### linux简介

linux是一种开源的、免费的操作系统，安装在计算机硬件上，用来管理计算机的硬件（cpu、内存、输入、输出……）和软件的资源的系统软件。和window操作系统类似。Linux系统比较注重系统的安全性，具备完善的权限管理机制；linux稳定性更好，相较于window，软件运行更稳定，不易出现卡顿；linux处理高并发的能力比window更强；linux没有优异的可视化操作界面（window牺牲了效率，做出了优异的可视化界面。linux侧重效率与性能，但是牺牲了可视化界面，毕竟一台计算机的资源是一定的，侧重一方，就要牺牲另一方，鱼和熊掌不可兼得）。window通常用于个人计算机，linux通常用于企业级应用的服务器上。

### linux的发行版本

linux内核系统程序诞生于1991年，后来很多软件开发组织一级大型的软件公司，在linux内核程序基础之上进行二次开发，陆续推出了很多不同版本的linux操作系统。包括**Ubuntu（乌班图）；RedHat（红帽）；CentOs；**红旗Linux（国产）；OpenSUSE等几十种版本；目前国内使用较多的是前面三种版本的操作系统。

### linux的目录结构（一切皆目录）

##### 1、linux只有一个根目录（ ‘/’ ）

linux只有一个根目录，里面存放所有的子目录，他不像window可以分成不同的盘。

##### 2、层级式的子目录

根目录 -> 一级子目录 -> 二级子目录 –> 三级子目录 ……

##### 3、默认的一级字目录

**root**：该目录为系统管理员目录，root是具有超级权限的用户

**bin -> usr/bin**: 存放系统预装的可执行程序，这里存放的可执行文件可以在系统的任何目录下执行(.sh脚本)。

**usr**：是linux的系统资源目录，里面存放的都是一些系统可执行文件或者系统依赖的文件库

**usr/local/bin**：存放用户自己的可执行文件，同样这里存放的可执行文件可以在系统的任何目录下执行(.sh脚本)。

**lib -> usr/lib**：这个目录存放着系统最基本的动态连接共享库，其作用类似于window里面的DLL文件，几乎所有的应用程序都需要用到这些共享库

**boot**：这个目录存放启动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

**dev**：dev是Device（设备）的缩写，该目录存放的是linux的外部设备，linux中的设备也是以文件的形式存在

**etc**：这个目录存放所有的系统管理所有要的配置文件(etc/profile配置环境变量)

**home**：用户的主目录，在linux中，每个用户都有一个自己的目录，一般该目录名以用户的账号命名，叫做用户的目录。用户登录以后，默认打开自己的目录

**var**：这个目录存放着不断扩充着的东西，我们习惯将那些经常被修改的文件存放在该目录下，比如运行的各种日志文件

**mnt**：系统提供该目录是为了让用户临时挂在别的文件系统，我们可以将光驱挂载在/mnt/上，然后进入该目录就可以查看光驱里的内容

**opt**：这是给linux额外安装软件所存放的目录。比如你安装Oracle数据库、nginx，tomcat、则就可以放到这个目录下，默认为空。相当于window系统中的program files目录

**tmp**：这个目录是用来存放一些临时文件的

### linux的远程操作

1. Xshell：linux的终端模拟软件
2. Xftp：文件传输软件
3. 命令行终端 ssh root@ip登录

### vi和vim编辑器

##### 简介：

linux提供的编辑器，用于创建&查看&编辑文本文件，相当于window里面的记事本功能。vim是vi的增强版本，功能更加完善。vi是最早期的编辑器。

##### 使用：

注意：使用vi或者vim操作文件，存在三种模式。

1. 一般模式：通过vim text.txt打开文件，这个时候出入一般模式，只允许查看内容，不允许编辑内容。
2. 编辑模式：在一般模式下，按i键或者a键或者I键或者A键即可进入编辑模式，此模式可以正常编辑文本内容，编辑结束后，按Esc键，返回至一般模式
3. 命令模式：在一般模式下，按：键即可进入到命令模式。常用有以下几种命令

