存后，执行mount -a指令，使设置生效

然后重启系统



1. 如果因为某种原因，如挂载时挂错了目录，可以将其卸载，使用umount指令，

**mount 设备名或挂载目录**



注意：在卸载在不能处于其挂载的目录中

## 查看磁盘使用情况

Linux运行速度变慢有可能是磁盘满了，所以有必要查看磁盘的使用情况。

1. **df**：查看系统整体磁盘的使用情况



1. **du**：查询指定目录的占用磁盘情况，语法为：

**du [选项] 目录**

常用选项：

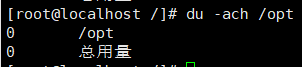
-h：显示计量单位；

-s：列出使用情况汇总；

-a：也列出文件的使用情况；

-c：列出明细的同时，也列出汇总；

--max-depth：指定子目录的深度；



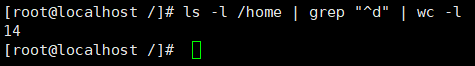
磁盘查询案例：

1. 统计/home目录下文件的个数（不包括子目录下的文件）

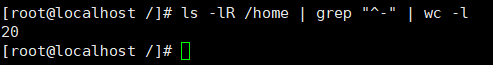


“^-表示”根据文件的类型进行过滤，文件类型通过第0位区分，前面在学习权限管理的时候已经讲过；

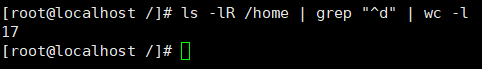
1. 统计/home目录下目录的个数（不包括子目录中的目录）



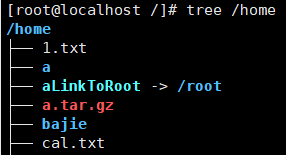
1. 统计/home目录下文件的个数（包括子目录下的文件）



1. 统计/home目录下目录的个数（包括所有的子目录）



1. 以树状的形式显示/home目录结构

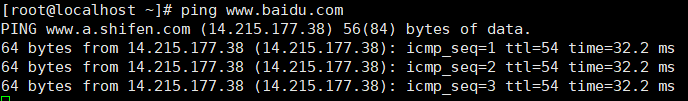


使用tree指令时，如不指定要显示的目录，则默认显示当前目录；

# 网络配置

## 检查网络是否有用

检查Linux系统是否有网络或者网络是否有用可以使用ping命令，如下：

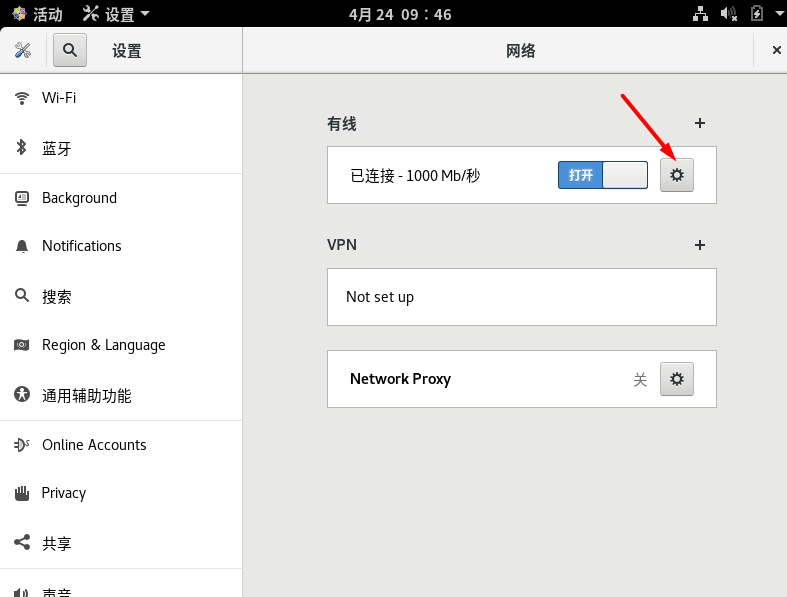


ping通表示有网络，如上图所示。

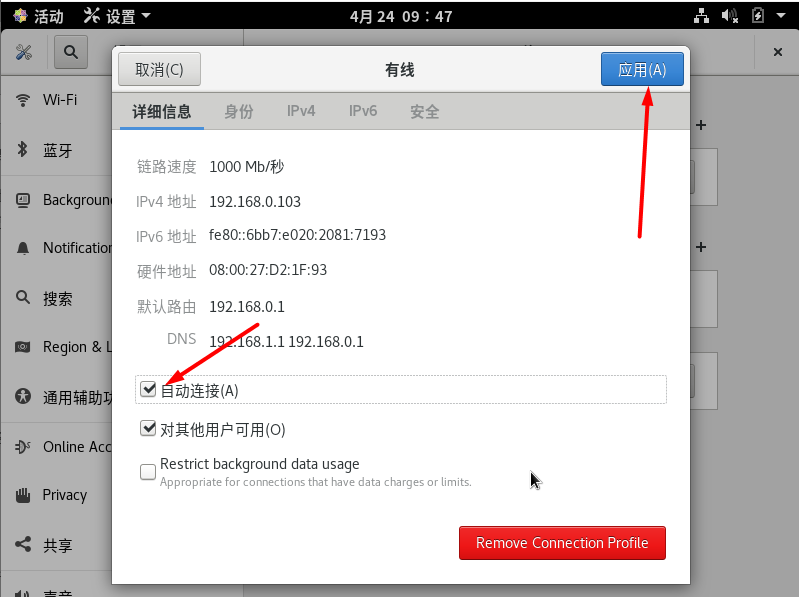
## 设置网络自动连接

Linux系统每次启动后，都需要手动进行网络的连接，对此，我们可以将其设置为自动连接，如下：

打开网络设置：



将网络设置成自动连接并应用



设置自动连接后，Linux系统启动将自动连接网络，不需要手动进行连接；

## 设置固定IP

如果Linux系统是作为服务器使用的，那我们还需要设置固定的IP（静态IP），否则将网站域名与IP绑定后，而Linux服务器的IP却随时变化，很明显这是不行的，所以，我们需要为Linux系统设置固定的IP地址（静态IP）；如果只是Linux只是作为个人电脑使用，你可以不设置。

由于老师课程中的操作与我实际中的操作环境的不一致问题，所以没有写下笔记，但需要记住的是：作为服务器的电脑，IP一定要固定。

# 进程管理

进程管理非常重要，必须熟练掌握；

1. 在Linux中，每一个执行的程序都是一个进程，每一个进程都有一个ID号；
2. 每一个进程，都会对应一个父进程，而这个父进程可以有多个子进程；
3. 每一个进程可能会以两种方式存在，前台与后台，所谓前台进程是指显示在屏幕上可以进行操作的程序，如打开终端，这个终端就是一个前台进程；后台进程是指运行在后台，但没有在屏幕上显示的程序，如 MySQL服务；

