

# 一、Linux运行环境搭建

## 操作系统

---

操作系统（Operating System，简称OS），是管理和控制计算机硬件与软件资源的计算机程序，是直接运行在“裸机”上的最基本的系统软件，任何其他软件都必须在操作系统的支持下才能运行。

操作系统身负注入管理与配置内存、决定系统资源供需的优先次序、控制输入与输出设备、操作网络与管理文件系统等基本事务。

操作系统分两大阵营：**Windows系统** 和 **Unix的衍生系统（Linux、MacOS等）**

## Linux系统

---

Linux是一套免费使用和自由传播的类Unix操作系统，是一个基于POSIX和UNIX的多用户、多任务、支持多线程和多CPU的操作系统，Linux继承了Unix以网络为核心的设计思想，是一个性能稳定的多用户网络操作系统。

Linux包含了Unix的全部功能和特性。主要特性有：

- 开放性：系统遵循世界标准规范
- 多用户：系统资源可以被不同用户各自拥有使用，每个用户可以对自己的资源进行管理、设置权限等
- 多任务：同时执行多个程序，而且各个程序的运行互相独立。
- 良好的界面：同时具有命令行界面以及图形化界面
- 丰富的网络功能：Linux在通信和网络功能方面优于其他操作系统，内置了很丰富的免费网络服务软件、数据库和网页的开发工具
- 可靠的系统安全：采取了许多安全技术措施，包括对读写控制、带保护的子系统、审计跟踪、核心授权等
- 良好的可移植性：将操作系统从一个平台转移到另一个平台仍然能按其自身的方式运行

**总结：免费、安全、可靠、稳定、多平台**

## Linux有很多的发行版本

- CentOS
- Redhat
- Suse
- Ubuntu
- Debian
- 红旗

源生的Linux是开源的，创始人林纳斯·托瓦兹，在GNU组织中，很多组织在维护，加了不同的功能，衍生出很多版本，有的开源有的收费。

- 商业版做的比较成功的：Redhat
- 免费版：生产环境下：CentOS，非常稳定，可以常年待机，消耗资源少

# 安装操作系统

## 1) 安装虚拟机软件

- VMware Workstation
- VMware Workstation Player
- Virtual Box

## 2) 安装CentOS

### 1、创建虚拟机

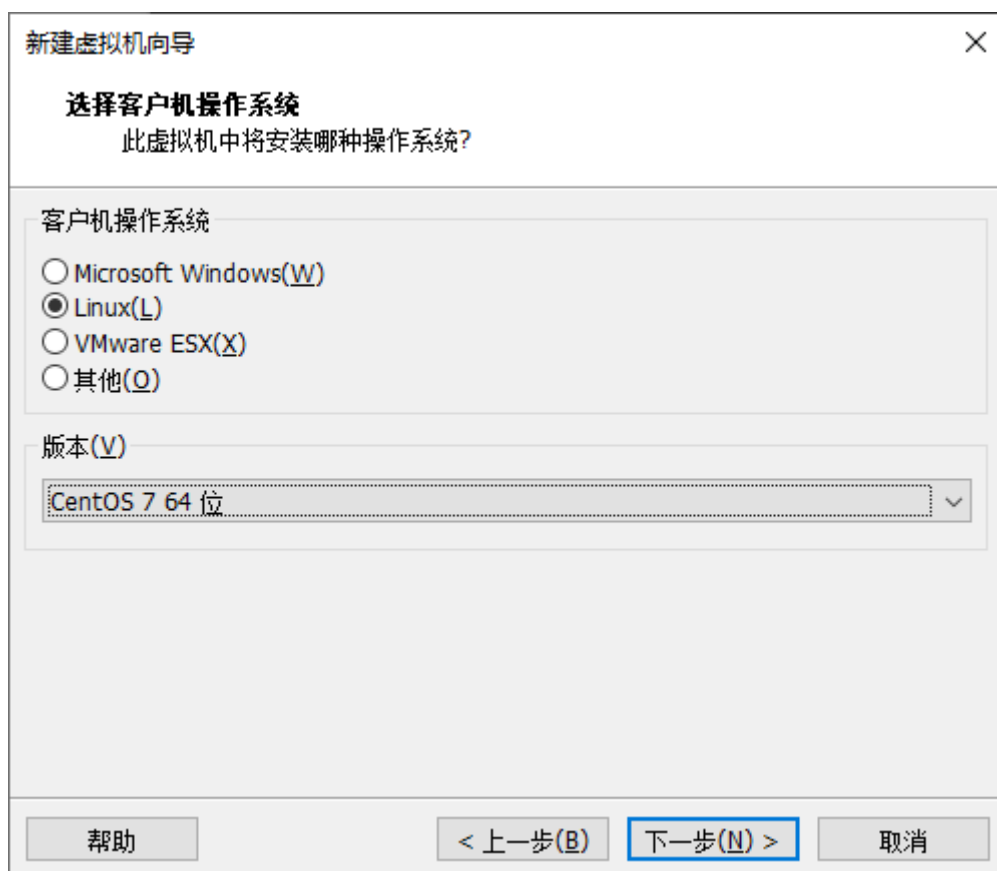
1、新建虚拟机，选择自定义类型



2、在下一步后，选择默认直接下一步，之后在安装来源中选择稍后安装



3、客户机操作系统中选择Linux，版本选择CentOS7



4、虚拟机名称可以自行修改，为了和控制台版本区分，可以加点说明

新建虚拟机向导

命名虚拟机

您希望该虚拟机使用什么名称？

虚拟机名称(V):

CentOS 7 64 位 GUI

位置(L):

D:\Virtual Machines\CentOS 7 64 位 GUI

浏览(R)...

在“编辑”>“首选项”中可更改默认位置。

< 上一步(B)

下一步(N) >

取消

5、处理器数量选1（一般机器也就1个CPU），内核数量可以设置本机的一半，例如本机的CPU有4个内核，为了保证其他程序的运行，就分配两个给虚拟机，全部分配也是可以的

新建虚拟机向导

处理器配置

为此虚拟机指定处理器数量。

处理器

处理器数量(P):

1

每个处理器的内核数量(C):

2

处理器内核总数:

2

帮助

< 上一步(B)

下一步(N) >

取消

6、内存上Linux系统运行消耗较少，可以根据实际情况分配，程序推荐1G，可以设置多一些

新建虚拟机向导

此虚拟机的内存  
您要为此虚拟机使用多少内存？

指定分配给此虚拟机的内存量。内存大小必须为 4 MB 的倍数。

64 GB -

32 GB -

16 GB -

8 GB -

4 GB -

2 GB -

1 GB -

512 MB -

256 MB -

128 MB -

64 MB -

32 MB -

16 MB -

8 MB -

4 MB -

此虚拟机的内存(M):

2048 MB

最大推荐内存:  
13.4 GB

推荐内存:  
1 GB

客户机操作系统最低推荐内存:  
512 MB

帮助

< 上一步(B)

下一步(N) >

取消

7、网络类型、IO控制器、虚拟磁盘类型、选择磁盘均选择默认

新建虚拟机向导

网络类型  
要添加哪类网络？

网络连接

☐ 使用桥接网络(R)  
为客户机操作系统提供直接访问外部以太网网络的权限。客户机在外部网络上必须有自己的 IP 地址。

☒ 使用网络地址转换(NAT)(E)  
为客户机操作系统提供使用主机 IP 地址访问主机拨号连接或外部以太网网络连接的权限。

☐ 使用仅主机模式网络(H)  
将客户机操作系统连接到主机上的专用虚拟网络。

☐ 不使用网络连接(I)

帮助

< 上一步(B)

下一步(N) >

取消

新建虚拟机向导

选择 I/O 控制器类型

您要使用何种类型的 SCSI 控制器?