q! + 回车键：不保存编辑的内容，并强制退出编辑器

wq! + 回车键：保存编辑的内容，并强制退出编辑器

! 表示强制退出

创建文件：vim test.txt（如果test.txt不存在，则创建文件并打开）

打开文件：vim test.txt（如果test.txt存在，则直接打开文件）

编辑文件：在一般模式下，按i键或者a键或者I键或者A键即可进入编辑模式

##### vi和vim编辑器常用的快捷键：

1、复制当前行：在一般模式下，按yy（连续两次y键），即可把光标所在的行（一行）内容复制到剪切板。

2、复制当前行及以下n行内容：在一般模式下，按nyy，即可将当前行及向下n行的内容复制到剪切板（n代表数字）

3、粘贴内容：在一般模式下，按p键，即可粘贴复制的内容

4、在文本文件中搜索关键字：在命令模式下，输入/和查找的关键字+回车键。按n可以按顺序定位到检索到的关键字

5、删除某一行内容：在一般模式下，按dd，即可删除光标所在的行内容

6、删除多行内容：按ndd，即可删除光标所在行下面n行内容（n代表数字）

7、撤销编辑的内容：在一般模式下，按u键即可实现撤销

8、显示行号：命令模式下，输入set nu + 回车即可显示行号，输入set nonu + 回车即可取消行号显示

### linux用户管理

##### linux用户简介：

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。root用户是系统默认创建的管理账号。由root创建普通的账号。

**添加用户（useradd [options] 用户名）：**

useradd xiaolu

1、执行此命令后，做了以下几件事：

2、在linux的用户列表里添加了一个叫xiaolu的用户；

在/home文件夹下，新增了一个xiaolu文件夹，作为xiaolu用户的默认工作目录。当xiaolu用户登录成功后，默认进入到/home/xiaolu目录下；

3、创建了一个组，组名叫xiaolu（在linux中，用户不能脱离组存在，如果我们创建了一个用户，未指定组，那系统将默认创建一个和当前用户同名的组，同时将该用户添加到创建的组内）

如果在创建用户时，需要自定义当前用户的工作目录（对应上述第二条的自定义配置），则执行以下命令：

useradd -d /home/xiaolu01 xiaolu

##### 给用户设置密码（passwd 用户名）：

passwd xiaolu + 回车

##### 删除用户（userdel [options] 用户名）：

userdel xiaolu

删除账号的同时，一般需要同步手动删除/home下面的xiaolu用户的工作目录，如果想用命令直接删除工作目录，则需要执行以下命令：

userdel -r xiaolu

##### 查看用户（id 用户名）：

id xiaolu

##### 切换用户（su 用户名）

su xiaolu

从权限高的用户切换到权限低的用户，不需要密码验证；从权限低的用户切换到权限高的用户，需要密码验证

##### 查看所有的用户列表：cat /etc/passwd

cat /etc/passwd

### linux的用户组管理

##### 简介：

linux中的组相当于角色的概念，可以对具有共性的用户角色进行统一管理（权限分配等）。每个用户至少属于一个组，也可以属于多个组。用户不能独立于组存在，权限控制是通过组进行分配的。root用户属于root组。