## 查看系统进程

ps命令可以系统中的进程，语法为：

**ps [选项]**

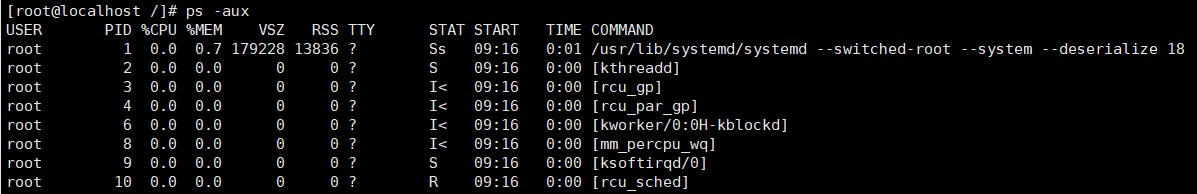
常用选项：

-a：显示当前系统中所有进程信息；

-u：显示进程是哪个用户启动的；

-x：显示进程运行的参数；

如：执行**ps -aux**命令后



USER：进程是哪个用户执行的；

PID：进程号；

%CPU：进程占用CPU的情况

%MEM：进程占用内存的情况；

VSZ：进程占用虚拟内存的情况；

RSS：进程占用物理内存的情况；

TTY：进程使用的终端；

STAT：进程的当前状态，S表示进程休眠、R表示进程正在运行、s表示该进程是是

会话的先导进程、N表示该进程拥有比普通进程优先级更低的优先级、D表

示进程正在短期等待、Z表示进程僵死、T表示进程被跟踪或停止等等；

START：进程是什么时候启动的；

TIME：进程占用CPU的总时间；

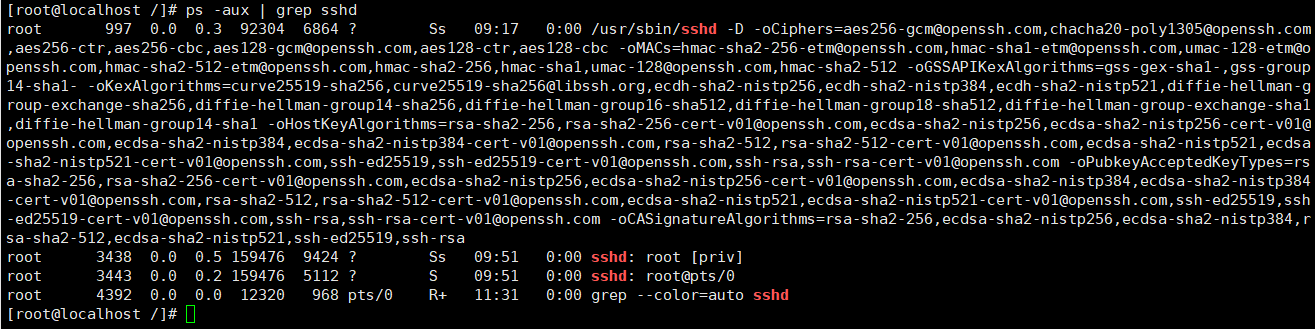
COMMAND：进程启动使用的命令（如果过长或会拦截显示）；

## 查看指定进程

如果不想查看全部进程的情况，而是只关心某一个进程的情况，可以使用ps命令并通过grep命令进行过滤。如查看sshd进程



执行后



可以发现并不是真正的实现了查看指定进程的功能，而是通过grep命令对内容进行过滤实现的，所以也没有显示USER、PID等字段

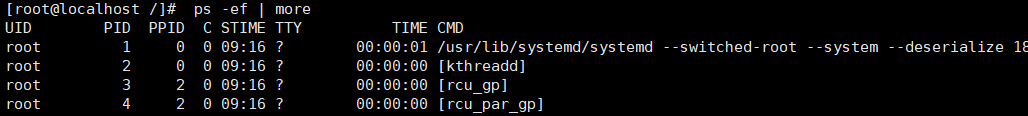
## 查看父进程



“ps -ef”用于查看父进程；

“| more”用于分页显示；

执行后，如下：



UID表示该进程是哪个用户启动的；

PID表示进程ID；

PPID表示PID所代表的进程的父进程ID；

STIME表示进程的启动时间；

CMD表示启动进程所用的命令和参数；

## 查看进程树

pstree命令可以已树状的形式显示进程，能够一眼就看出父子进程关系，语法为：

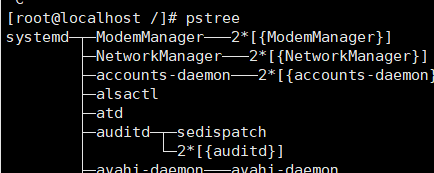
**pstree [选项]**

常用选项：