I/O 控制器类型

SCSI 控制器:

☐ BusLogic(U) (不适用于 64 位客户机)

☒ LSI Logic(L) (推荐)

☐ LSI Logic SAS(S)

帮助

< 上一步(B)

下一步(N) >

取消

新建虚拟机向导

选择磁盘

您要使用哪个磁盘?

磁盘

☒ 创建新虚拟磁盘(V)

虚拟磁盘由主机文件系统上的一个或多个文件组成，客户机操作系统会将其视为单个硬盘。虚拟磁盘可在一台主机上或多台主机之间轻松复制或移动。

☐ 使用现有虚拟磁盘(E)

选择此选项可重新使用以前配置的磁盘。

☐ 使用物理磁盘 (适用于高级用户)(P)

选择此选项可为虚拟机提供直接访问本地硬盘的权限。需要具有管理员特权。

帮助

< 上一步(B)

下一步(N) >

取消

8、磁盘容量可以指定为默认值，20G足够CentOS使用了（这里是最大，实际占据主机的容量其实不到，除非将虚拟机塞满）

新建虚拟机向导

指定磁盘容量  
磁盘大小为多少?

最大磁盘大小 (GB)(S):

针对 CentOS 7 64 位 的建议大小: 20 GB

☐ 立即分配所有磁盘空间(A)。  
分配所有容量可以提高性能，但要求所有物理磁盘空间立即可用。如果不立即分配所有空间，虚拟磁盘的空间最初很小，会随着您向其中添加数据而不断变大。

☐ 将虚拟磁盘存储为单个文件(Q)  
☒ 将虚拟磁盘拆分成多个文件(M)  
拆分磁盘后，可以更轻松地在计算机之间移动虚拟机，但可能会降低大容量磁盘的性能。

帮助 < 上一步(B) 下一步(N) > 取消

9、之后点击下一步（默认选择）后完成虚拟机创建。

## 2、选择操作系统镜像文件

1、下载CentOS7系统镜像文件（<https://www.centos.org/download/>）

# CentOS Linux

8 (2011)

7 (2009)

| ISO                  | Packages | Others                       |
|----------------------|----------|------------------------------|
| x86_64               | RPMs     | Cloud   Containers   Vagrant |
| ARM64 (aarch64)      | RPMs     | Cloud   Containers   Vagrant |
| IBM Power BE (ppc64) | RPMs     | Cloud   Containers   Vagrant |
| IBM Power (ppc64le)  | RPMs     | Cloud   Containers   Vagrant |
| ARM32 (armhfp)       | RPMs     | Cloud   Containers   Vagrant |
| i386                 | RPMs     | Cloud   Containers   Vagrant |

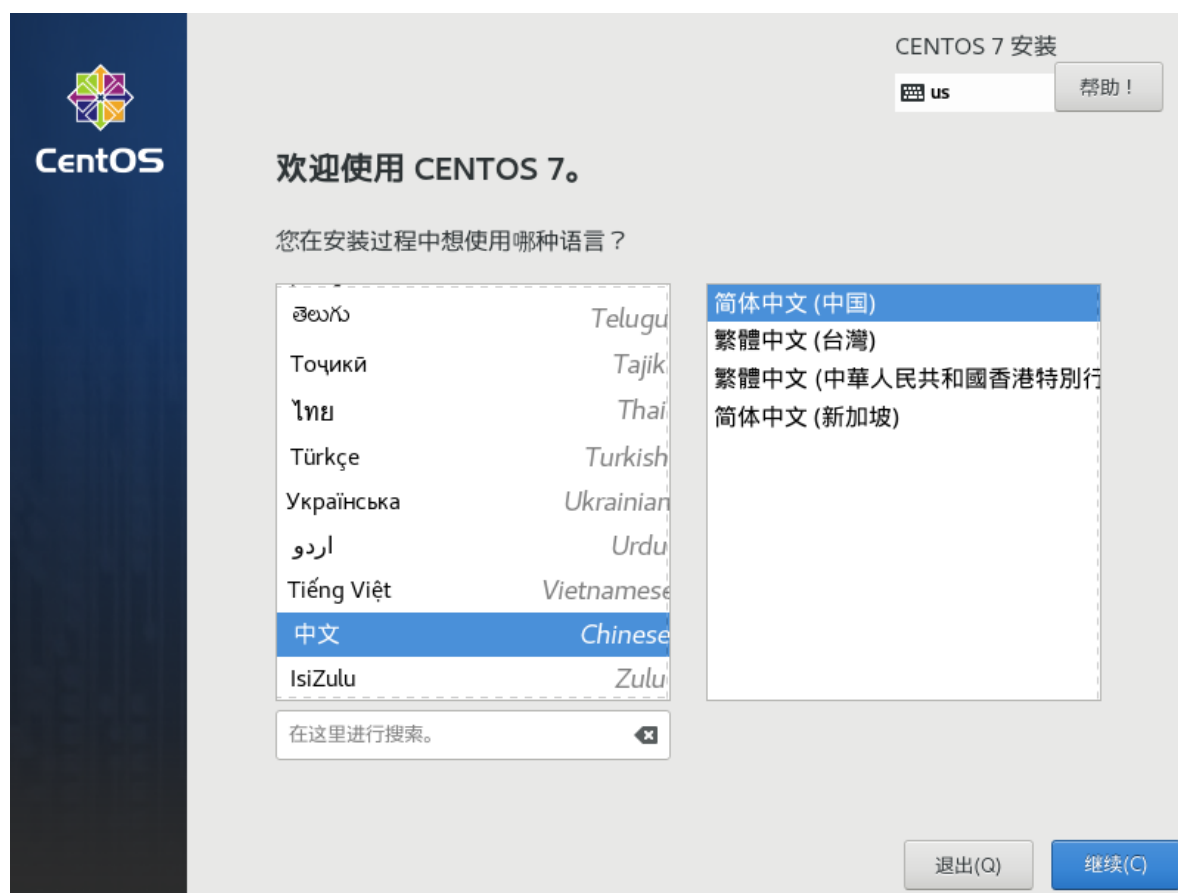
2、下载ISO镜像文件后，在虚拟机设置中选择该文件



之后就可以启动虚拟机进行安装了

### 3、安装CentOS 7

#### 1、选择语言



2、安装信息摘要中，点击想要定制的部分进行定制，其中需要修改【软件选择】，默认最小安装则会安装 **控制台版本**，不带其他工具。初学者可以使用图形化界面操作Linux系统，不过在生产环境中，为了减少不必要的系统消耗，通常采用最小安装，使用控制台进行管理。





点击完成返回。

3、点击【安装位置】，进行系统分区

安装目标位置

CENTOS 7 安装

完成(D)

cn

帮助！

设备选择

选择要在其中安装系统的设备。点击主菜单中的“开始安装”按钮前不会对该设备进行任何操作。

本地标准磁盘

20 GiB

VMware, VMware Virtual S

sda / 20 GiB 空闲

不会对未在此处选择的磁盘进行任何操作。

专用磁盘 & 网络磁盘

添加磁盘(A)...