##### 添加组（groupadd 组名）：

groupadd dev

##### 删除组（groupdel 组名）：

groupdel dev

##### 把用户添加到某个组中（gpasswd -a 用户名 组名）：

gpasswd -a xiaolu dev

##### 把用户从组中移除（gpasswd -d 用户名 组名）：

gpasswd -d xiaolu dev

##### 创建用户的时候指定所属组【主组】（useradd -g 组名 用户名）：

useradd -g dev xiaolu

##### 查看当前用户所属哪个用户组：groups

groups // 返回当前用户所属组名

##### 查看所有的组列表：cat /etc/group

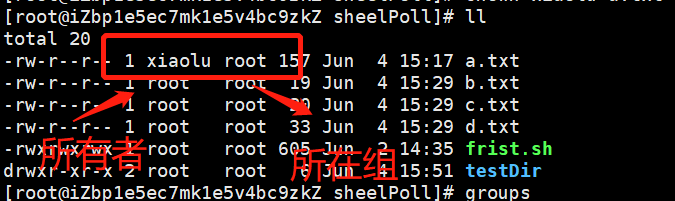
cat /etc/group

### linux的文件组管理

##### 简介：

linux中的每一个用户必须属于一个组，不能独立于组外，可以改变用户所属组。linux中的每个文件或目录也必须属于一个组，有所有者，所在的组，也可以改变文件所在组。与用户组不同的是，一个文件只能属于一个组。用户可以属于多个组。文件或者目录通过组来控制哪些用户可以对他进行哪些操作，即文件或者目录的访问&读写权限。在文件或者目录看来，在linux系统中，所有的用户分为3类：（1）文件或目录的所有者（默认情况下，所有者即创建者。所有者可以更改）；（2）同组用户，和文件或目录属于同一个组的用户；（3）其他组用户，既不是所有者也不是同组用户。

##### 查看文件的所有者和所在的组：ls -l 或者 ll



##### 修改文件&目录的所有者和所在组：chown 新的所有者[:新的组] 文件名 或者 chgrp 新的组名 文件名

chown xiaolu a.txt // 把a.txt的所有者改为xiaolu用户

chown xiaolu:xiaolu a.txt // 把a.txt文件的所有者改为xiaolu，所有组改为xiaolu

chown xiaolu:xiaolu testDir // 把testDir目录的所有者改为xiaolu，所有组改为xiaolu，注意，如果testDir中有子目录，那在执行这段指令的时候，默认只改变了testDir的所有者和所在组，其所有子目录默认不同步更改，还保持原有的所属性。那如果我们需要所有的子目录也同步更改呢？需要加-R参数，递归修改testDir目录的所有子目录，chown -R xiaolu:xiaolu testDir

chgrp xiaolu a.txt // 把a.txt文件的所有组改成xiaolu

chgrp xiaolu testDir // 把testDir目录的所有组改成xiaolu

文件或者目录的所有者，默认是创建者

chown 和 chgrp的区别在于，chown可以一个命令同时修改所有者和所在组，但是chgrp只能修改所在组。

### linux的文件&目录权限管理

##### 权限的基本介绍：

在linux中，每一个文件和目录都有自己的访问权限，这些权限通过组来管理。

linux中，任何文件和目录都有三种权限：（1）读（Read）（2）写（Write）（3）执行[.sh脚本的执行权限或者目录的cd权限]（Execute）

**对于文件而言，这三种权限的含义是：**

读：可以读取或者查看文件的内容，可执行cat，more，head，Less……

写：可以编辑文件的内容，可执行vi，vim……

执行：如果该文件是可执行文件(.sh)，那该文件可以直接运行，可执行./nginx.sh等.sh文件

**对目录而言，这三种权限的含义是：**

读：可以读取、查看目录里面的内容，可执行ls，ll……

写：可以修改目录中的内容，对目录中的子目录增删改查或者cd命令，可执行cp，mv，touch，vim，vi，rm……

执行：是否能进入到改目录，可执行cd，rm等命令。

##### 文件或者目录权限控制：

在linux中，任何一个文件或者目录的权限，与文件和当前用户的关系有关。

不同的关系用户，给与了不同的权限集合。对应文件与用户的三种关系，这里一共分为3中权限类型：

1）所有者权限

文件或者目录的所有者对该文件的权限，使用r，w，x分别表示读，写，执行的权限。比如qwx：表示读写执行的权限；r-x：表示读执行的权限；w--： 表示只拥有写的权限；---：表示没有任何权限

2）同组用户权限

与文件或者目录同组用户的权限，使用r，w，x分别表示读，写，执行的权限。

3）其他组用户权限

与文件或者目录不同组用户的权限，使用r，w，x分别表示读，写，执行的权限。

##### 查看文件或者目录的权限：ll 或者 ls -l



##### 修改文件或者目录的权限：chmod [(u，g，o，a)(-，+，=)(r，w，x)] 文件名

chmod u-r a.txt // 对a.txt文件的所有者删除读的权限

chmod u-r,g+x a.txt // 对a.txt文件的所有者删除读的权限，同组者添加执行的权限。逗号分隔可以执行多个权限操作。

chmod g-rwx a.txt// 表示对a.txt的同组者设置读，写，执行的权限

用r，w，x分别代表读，写，执行的权限。用”-“号，“+”号和“=”号代表删除，新增，设置权限。例如：+w代表了对目标用户新增写的权限；-x：对目标用户删除执行的权限。目标用户也是通过字母表示，u代表拥有者，g代表同组用户，o代表其他组用户，a代表所有用户。例如：u-w：表示对拥有者删除写的权限；g+x：表示对同组用户添加执行的权限。如果要同时操作多个权限，可以执行 chmod u-w,g+x a.txt，多个权限操作的时候，用逗号分隔即可。