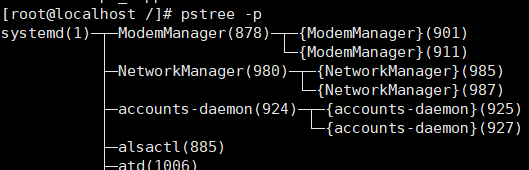
-p：表示以数状的形式显示进程号（PID）；

案例：

1. 以树状的显示进程



1. 以树状的形式显示进程号（PID）



## 终止进程

当某一个进程不需要的时候，比如非法进程、黑客入侵执行的进程、占用大量内存的进程等，我们可以将其终止，终止一个进程可以使用kill或killall命令。

kill指令可以根据进程号终止该进程，语法为：

**kill [选项] 进程号**

常用选项：

-9：在终止核心进程时，可能无法终止掉，此时追加-9选项，可以强制终止该

进程；

killall指令可以根据进程名终止该进程，因为进程名支持通配符，所以它也可以用来

一次性终止多个进程，语法为：

**killall [选项] 进程名**

常用选项：

-9：在终止核心进程时，可能无法终止掉，此时追加-9选项，可以强制终止该

进程；

案例：

1. 踢掉某个非法登录用户

打开一个连接，使用tom用户登录Linux，如下：



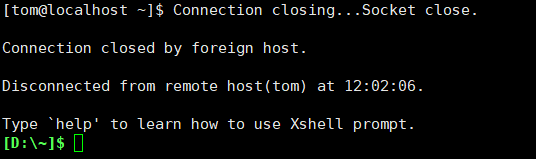
此时root用户需要将其踢掉，查看进程，找到tom用户进程的进程号，如下：



将tom用户的进程终止，如下：



然后，查看tom用户，成功被踢掉，强制退出了，如下：



1. 终止远程登录服务

比如Linux运维师正在做维护，不希望有任何人登录Linux，所以想暂时终止远程登录服务，在合适的时候再开启；

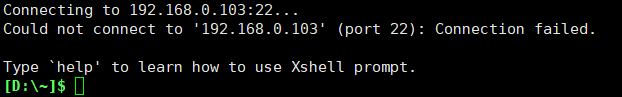
查看进程，找到sshd进程的进程号：



将其终止：



此时其他用户在登录时，将登录不上：



注意：已登录的用户不会被踢掉；

1. 根据名称终止进

登录Linux系统（不是使用远程登录），打开终端，执行gedit命令，打开一个文本编辑器，如下：



执行后，会打开一个文本编辑器：



再开一个终端，使用killall命令根据名称终止这个文本编辑器的进程



此时会发现文本编辑器被关闭了，同时另一个终端显示“已终止”