不会对未在此处选择的磁盘进行任何操作。

其它存储选项

分区

☒ 自动配置分区(U)。☐ 我要配置分区(I)。

☐ 我想让额外空间可用(M)。

加密

[完整磁盘摘要以及引导程序\(F\)...](#)

已选择 1 个磁盘；容量 20 GiB；20 GiB 空闲 [刷新\(R\)](#)

初学者建议直接点击完成，采用【自动配置分区】方案，也可参考网络上其他资料推荐的分区方案。

4、【KDUMP】并不关心其作用，可以关闭

5、点击【网络和主机名】，进行网络配置

注意如下图，若此时显示“以太网被拔出”不可用状态，检查系统服务，是否VMWare的关于网络的服务被关闭了

网络和主机名(\_N)

CENTOS 7 安装

完成(D)

cn

帮助！

以太网 (ens33, 被拔出)

Intel Corporation 82545EM Gigabit Ethernet Controller

以太网 (ens33)

不可用

硬件地址 00:0C:29:8D:2F:81

速度 1 Mb/s

子网掩码 127.0.0.1

打开

+

-

配置(O)...

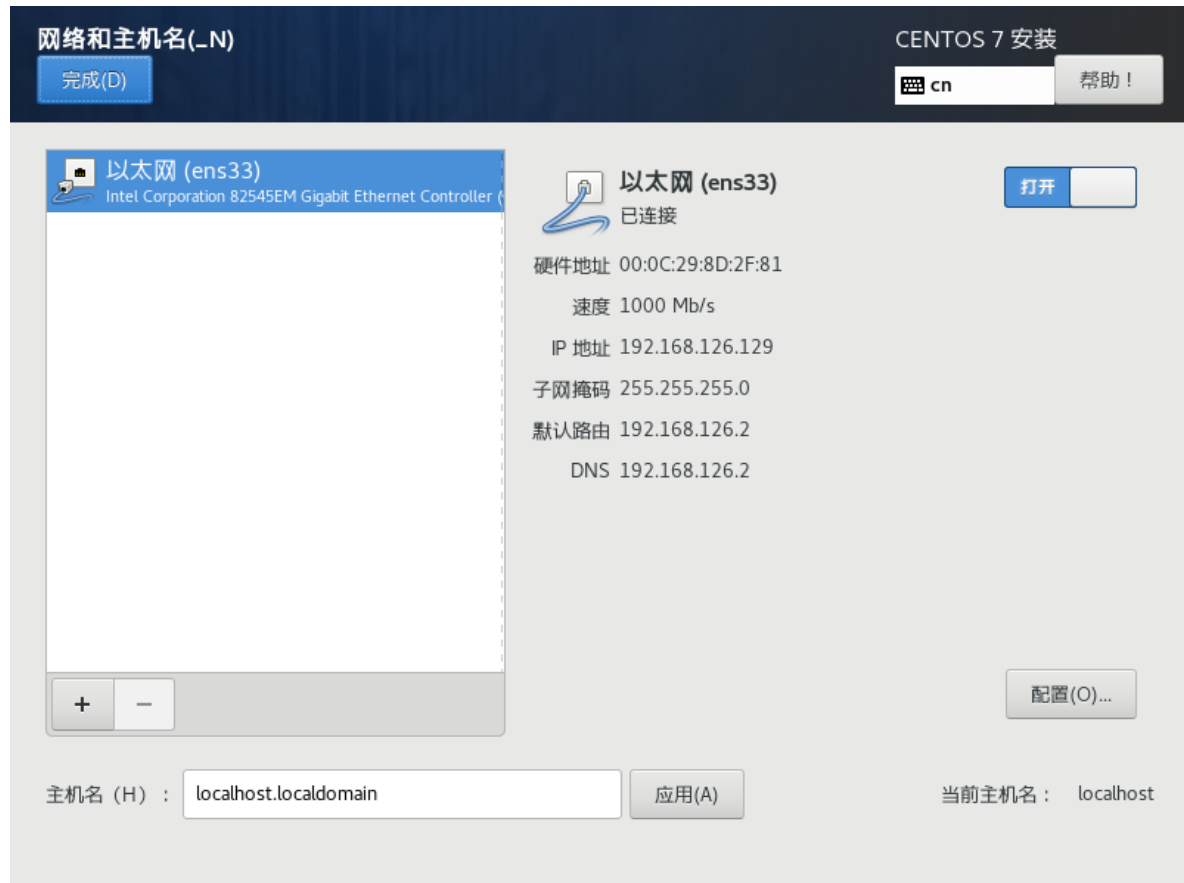
主机名(H) : localhost.localdomain 

应用(A)

当前主机名 : localhost

|                                |                                       |      |    |
|--------------------------------|---------------------------------------|------|----|
| VMware Authorization Service   | 用于启动和访问虚拟机的授权及身份验...                  | 正在运行 | 手动 |
| VMware DHCP Service            | DHCP service for virtual networks.    |      | 手动 |
| VMware NAT Service             | Network address translation for vi... |      | 手动 |
| VMware USB Arbitration Service | Arbitration and enumeration of US...  |      | 手动 |
| VMware Workstation Server      | 用于虚拟机注册和管理的远程访问服务...                  |      | 手动 |

启动这两个服务，再次点击【网络和主机名】：



打开网络后完成。

6、点击【开始安装】，进入配置页



点击【ROOT密码】设置密码：



注意下方提示，如果密码过于简单，需要点击两次【完成】

7、之后等待安装完成。



完成安装后点击重启。

```
CentOS Linux 7 (Core)
Kernel 3.10.0-1062.el7.x86_64 on an x86_64

localhost login: root
Password:
[root@localhost ~]#
```

登录系统后显示的是：【用户名@主机名】的形式，中括号中剩下的部分表示当前路径 这里的 `~` 表示当前用户的 `home` 目录，后面的符号 `#` 代表 `root` 用户，普通用户将以 `$` 符号表示

## 云服务器

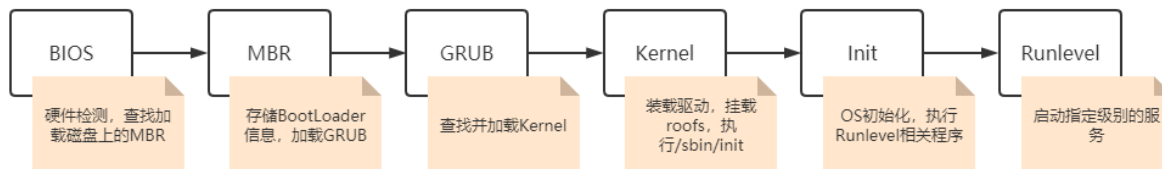
有条件的同学可以开通云服务器进行学习和研究，一般阿里云和腾讯云、华为云都提供学生云服务器，以学生身份购买比较优惠。