##### 用数字的方式改变文件或者目录的权限：chmod 数字 文件名

每一个权限也都可以用一个数字来表示。类似上面说的r，w，x，用数字表示的话4：读；2：写；1：执行。例如rw-对应数字的表示是420，对于无权限的占位，数字的方式使用0（不同于字母使用-）。对比上面字母的表示形式，数字表示的形式是这样的，例如 chmod 760 a.txt，意思是对a.txt文件，拥有者权限是7（4+2+1，可读，可写，可执行）；同组用户权限是6（4+2，可读，可写）；其他组用户是0（0+0+0，没有任何权限）。相对于字母设置的形式，这种数字的更为简介，可以通过几个简单的数字，就设置了三种用户关系的权限。但是如果是字母的形式，则要通过（u，g，o，a）（+，-，=）（r，w，x）来设置，复杂度会增加。

### linux的帮助命令

在使用linux过程中，如果遇到不熟悉的指令，一个是可以到网上去查找，另一个是可以使用帮助命令。

##### 1、man 命令名（按回车翻行，按空格翻页，按q退出查看）

用来查看linux系统手册上的帮助信息

##### 2、help 命令名

用来查看命令的内置帮助信息

### linux关机、重启

##### 关机：

shutdown now 系统立即关机

shutdown -h 20：00 系统会在20：00关机

shutdown -h +10 系统会在十分钟后关机

##### 重启：

shutdown -r now 系统立即重启

shutdown -r +30 系统会在30分钟后重启

reboot 系统立即重启

### linux中的文件和目录操作的命令：

##### 查看当前所在的目录：pwd

查看当前所在的目录

##### 查看当前或者指定目录信息：ls [options] [指定目录]

查看 当前或者指定(未输入指定目录时，默认查看当前目录) 目录下的子目录及文件列表。默认列表展示形式为平铺，如果想以列表形式展现，需要执行**ls -l [指定目录]**；如果想查看当前或指定目录的所有文件（包括隐藏文件或者虚拟目录），需要执行**ls -a [指定目录]**，options中的命令可以组合使用，**比如ls -al [指定目录]**

##### 切换目录：cd 目录名[相对路径 || 绝对路径]

cd /etc/nginx 或者 cd ../../etc

有一个特殊的绝对路径表示符~，~表示当前用户的工作目录路径。比如/home/xiaolu

..表示从当前目录切换至上一级目录下

##### 创建目录：mkdir [options] 目录名[相对路径 || 绝对路径]

mkdir /home/test 或者 mkdir test

如果需要创建多级目录，则需要使用-p参数，比如，用户当前在/home目录下，想创建test1目录，同时test1下再创建test2目录，那直接执行mkdir test1/test2是不行的，mkdir只能创建一级目录，这个时候需要执行mkdir –p test1/test2，这样就会递归生成我们需要的文件夹