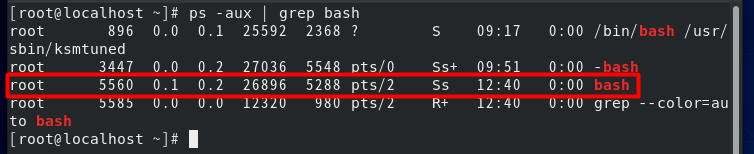


killall指令常用杀掉一批进程，比如某个父进程启用了很多子进程，此时使用killall命令根据父进程名称终止父进程，此时它的所有子进程也会被终止；

1. 强制终止某个进程

以强制关闭终端进程为例

登录Linux（不是远程登录），开启一个终端，并查看该终端的进程（终端本身是一个shell脚本，进程名为bash）：



根据终端进程号终止该终端



会发现当前正在使用的终端并没有被关闭，这是Linux认为终端是个比较重要的程序，不会轻易终止

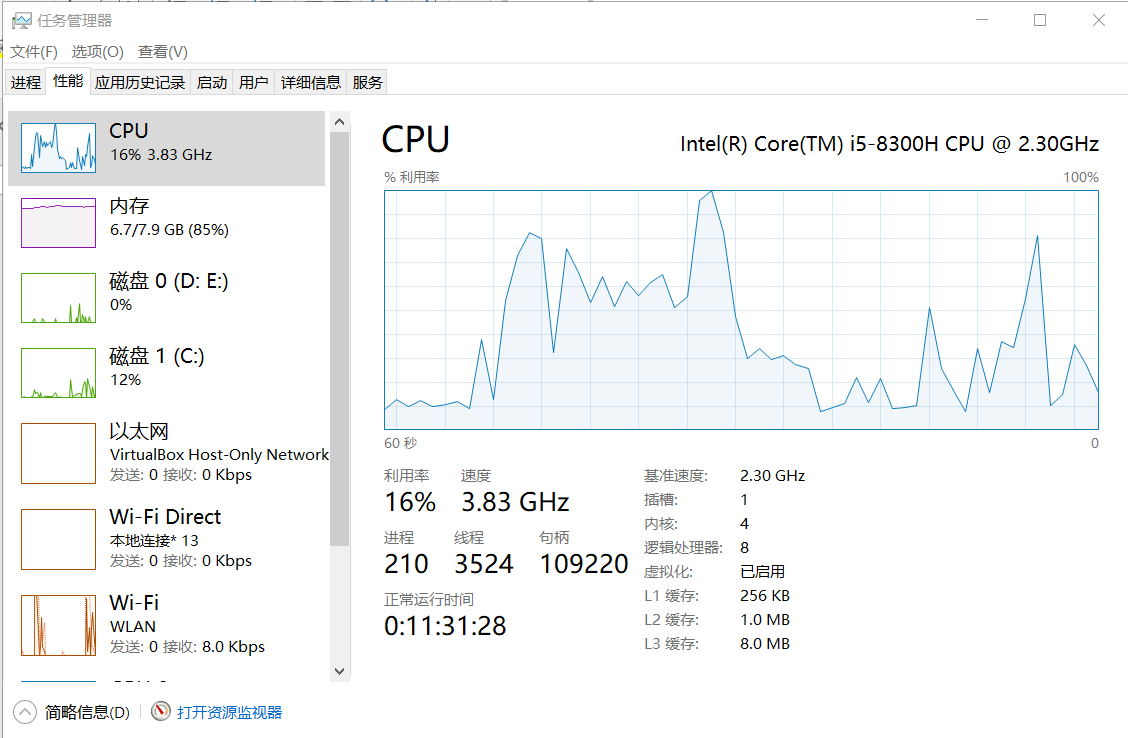
此时，我们使用-9选项强制终止：



执行该命令后，会发现当前正在使用的终端被关闭；

## 监控进程

top命令可以动态的监控进程，它与ps命令功能相似，用于显示正在执行的进程，不同的是top命令在执行一段时间后可以更新显示的进程，如CPU占用时间等，有点类似于Window下的性能监控：



top命令的语法为：

**top [选项]**

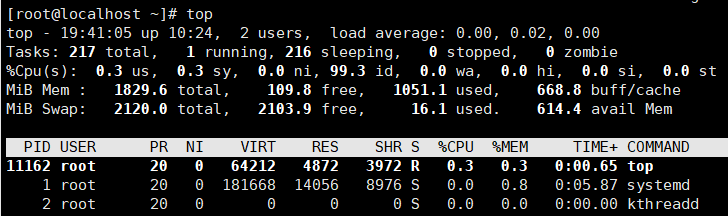
常用选项：

-d：指定top命令每隔几秒更新，默认为3秒；

-i：不显示闲置的或僵死的进程；

-p：只监控指定进程号对应的进程；

如执行top命令后



在显示的内容中，第一行的top -后面表示闲现在的实际，up后面表示的是系统从启动开始运行了多久，这里是10:16，表示10个小时16分钟；2 users表示有2个用户正在使用系统；load average后面跟着的是负载均衡，当这3个值加起来得到的平均值超过了0.7，说明系统已经有点跑不动了，

在第2行中，Tasks后面的表示的是一共有217个进程，1个正在运行的，216个休眠的，0个停止的，以及0个僵死的；

在第3行中，%Cpu(s)后面显示的是CPU的占用情况，0.3us表示用户占了0.3%，0.3sy表示系统占了0.3%，99.3 id表示剩余99.3%是空闲的，说明CPU还够用；

在第4行中，Mib Mem后面显示的是内存的使用情况；total表示总的，free表示空闲的，userd表示使用了的

在第5行中，表示的是硬盘的使用情况， otal表示总的，free表示空闲的，userd表示使用了的；这个硬盘是在学习磁盘管理时新挂载的一个评判，所以没有被使用多少；

PID表示进程号，USER表示是谁开启的这个进程，这些在上面已经介绍过了；

案例：

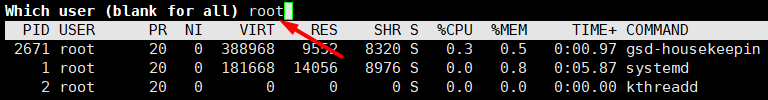
1. 监视特定用户启动的进程

执行top命令后，会监视所有用户的进程，如果想要监视某个指定用户或者自己的进程，也是可以的；

执行top命令



显示监控信息后，输入u，然后会提示你输入用户名，输入你要监视的用户



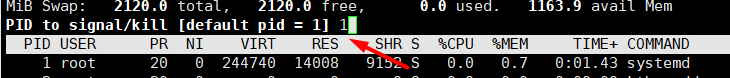
这里以root为例，回车后，将不再监视其他用户的进程，而是只监视root用户的进程；