云服务器省略了安装虚拟机和操作系统的过程，通过配置直接选择安装的服务器的目标版本。

## 二、Linux系统基本操作

### Linux系统启动过程

Linux系统的启动流程如下：



- Runlevel
  - runlevel，运行级别，不同的级别会启动的服务不一样，init会根据定义的级别去执行相应目录下的脚本，Linux的启动级别分为以下几种
    - 0: 关机模式
    - 1: 单一用户模式(直接以管理员身份进入)
    - 2: 多用户模式 (无网络)
    - 3: 多用户模式 (命令行)
    - 4: 保留
    - 5: 多用户模式 (图形界面)
    - 6: 重启

## XShell工具连接Linux

对于很多在Windows上进行Linux开发和运行维护的人员来说，远程连接访问工具必不可少。现在主流的较为强大的远程连接工具有：

- SecureCRT
- Xshell
- MobaXterm
- ...

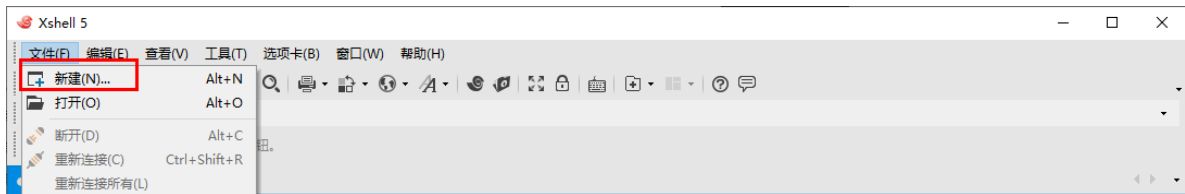
利用远程连接访问工具可以在Windows界面下用来访问远端不同系统下的服务器，从而比较好的达到远程控制终端的目的。

Xshell的使用者大部分为自由开发者和学生使用，可以申请免费版本用于 **非商业用途**。

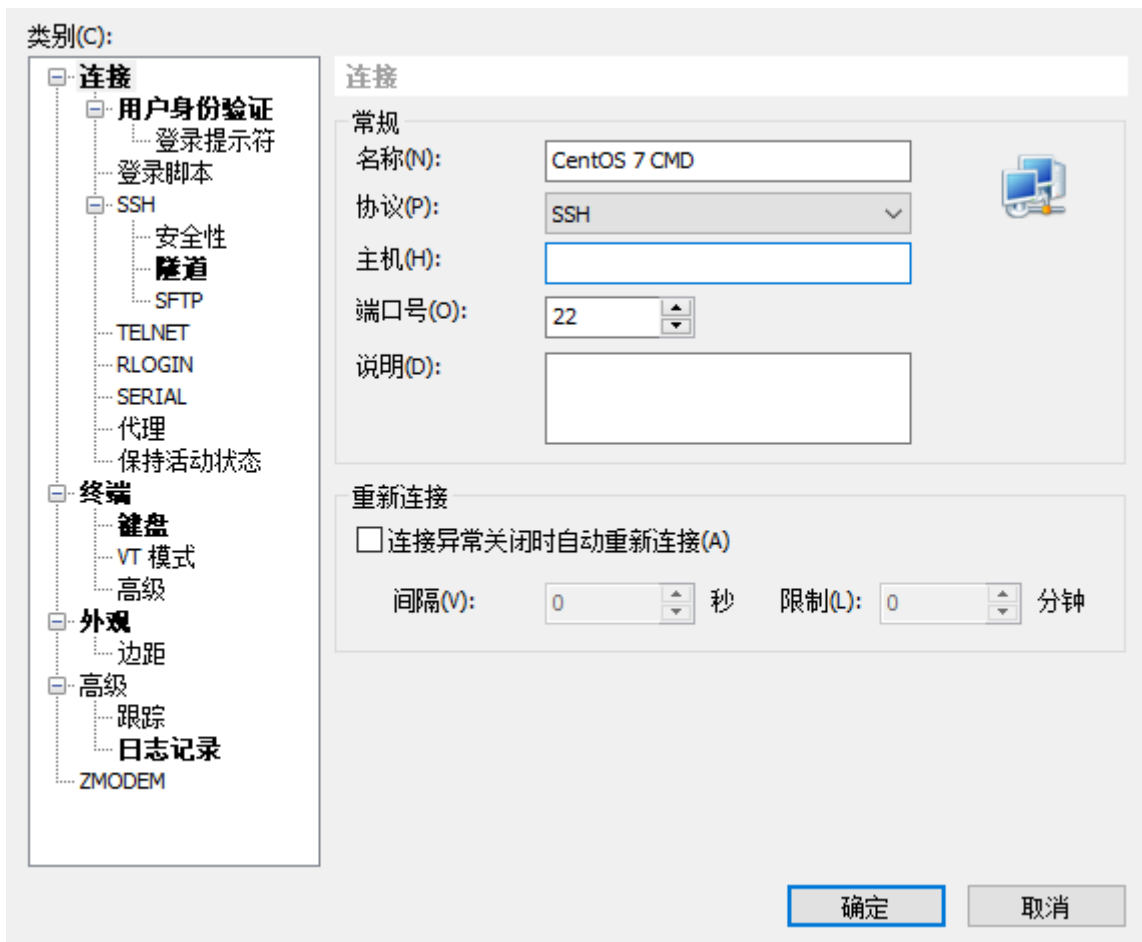
## 1、下载与安装

(略)