##### 删除目录：rmdir [options] 目录名

rmdir 目录名：删除一个空文件夹

rmdir -rf 目录名：删除一个非空文件夹

##### 创建文件（一个或者多个）：touch 文件名列表(多个文件时，文件名用空格分开)

touch命令支持相对路径和绝对路径，使用实例如下（相对路径）

创建一个文件 touch test.sh

创建多个文件 touch test1.sh test2.sh test3.sh

上面使用过的vi/vim也可以实现创建文件，但是这两个指令有些限制，一个是创建文件后直接打开了，另外的限制是一次只能创建一个文件

##### 复制文件：cp 源文件 目标文件

源文件、目标文件支持相对路径和绝对路径

cp /home/test.sh testDir // 把home目录内的test.sh 复制到testDir目录内

cp /home -r testDir // 把home目录内的所有内容 复制到testDir目录内

cp指令，默认只支持复制某个文件，当我们要复制某个文件夹时，且源文件夹内有子目录内容，这个时候只用cp指令就不行了，需要加上-r参数，表示需要递归复制所有内容

如果我们复制了一个文件，但是这个文件在目标目录中已经存在了，那会先询问用户是否覆盖

##### 删除文件或者目录：rm 文件名||目录名

文件名||目录名支持相对路径和绝对路径

rm test.sh // 删除test.sh文件，删除过程中会二次确认是否删除

rm -f test.sh // 强制删除文件，删除过程中进行二次确认

rm命令默认是无法删除目录（文件夹）的，如果需要删除目录，则需要执行：

rm -r testDir // 递归删除testDir下面的所有内容

rm -rf testDir // 如果文件夹中内容很多，递归删除的时候，每个文件都需要二次确认，很麻烦，这是时候执行-rf，强制递归删除testDir目录

##### 移动目录或者文件：mv 源文件||目录 目标目录

源文件||目录 目标目录支持绝对路径和相对路径

mv /home/test.sh testDir // 移动home目录下的test.sh至testDir目录下

mv /home testDir // 移动home目录（包括home）所有内容至testDir目录下

mv test.sh test1.sh // 字面上理解，是test.sh文件移动到test1.sh上，但是test1.sh并不是目录，是一个文件，这个时候其实就是重命名的效果，将test.sh重命名成test1.sh

##### 查看文件内容：cat 文件名 || more 文件名 || less 文件名 || head 文件名 || tail 文件名

cat test.sh // 查看test.sh文件的内容

more test.sh // 查看test.sh文件的内容

less test.sh // 查看test.sh文件的内容

head test.sh // 查看test.sh文件的开头部分的内容，有一个常用的参数 **-n** 用于显示行数，默认为 10。

tail test.sh // 查看test.sh文件的内容

**cat**：cat指令是将目标文件的内容全部显示在终端，供查看。cat -n test.sh，-n参数可以是内容显示行号

**more**：more指令是以一页一页的形式展现目标文件的内容，可以通过回车键翻行，空格键翻页，但是more不能向前翻页，只能向后翻页，q退出查看

**less**：less指令类似more，是以一页一页的形式展现目标文件的内容，但是比more更加强大，可以通过 [pageup] [pagedown]按键实现向前和向后翻看内容。除此之外，在 less 里头可以拥有更多的搜索功能，不止可以向下搜，也可以向上搜（/字符串：向下搜索“字符串”的功能 ； ？字符串：向上搜索“字符串”的功能），q退出查看

**head**：用于查看文件的开头部分的内容，有一个常用的参数 **-n** 用于显示行数，默认为 10，即显示 10 行的内容

**tail**：功能类似cat，但是可以查看指定后几行内容，具体可查看文档

### 输出系统变量或者某个常量的值到终端：echo

echo $PATH

C:\Users\lupp\AppData\Local\Temp\1622790440(1).png

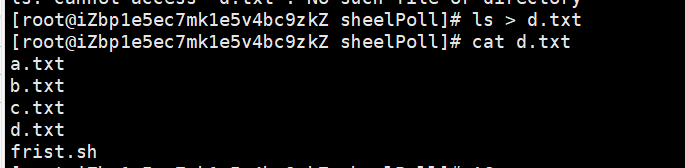
echo hello world (输出常量，此用法多用于.sh脚本编程)

C:\Users\lupp\AppData\Local\Temp\1622790699(1).png

### 向指定文件中填写前一个查看命令的输出结果：查看命令 文件名 > 文件名

cat a.txt > b.txt // 将a.txt查看的内容填写到b,txt中。如果b.txt已经存在，则直接添加内容；如果b.txt文件不存在，则创建b.txt文件并添加内容，效果类似跨文件复制内容，注意，a.txt查看的内容将全量覆盖b.txt中原有的内容。