1. 在监控进程时终止某个进程

执行top命令



显示监控信息后，输入k，然后输入你要终止进程的进程号（PID）：



回车，即可把进程号对应的进程给终止；

1. 指定监控信息更新的时间

执行top命令监控进程，默认是每隔3秒更新一次，我们也可以自定义，如设置为每隔5秒更新一次

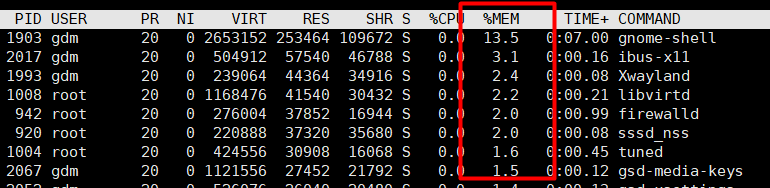


1. 将监控信息进行排序显示

执行top命令后，被监控的进程默认是按照PID从小到大进行排序显示的，但我们也可以指定

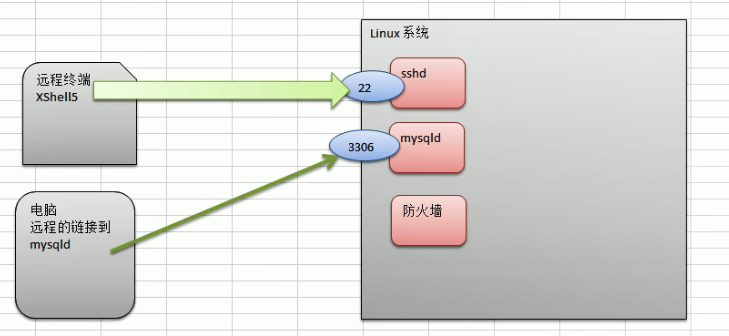


如执行top命令，显示监控信息后，输入一个M，那么被监控的进程将内存的使用率排序并进行显示；



# 服务管理

服务（service）的本质是进程，是运行在后台的进程，通常服务会监听某个端口，等待其他程序的请求，比如mysql服务默认监听3306端口、sshd服务默认监听20端口、防火墙等。因此在Linux中服务又称为守护进程。