## 2、创建连接

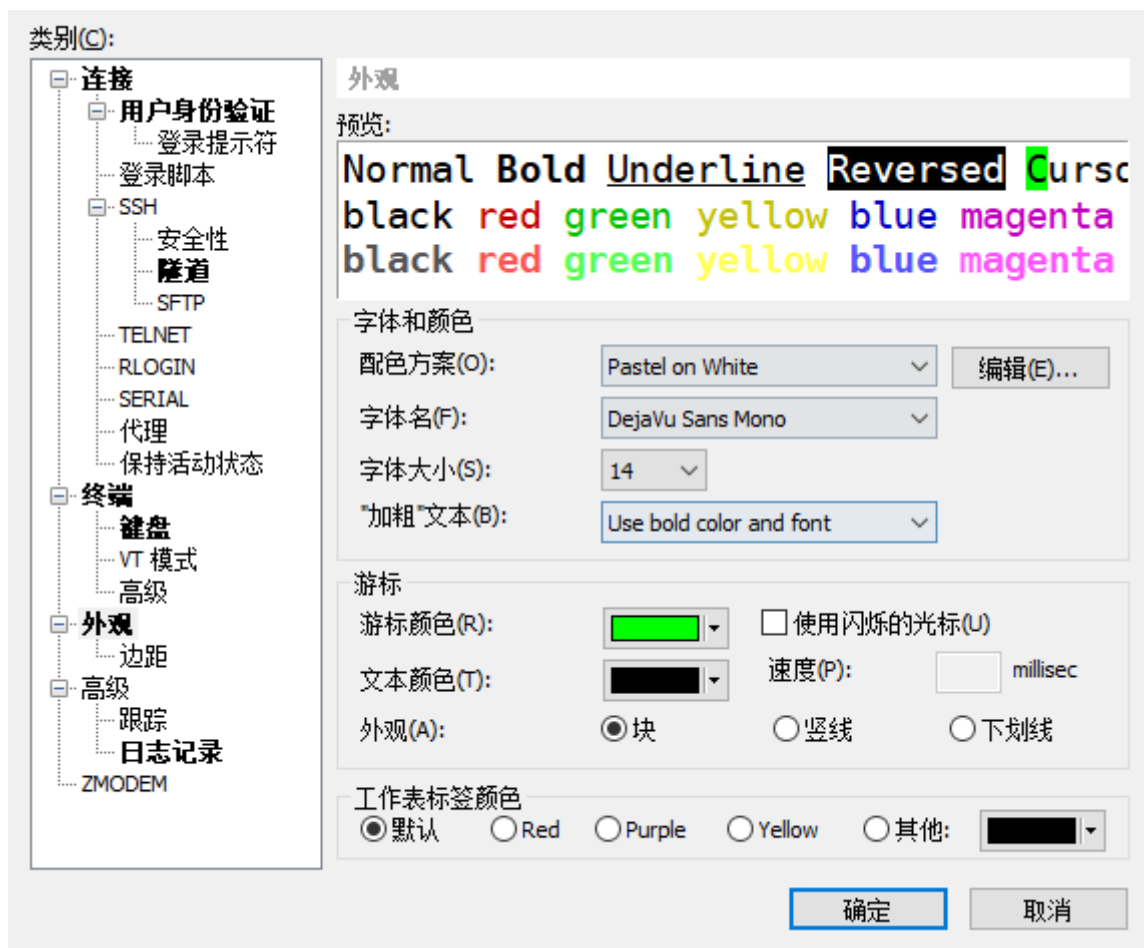


在新建会话属性窗口中，修改名称，填入主机地址（即CentOS中的地址）

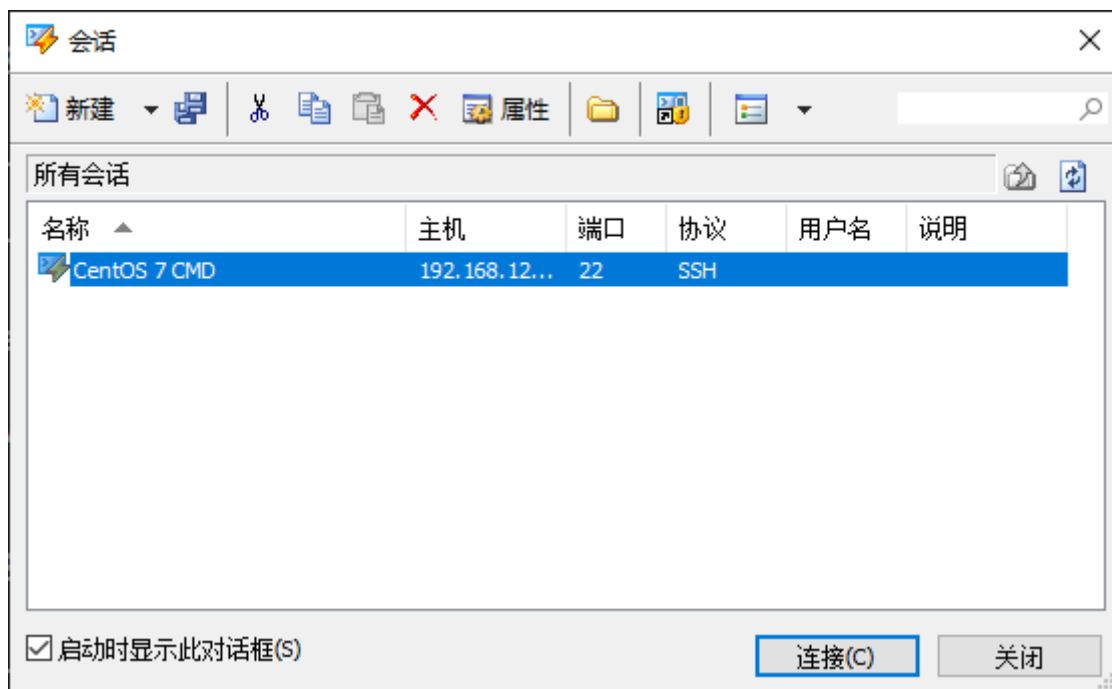


当然，此刻CentOS系统一定要先打开并且连接上网络，通过 `ip addr` 命令可查看当前系统的ip地址，然后填入窗口。

在【外观】选项中，选择个人的配色方案、字体、字体大小等。



确定后，在打开的会话窗口中点击【连接】



连接时需要输入用户名和密码，可选中记住，之后无须重复输入。



SSH 用户名

远程主机: 192.168.126.131:22 (CentOS 7 C

服务器类型: SSH2, OpenSSH\_7.4

请输入登录的用户名(E):

root

☒ 记住用户名(R)

确定 取消

SSH用户身份验证

远程主机: 192.168.126.131:22 (CentOS 7 CMD

登录名: root

服务器类型: SSH2, OpenSSH\_7.4

请在下面选择恰当的身份验证方法并提供登录所需的信息。

☒ Password(P)

密码(W):

☐ Public Key(U)

用户密钥(K): 浏览(B)...

密码(H):

☐ Keyboard Interactive(I)

使用键盘输入用户身份验证。

☒ 记住密码(R)