ls > a.txt // >配合ls命令执行，则会将ls查询的结果（子目录名列表）填写至a.txt中



### linux关于时间和日期的操作命令：date [options] && cal [options]

date // 查看系统当前的完整的日期和时间

date +%Y // 查看系统当前的年份

date +%m // 查看系统当前的月份

date +%d // 查看系统当前的日期

date "+%Y-%m-%d %H:%M:%S“ // 按照yyyy-MM-dd HH:mm:ss 格式显示系统时间

date -s “2021-06-14 00:00:00” // 设置当前的系统时间

cal // 查看系统当前月份的日历

cal 2020 // 查看2020年的所有日历

### linux中关于搜索文件或目录的命令：find [options] 目标文件名||目录名||\*关键字 或者locate 目标文件名||目录名||\*关键字

find a.txt // 查找当前目录及子目录下a.txt文件是否存在

find testDir // 查找当前目录及子目录下testDir目录是否存在

find \*.txt // 查找当前目录及子目录下所有.txt后缀的文件

find \*keyword\* // 查找当前目录及子目录下所有文件名上有keyword的文件

find /etc \*.txt // 查找指定目录下/etc所有的.txt文件

find /etc -size +5M // 查找指定目录/etc中所有大于5M的文件

find /etc -size -5M // 查找指定目录/etc中所有小于5M的文件

find /etc -size 5k // 查找指定目录/etc中所有等于5k的文件

find /etc -user xiaolu // 查找指定目录/etc中所有xiaolu用户创建的文件

locate \*.txt // 全局查找所有后缀名为txt的文件

locate aa.txt // 全局查找所有命名为aa .txt的文件

注意：find指令默认按名字搜索，也就是说，默认find后面是-name搜索规则

find：find指令会按照我们指定的所有范围，按条件查找所有目录及子目录的内容，并将结果返回

locate：locate指令的查找并不会真实遍历物理文件去查找，他是直接去linux系统中的文件系统数据库，在数据库的文件目录树中查找，并将结果返回。locate的查找范围是全局的，并且只能按名称搜，其查找效率要高于find指令。但是locate指令并不常用，由于其查找机制依赖目录树，而在linux系统中，为了系统运行效率，目录树并不是实时更新的，一般是按照一定的算法批量更新，也就是会存在内容滞后的现象，导致查找结果不准确，所以不是很常用，如果一定要用，需要在使用locate之前，执行updatedb指令，通知linux数据库更新，然后再查找，这样保证查找结果正确性。

### linux中关于过滤（查看过滤，搜索过滤等）的命令：grep(搜索命令 | grep [options] 过滤条件 或者 查找命令 | gerp [options] 过滤条件 或者ls | grep [options] 过滤条件…)

cat a.txt | grep food // 查看a.txt的内容，并且过滤出包括food关键字的内容输出

find \*.txt | grep keyword // 查找当前目录及子目录下所有.txt后缀且过滤出文件名中包含keyword的文件，返回给用户

find -size -5M | grep keyword // 查找当前目录及子目录下所有小于5M且过滤出文件名中包含keyword的文件，返回给用户

grep命令过滤时默认区分大小写，如果我们不需要区分大小写，则需增加-i参数，例如find -size -5M | grep -i keyword；-n参数，可以使查找的内容带有行号

**注意：| grep [options] 过滤条件是linux中通用的过滤命令，在大部分查找&查询指令中都可以使用gerp指令对结果进行过滤**

### linux中的压缩和解压缩命令：gzip；gunzip；zip；unzip；tar

##### 压缩或者解压单个文件（不能操作目录）：gzip& gunzip

gzip 文件名 // 压缩文件，压缩的结果默认是一个.gz的压缩包，同时将原有压缩的目标文件自动删除

gunzip 文件名 // 解压文件

##### 压缩（打包）或者解压多个文件和目录：zip&unzip

zip目标压缩包后缀类型 （目标压缩包名称，后缀类型通常选择.zip类型） 文件名||目录名

zip test.zip testDir // 将textDir目录压缩至test.zip

zip test.zip a.txt // 将a.txt文件压缩至test.zip

zip test.zip a.txt testDir // 将a.txt文件和testDir目录压缩至test.zip

unzip 文件名 // 解压文件至当前目录下

unzip 文件名 -d testDir // 解压文件至testDir（指定）目录下

##### 压缩（打包）或者解压多个文件和目录：tar（功能类似zip&unzip）

tar [options] xxx

options简介（tar只有一个命令，通过参数区分压缩还是解压）：

-c：打包或者压缩 // eg：tar -c test.tar.gz a.txt testDir（把a.txt文件和testDir目录压缩打包至test.tar.gz）