systemctl命令用于进行服务管理，语法为：

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

start表示启动服务；stop表示停止服务，restart表示重启服务于，reload表示重载服务、status表示服务的详细信息；

## 启动服务

启动服务，以firewalld（防火墙）为例



## 停止服务

停止服务，以firewalld为例



## 重启服务

重启服务，以firewalld为例



重启服务即先停止服务，然后再启动服务；

## 查看服务是否启动

判读服务是否启动，以firewalld为例



active表示已启动，inactive表示未启动；

## 查看服务详细信息

查看服务的详细信息，以firewalld为例



## 判断服务是否开机自启动

判读服务是否是开机自启动，以firewalld为例



enabled表示是；disabled表示不是；

## 禁止服务开机自启动

禁止服务开机自启动，以firewalld为例



## 实现服务开机自启动

实现服务开机自启动，以firewalld为例



## 查看所有的服务

**systemctl list-units –type=service**



## 查看所有未启动的服务

## 查看所有已启动的服务

查看所有已启动的服务



在显示结果中，UNIT字段表示服务名；ACTIVE字段表示状态；

## 查看网络服务

netstat命令用于检验本机各端口的网络连接情况，以及查看本机端口占用情况、使用情况，查看某个端口的连接情况（如被谁连接了）等，netstat命令的语法为：

**netstat [选项]**

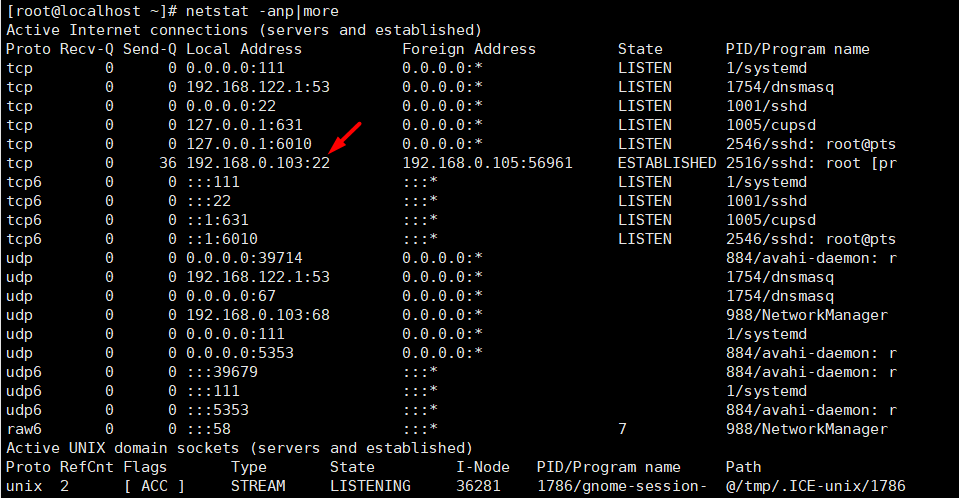
常用选项：

-an：按一定顺序排列输出；

-p：显示哪个进程在调用；

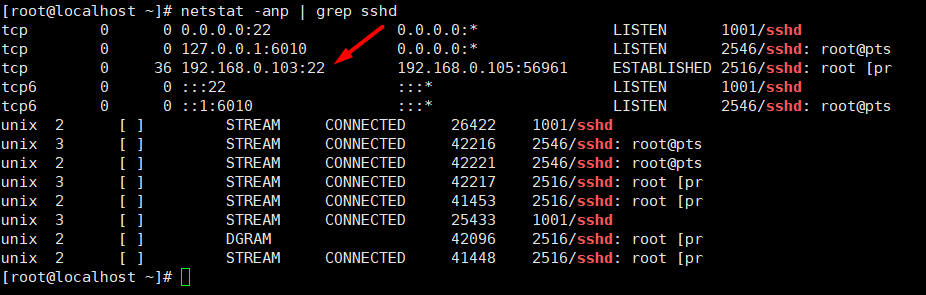
案例：

1. 查看所有的网络服务，这其中包括了端口的使用情况，连接情况等，如下：



如sshd服务使用了22端口，用于远程登录，当前连接到该端口的IP有192.168.0.105:56961；

1. 如果想要查看某个服务使用的端口，我们可以使用grep命令进行过滤，如查看sshd服务使用端口是什么端口



# 软件包管理

Linux中的软件包管理分为RPM包管理与YUM包管理。

## RPM包管理

1. RPM是RedHat Package Manager的缩写，即RedHat软件包管理工具，它包含在某些Linux发行版本中，生成具有.RPM扩展名的文件。这个玩家格式虽然打上了RedHat的标志，但是它是通用的，Linux发行版本都有采用RPM。