确定 取消

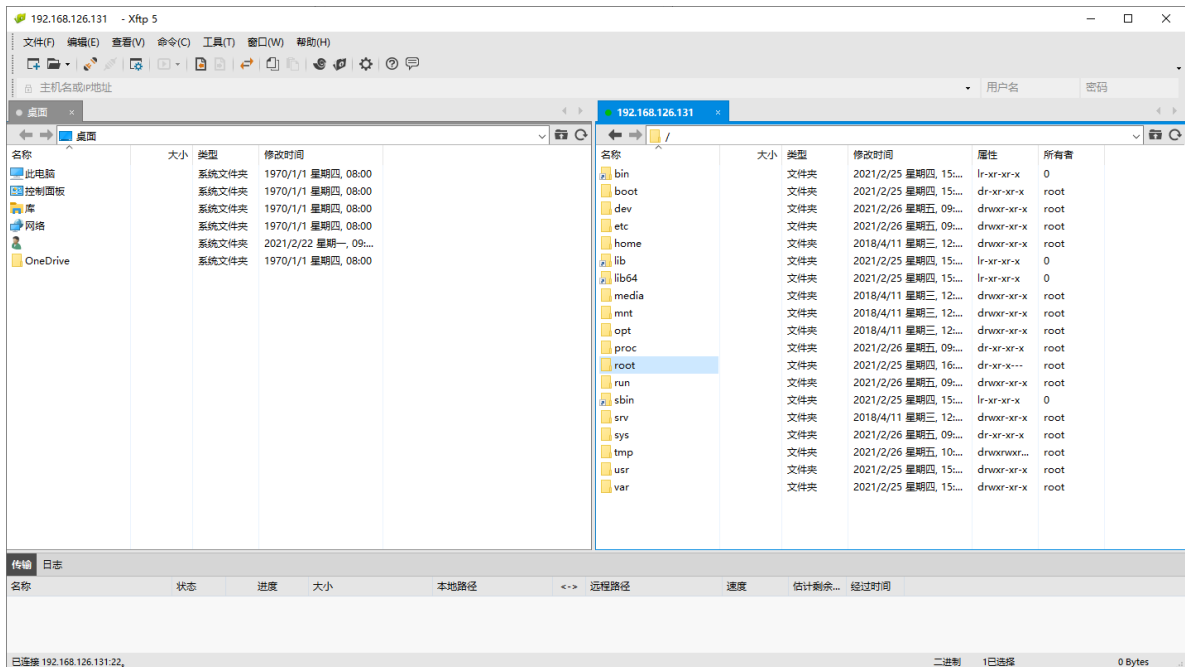
1 CentOS 7 CMD x +

[root@localhost ~]#

### 3、传输文件

当和远程系统建立了连接之后，可以通过Xshell中的Xftp工具非常方便的进行文件传输





## Linux关机操作

在图形化（GUI）界面中，关机可以通过点击系统菜单栏的关机按钮实现，在命令行模式下可以通过输入关机指令来完成关机操作（不推荐直接关闭虚拟机）。

Linux系统中和关机相关的命令有：

- **shutdown**

- shutdown会给系统计划一个时间关机。它可以被用于停止、关机、重启机器。
- 示例：

```
# shutdown ### 默认关机动作：10分钟后关机
# shutdown -p 10 ### 10分钟后关机，和上一条等效
# shutdown now ### 立刻关机
# shutdown 13:20 ### 在 13:20 关闭机器
# shutdown -p now ### 立刻关闭机器
# shutdown -h now ### 立刻停止机器
# shutdown -r 09:35 ### 在 09:35am 重启机器
```

- 要取消即将进行的关机，使用命令：**shutdown -c**

- **halt**

- 最简单的关机命令，halt通知硬件来停止所有的 CPU 功能，但是仍然保持通电，需要手动关闭电源，有些发行版增强了功能可以直接关闭电源。可以用它使系统处于低层维护状态。其实就相当于调用了 **shutdown -h**
- 示例：

```
# halt          ### 停止机器
# halt -p       ### 关闭机器
# halt --reboot  ### 重启机器
```

- **poweroff**

- poweroff会发送一个 ACPI 信号来通知系统关机。

- 示例：

```
# poweroff          ### 关闭机器
# poweroff --halt    ### 停止机器
# poweroff --reboot  ### 重启机器
```

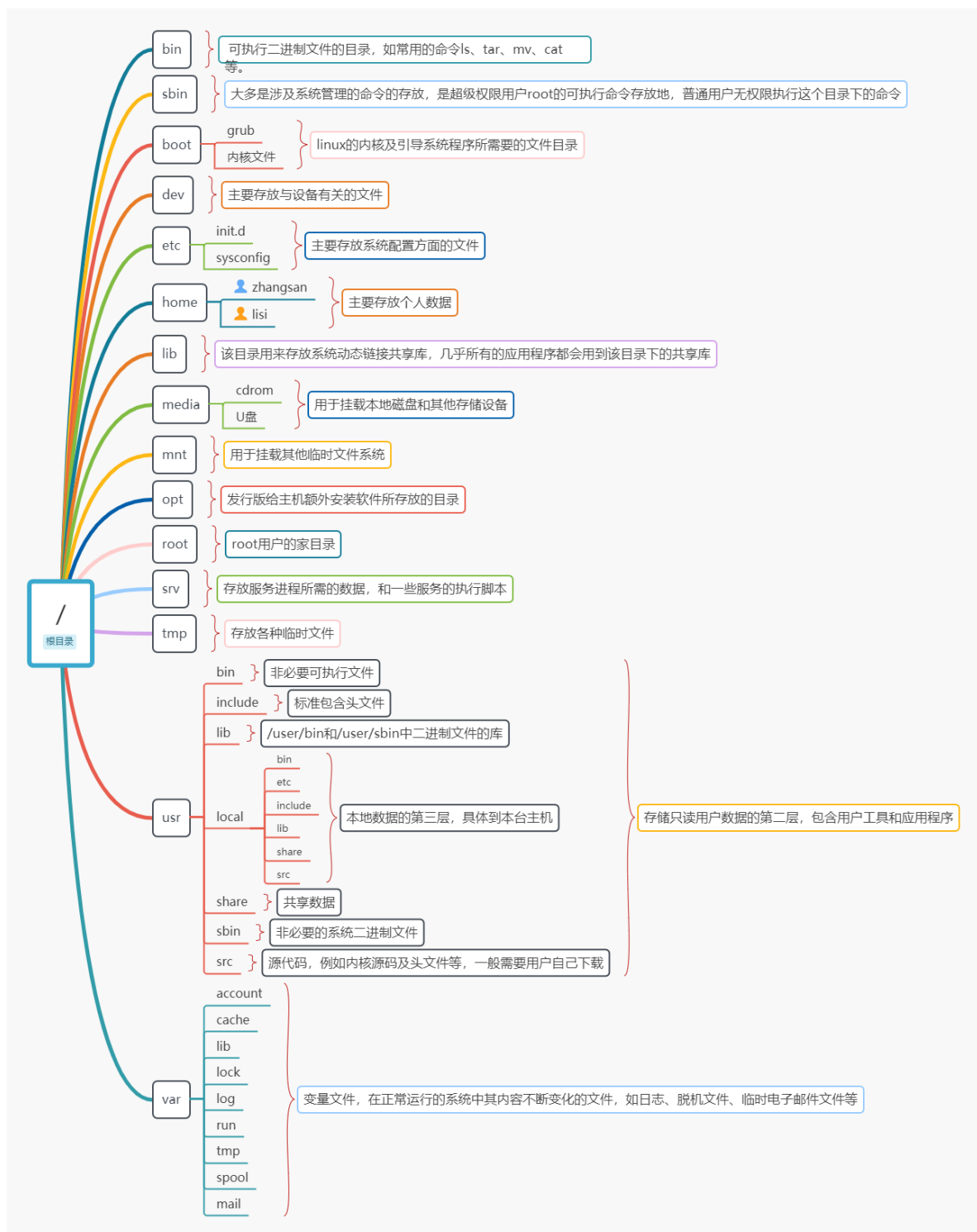
- **reboot**

- reboot 通知系统重启。
- 示例：

```
# reboot          ### 重启机器
# reboot --halt    ### 停止机器
# reboot -p        ### 关闭机器
```