-x：解压 // eg：tar -x test.tar.gz（将test.tar.gz解压至当前目录中），如果想将解压的结果，放到指定的目录中，则需执行tar -xf test.tar.gz -C testDir，这样就将test.tar.gz中的内容解压至testDir目录中了

当执行tar -c时，-c后面需要指定压缩后的包名称及后缀，一般使用tar压缩的话，其后缀约定俗称的以.tar.gz作为后缀。

### linux中的网络管理

在linux中的配置文件中配置ip地址：/etc/sysconfig/network-scripts/ifcfg-ens33

### linux中的进程管理

##### 进程介绍：

在linux系统中，每个执行的程序都是一个称为一个进程，每个进程都会分配一个ID号码，每个进程都要占用一个端口，每一个进程都会对应一个父进程。

进程有两种运行方式：前台和后台，前台方式是目前用户可以在前台操作的，后台方式是实际在运行，但是用户在前台看不见。

一般系统的服务都是以后台进程的方式存在，而且都会常驻在系统中，知道关机才结束。

##### 查看进程：ps

ps // 查看当前系统运行的用户应用级别的进程信息，系统后台的进程并不会显示

ps -e // 查看当前系统运行的所有进程

ps -ef // 以全格式的形式查看当前系统运行的所有进程，-f可以显示更多的进程信息

##### 关闭进程：kill

kill -9 PID // 强制杀死进程

kill -15 PID // 安全杀死进程

kill -1 PID // 重启进程

### linux中服务的管理

##### 服务介绍：

服务是支持linux运行的 一些必要的程序，本质上也是进程，叫守护进程。守护进程通常默默的运行在后台，为应用程序提供必要的支撑。比如sshd，防火墙等。

##### 服务操作：systemctl [start|stop|restart|reload|status|enable] 服务名称（CentOS7之前，systemctl是service）

systemctl start firewalld // 启动防火墙服务

systemctl stop firewalld // 关闭防火墙服务

systemctl restart firewalld // 重启防火墙服务

systemctl reload firewalld // 重新加载防火墙服务(一般改配置文件后需要重新加载)

systemctl status firewalld // 查看防火墙服务的状态

systemctl enable firewalld // 设置防火墙服务开机启动

### linux中的软件安装包管理

##### RPM介绍

软件的开发商，会将开发好的软件通过打包工具打包成.rpm的软件包发布到网络上，用户下载软件包后，即可使用rpm开始安装软件

##### 使用RPM

**查看当前系统中已经安装的rpm包**

rpm -qa

**卸载rpm软件包**

rpm -e 软件包关键字，例如：rpm -e firefox // 卸载火狐浏览器

**安装rpm软件包**

1、去官网下载.rpm安装包至linux系统中

2、rpm -ivh \*\*\*.rpm，例如：rpm -ivh firefox.rpm // 安装火狐浏览器，其中-i表示安装，-v表示提示，-h表示进度条。

##### YUM简介

yum包本质也是rpm包，只不过是在rpm包的基础上做了改进优化。相对于rpm包安装时需要用户自己去网络上下载相应的软件包（麻烦，耗时，且如果一个rpm包依赖其他的rpm包，用户需要逐一下载完所有rpm包才能完成安装，这个过程非常麻烦，同时容易遇到问题），yum自己提供了一个源，里面基本包含了世界上所有的安装包。用户通过yum安装的时候，不需要自己去网上下载，yum会自动去自己的源里获取安装包（获取当前软件包并分析其所有的依赖关系，最后找到所有的安装包一起下载下来），然后再进行安装，非常的省事。yum安装有一个要求，就是必须要有外网环境。

##### YUM使用

**查看当前系统中安装的所有rpm包**

yum list installed // 查看当前系统中安装的所有rpm包

**卸载rpm软件包**

yum remove 软件包全名 // 例如：yum remove firefox.x86\_64

安装rpm软件包

yum -y install 软件包关键字 // 例如 yum -y install firefox