2. RPM类似于windows下的setup.exe，通过RPM可以给Linux系统安装程序，也可以查看系统是否安装了某些程序，如火狐浏览器、MySQL等；

3）RPM包即软件；

### 查询RPM包

查询RPM包（查看系统安装了哪些软件）的语法为：

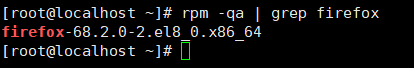
**rpm -qa**

案例：

1）查看系统已安装的所有RPM包（程序），



1. 如果需要查询某个指定的RPM包是否安装，如火狐，则可以使用grep命令对查询结果进行过滤；



firefox表示软件名称，68.2.0-2表示软件版本，el8\_0.x86\_64表示软件位数；

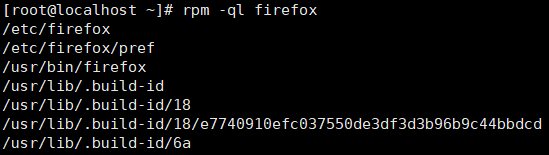
1. 不使用grep进行过滤，查看某个RPM包是否安装，如火狐



1. 查看安装的RPM包的详细信息，如安装日期、安装版本等，以火狐为例



1. 查看某个RPM包的安装路径以及安装文件等，以火狐为例：



1. 查询某个文件是属于哪个RPM包的，以/etc/passwd文件为例



### 卸载RPM包

卸载RPM包（卸载已安装的软件）的语法为：

**rpm -e RPM包名称**

比如卸载火狐，如下：



注意：如果卸载某个RPM包时，该RPM被另外某个或多个RPM依赖了，则会提示该

RPM包被依赖了，此时你有两个选择，一是放弃，二是强制卸载，强制卸载如下（加上- -nodeps参数即可）：



但尽量不要这么做，强制卸载有风险，除非是非常必要要卸载这个软件；

### 安装RPM包

安装RPM包（安装软件）的语法为：

**rpm -ivh RPM包名称**

-i表示安装，-v表示提示，-h表示显示安装进度；

安装步骤：存在火狐rpm包，进入该包所在目录（也可使用绝对路径），执行安装命令，

## YUM包管理

yum是一个Shell前端软件包管理器，基于RPM包管理，它能够从指定的服务器自动下载RPM包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软件包（比如安装A软件，那你得先安装B软件，而YUM包管理解决了这一问题，它会先给你把B软件装了，然后再安装A软件）；

使用yum命令安装软件时，会从yum服务器下载对应的RPM文件，然后进行安装，这个yum服务器是在公网上的，不是由自己维护的，所以，使用yum命令安装软件需要联网；