## Linux目录结构

---



## 三、Linux系统基本指令

### 系统操作

- **cd**
  - 切换目录，改变当前的工作目录（change directory）
  - 基本语法：**cd 目录名**
  - 案例演示：

```
# Linux是多用户系统，允许开多个用户，每个用户都有自己的home目录，例如Windows下
C:/Users/xxx
# Linux下的家目录不显示路径，显式~
cd ### 切换到用户主目录（home目录）
cd ~ ### 同上

# Linux下没有盘符的概念，所有的文件都在/根目录下，切换目录时注意是不是在当前目录下
cd / ### 切换到根目录
cd .. ### 上一级目录
cd . ### 当前目录
cd - ### 上一个历史工作目录，相当于windows资源管理器中的后退、
cd /xx ### 绝对路径，根路径开始
cd xx ### 相对路径，当前路径下的xx目录
```

- **pwd**
  - 打印当前的工作目录（print working directory）
  - 基本语法： **pwd**
- **history**
  - 查看历史命令记录，Linux系统会记录用户输入过的指令，关机重启后还存在
  - 基本语法： **history** [选项]
  - 常用选项：
    - **-c** : 清除历史指令
    - **n** : 输出最近的n条指令
  - 案例演示：

```
history 10 ### 显示最近的10条指令记录
history -c ### 清除历史指令记录
```

- **man**
  - 查看指定指令的帮助手册 manual
  - 基本语法： **man** 指令名
- **clear**
  - 清除屏幕（其实没有清除，只是把屏幕滚动到上面去了）
  - 基本语法： **clear**

## 文件基本操作

- **ls**
  - 显式指定目录下的文件清单，ls x则表示显式x目录下的文件清单
  - 基本语法： **ls** [选项] [目录名]
  - 常用选项：
    - **-l** : long，长格式，相当于windows资源管理器中的详细信息，整个指令可以直接缩写为 **ll**
    - **-a** : all，所有，包括隐藏文件（以点开头的文件均为隐藏文件）
    - **-A** : 所有，包括隐藏文件，但不包括 **.** 和 **..** 这两个特殊目录
    - **-R** : 递归子目录中的文件或目录
  - 案例演示：

```
ls -al
ll -a
```

- **touch**

- 修改文件或者目录的时间属性，包括存取时间和更改时间。若文件不存在，系统会建立一个新的文件。**通常用来创建一个新文件**
- 基本语法: **touch 文件名**

- **mkdir**

- make directory, 创建文件夹
- 基本语法: **mkdir [选项] 目录名**
- 常用选项:
  - **-p**: 建立多级目录, 父目录不存在时会创建父目录
- 案例演示:

```
[root@localhost ~]# mkdir test
[root@localhost ~]# ll
总用量 4
-rw-----. 1 root root 1201 2月 25 15:31 anaconda-ks.cfg
drwxr-xr-x. 2 root root  6 3月  4 14:36 test
[root@localhost ~]# mkdir a/b/c
mkdir: 无法创建目录"a/b/c": 没有那个文件或目录
[root@localhost ~]# mkdir -p a/b/c
[root@localhost ~]# ll
总用量 4
drwxr-xr-x. 3 root root  15 3月  4 14:37 a
-rw-----. 1 root root 1201 2月 25 15:31 anaconda-ks.cfg
drwxr-xr-x. 2 root root  6 3月  4 14:36 test
```

- **rmdir**

- remove directory, 删除空的文件夹, 若目录中有内容则删除失败
- 基本语法: **rmdir [选项] 目录名**
- 常用选项:
  - **-p**: 删除指定目录以及其父目录
- 案例演示:

```
[root@localhost ~]# rmdir a
rmdir: 删除 "a" 失败: 目录非空
[root@localhost ~]# rmdir -p a/b/c
[root@localhost ~]# ll
总用量 4
-rw-----. 1 root root 1201 2月 25 15:31 anaconda-ks.cfg
drwxr-xr-x. 2 root root  6 3月  4 14:36 test
```

- **rm**

- remove, 删除文件夹或文件
- 基本语法: **rm [选项] 文件或文件名**
- 常用选项:
  - **-f**: force, 强制删除, 无需确认
  - **-r**: recursion, 目录及目录之下的文件递归, 逐个询问删除
- 特别注意: **rm -rf \*** ==> 强制删除当前目录下所有文件, 同理, 特别注意 **rm -rf /\***

- 案例演示:

```
rm -rf a ### 直接删除a, 如果a是目录, 则会删除a下所有文件和目录内容
```

```
[root@localhost ~]# ll
总用量 4
-rw-r--r--. 1 root root    0 3月   4 14:44 123.txt
-rw-----. 1 root root 1201 2月   25 15:31 anaconda-ks.cfg
drwxr-xr-x. 2 root root    6 3月   4 14:36 test
[root@localhost ~]# rm 123.txt
rm: 是否删除普通空文件 "123.txt"? y
[root@localhost ~]# ll
总用量 4
-rw-----. 1 root root 1201 2月   25 15:31 anaconda-ks.cfg
drwxr-xr-x. 2 root root    6 3月   4 14:36 test
```

- **cp**

- copy, 复制文件或目录
- 基本语法: `cp [选项] 源 目标`
- 常用选项:
  - `-r: cp -r a b` 将a以及a中的子文件和子文件夹都拷贝到b中, 使用cp命令复制目录时, 必须带-r参数
  - `-f`: 强制复制, 若有同名文件则覆盖且不会提示
  - `-i`: 和-f相反, 覆盖之前给出提示

- **mv**

- move, 为文件或目录改名、或将文件或目录移入其它位置 (类似于剪切-粘贴)。
- 基本语法: `mv [选项] 源文件或目录 目标目录`、`mv 原文件名 新文件名`
- 常用选项:
  - `-f`: 直接覆盖不提示
  - `-i`: 给出提示

- 通配符

- `*`: 匹配任意多个字符
  - 案例演示:

```
rm *.txt ### 删除所有后缀为.txt的文件
rm a*     ### 删除所有以a开头的文件
```

- `?`: 匹配一个字符
  - 案例演示:

```
rm ?.txt ### 删除文件名只有一个字符的txt文件
```

- **> 和 >>**

- 用于将命令执行结果输入到文件中
- 例如: `ll` / 本来是以列表形式输出根目录下的所有文件到控制台, 但是想输出到 `log.txt` 中, 再比如: `echo "Hello"`, 是在当前控制台输出一个内容, 同样的, 想把它写入到文件中。
- 基本语法: 重定向 (覆盖写): `命令 > 指定文件`、追加写: `命令 >> 指定文件`
- 案例演示:

```
echo "Hello Linux" > log.txt
echo "你好" >> log.txt
```

- **ln**

- link, 为某一个文件在另外一个位置建立一个链接
- 基本语法: **ln** [选项] 目标文件或目录 链接名
- 常用选项:
  - **-s**: symbolic, 符号链接, 也叫软链接。
- 案例演示:

```
# 创建一个软链接linktoetc, 链接到/etc:
ln -s /etc linktoetc
# 提示: 不加-s选项将会创建硬链接, 但是不能对目录创建硬链接, 会提示出错
# cd linktoetc: 将会显示进入了linktoetc, 不过实际工作在etc下
# pwd将会显示链接名,
# pwd -P则会显示软链接的真实路径
# 在软链接所在目录中使用ls -l也可以观察链接的目标

# 删除软链接:
rm linktoetc
```