使用yum命令安装软件比使用rpm命令安装软件更高效；

### 查询软件

在使用yum命令安装一个软件时，需要先查看yum仓库中是否有这个软件，语法为：

**yum list**

执行这个命令时会列出系统已安装的软件，以及yum仓库的软件（可以安装的软件）；

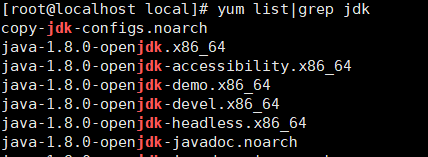
案例：

1. 查看yum服务器上所有的软件包



一般需要分页显示，因为是非常多的；

1. 如果想要判断某个软件包在yum服务器上是否存在，可以使用grep命令进行过滤，如查看jdk的软件包是否存在；



### 安装软件

使用yum命令安装软件的语法为：

**yum install 名称**

以安装火狐浏览器为例

1. 先查看yum服务器上是否有火狐浏览器的rpm文件



通过对查询结果进行过滤发现有；

1. 执行安装命令



# sudo

请参考<https://www.cnblogs.com/sparkdev/p/6189196.html>

## 介绍

简单来讲，sudo是一种权限管理机制，管理员可以授权一些普通用户去执行root执行的操作，而不需要知道root的密码；

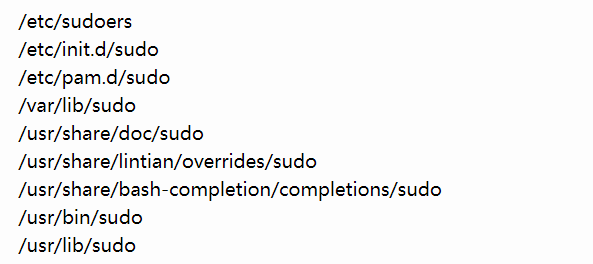
严谨点讲，sudo允许一个已授权用户以超级管理员或者其他用户的角色运行一个命令，当然，能够做什么不能做什么都是通过安全策略来指定的。sudo支持插件架构的安全策略，并能把输入输出写入日志。第三方可以开发并发布自己的安全策略和输入输出日志插件，并让他们无缝的和sudo一起工作。默认的安全策略记录在/etc/sudoers文件中。而安全策略可能需要用户通过密码来验证他们自己，也就是用户执行sudo命令时要求输入自己账号的密码，如果验证失败，sudo命令将会退出。

## 基本语法

sudo [-bhHpV][-s ][-u <用户>][指令]  
或  
sudo [-klv]

参数：  
  -b  在后台执行指令。  
  -h  显示帮助。  
  -H  将HOME环境变量设为新身份的HOME环境变量。  
  -k  结束密码的有效期限，也就是下次再执行sudo时便需要输入密码。  
  -l  列出目前用户可执行与无法执行的指令。  
  -p  改变询问密码的提示符号。  
  -s  执行指定的shell。  
  -u  <用户>  以指定的用户作为新的身份。若不加上此参数，则预设以root作为新的身份。  
  -v  延长密码有效期限5分钟。  
  -V  显示版本信息。  
  -S  从标准输入流替代终端来获取密码

## 相关文件



## 基本使用

系统默认创建一个名为sudo的组。只要把用户加入到这个组，用户就具有sudo的权限了。

至于如何把用户加入sudo组，你可以直接编辑/etc/group文件，如下：

# 防火墙常用指令

参考：<https://www.cnblogs.com/suruozhong/p/11589790.html>

1. 启动防火墙：**systemctl start firewalld**
2. 查看防火墙状态：**systemctl status firewalld**
3. 查看防火墙是否是开机自启动：**systemctl is-enabled firewalld**
4. 禁用防火墙开机自启动：**systemctl disable firewalld**
5. 开启防火墙开机自启动：**systemctl enable firewalld**
6. 关闭防火墙：**systemctl stop firewalld**
7. 查看防火墙版本： **firewall-cmd --version**
8. 查看防火墙状态： **firewall-cmd –state**
9. 重载防火墙，更新规则： **firewall-cmd --reload**
10. 开放80端口：**firewall-cmd --zone=public --add-port=80/tcp --permanent**

  （--permanent永久生效，没有此参数重启后失效）

11）查看所有开放的端口： **firewall-cmd --zone=public --list-ports**

12）关闭80端口：**firewall-cmd --zone=public --remove-port=80/tcp --permanent**

13）查看80端口是否开放：**firewall-cmd --zone=public --query-port=80/tcp**