## 文件内容显示

- **cat**

- concatenate, 显式文件内容在屏幕上
- 基本语法: **cat** [选项] 文件名
- 常用选项:
  - **-n**: 显示行号
- 案例演示:

```
[root@localhost ~]# cat -n anaconda-ks.cfg
 1 #version=DEVEL
 2 # System authorization information
 3 auth --enableshadow --passalgo=sha512
 4 # Use CDRom installation media
 5 cdrom
 6 # Use graphical install
```

- **more**、**less**

- 阅读形式浏览文件内容, 以一页一页的形式展示文件内容。 **more** 先将文件内容全部加载, 然后分屏显式, **less** 只加载当前需要显式的一屏内容, 所以一般浏览大型文档用 **less**
- 基本语法: **more** 文件名、**less** 文件名
- 操作模式:
  - **more**: **[回车]** 逐行滚动, **[空格]** 或者 **[ctrl] + [f]** 翻下一屏, **[ctrl] + [b]** 回上一屏, 看完自动退出文件
  - **less**: 相对于more, less浏览完后不会自动退出, 允许随意浏览, **[回车]** 显式下一行, **[空格]** 翻下一屏, 可以使用 **[pagedown]** 和 **[pageup]** 来翻页, 按 **[q]** 退出浏览



- `head`、`tail`
  - `head` 命令用于显示文件的开头的内容。在默认情况下，`head` 命令显示文件的头10行内容。
  - `tail` 命令用于显示文件的结尾的内容。在默认情况下，`tail` 命令显示文件的后10行内容。
  - 案例演示：

```
head -5 文件名 ### 显示文件的前5行
```

## 搜索文件内容

- `grep`
  - Globally search for a Regular Expression and Print the results, 根据关键字搜索并显示关键字所在的行号
  - 基本语法: `grep [选项] 关键字 文件名`
  - 常用选项:
    - `-i`: 忽略大小写
    - `-v`: 显示不匹配的行
    - `-n`: 显示行号
    - `-c`: 统计查找到匹配的总行数
  - 案例演示:

```
[root@localhost etc]# grep 514 rsyslog.conf
#$UDPServerRun 514
#$InputTCPServerRun 514
# remote host is: name/ip:port, e.g. 192.168.0.1:514, port optional
#*. * @@remote-host:514

[root@localhost etc]# grep -n 514 rsyslog.conf
16:#$UDPServerRun 514
20:#$InputTCPServerRun 514
89:# remote host is: name/ip:port, e.g. 192.168.0.1:514, port optional
90:#*. * @@remote-host:514
```

```
grep "a b" 1.txt ### 如果要查找的内容中包含空格，需要用引号括起来
```

## 搜索文件

- `find`
  - 根据指定的条件查找文件
  - 基本语法: `find [查找范围] [查找条件] 文件名 [动作]`
  - 查找条件:
    - `-name`: 根据文件名查找
    - `-type`: 根据文件类型查找，后面需跟上文件类型字母，如d代表目录，l代表链接，f代表普通文件
    - `-user`: 根据所属用户查找
    - `-size`: 根据文件大小进行查找
  - 案例演示:

```
# 在/home目录下查找xyz.txt:
find /home -name xyz.txt ### 将会递归查找子目录

# 查找系统中所有属于用户zhangsan的文件:
find / -user zhangsan

# 查找系统中大于20M的文件:
find / -size +20M ### 大于是加号+, 小于就是减号-, 不写表示等于, 单位兆用大写M, 千字节则用小写k
```

- 动作表示对查找结果进一步操作:

```
find / -size +20M -ls ### 查询出结果并显示详细信息
```

- **locate**

- **locate** 命令其实是 **find -name** 的另一种写法, 但是要比后者快得多。
- 基本语法: **locate** [选项] 文件名
- 案例演示:

```
# 搜索etc目录下所有以sh开头的文件:
locate /etc/sh

# 搜索用户主目录下, 所有以m开头的文件:
locate ~/m

# 搜索用户主目录下, 所有以m开头的文件, 并且忽略大小写:
locate -i ~/m
```

- **whereis**

- 搜索一个命令所在的路径以及它的帮助文档所在的位置, 将会搜索二进制文件 (参数 **-b**)、man说明文件 (参数 **-m**) 和源代码文件 (参数 **-s**)。如果省略参数, 则返回所有信息。
- 基本语法: **whereis** [选项] 命令名
- 案例演示:

```
whereis grep

whereis cd
```

- **which**

- 搜索一个命令所在的路径及它的别名
- 基本语法: **which** 命令名
- 案例演示

```
# 查看ll命令的别名以及命令所在的路径:
which ll
```

```
[root@localhost ~]# which ll
alias ll='ls -l --color=auto'
/usr/bin/ls
```

# 四、Linux系统管理（一）

## 用户和组管理

### 用户组概念

Linux采用“组”来组织和管理用户，在Linux中每个用户有唯一的用户标识符UID，该UID是一个无符号整数，

同时每个用户也必须至少属于一个组，也有组标识符GID，其中UID和GID独立编号。

群组可以让多个用户具有相同的权限，同时也可以这样理解，一个用户可以所属多个群组，并同时拥有这些群组的权限，所以在Linux中又有主组和附加组的概念，主组也被称为初始组。一个用户可以所属多个附加组，但只能有一个主组。

### 组管理

- **groupadd**
  - 添加一个组
  - 基本语法：**groupadd** [选项] 组名
  - 说明：创建一个组的同时会在 **/etc/group** 文件中为该组增加相应的一行，用来记录该组的名称、GID及成员等信息
  - 常用选项：
    - **-g** :指定要添加的组的GID
  - 案例演示：

```
# 创建一个新的组，其GID号为已保存的GID的下一个顺序编号
groupadd group1

# 增加一个新的组，并指定其GID
groupadd -g 888 group1
```

- **groupdel**
  - 删除组
  - 基本语法：**groupdel** 组名
  - 说明：当某个组是某个现有用户的主要组时不能被删除

### 用户管理

- **useradd**
  - 添加用户
  - 基本语法：**useradd** 用户名
  - 说明：新增用户，如果没有指定用户组，则用户自己成为一组，同时在 **/etc/passwd** 文件和 **/etc/shadow** 文件增加一行，并自动为用户创建相应的主目录 **/home/用户名**。**/etc/passwd** 文件记录了系统中每个用户的用户名、UID、GID、主目录、shell等信息。
  - 案例演示：

```
useradd -u 600 -g group1 -G group2 -d /home/student1 user1
# 功能：创建用户user1，其中参数：
# -u指定UID号为600
# -g指定用户的主要组为group1，也可以在这里写GID
# -G指定用户的附加组为group2，每个用户可以有多个附加组
# -d指定用户的主目录为/home/student1
```

```
# 查看系统中有哪些用户
cut -d : -f 1 /etc/passwd
# cut命令用来提取字符串，其中参数：
# -d : 表示使用冒号作为分隔符
# -f 1 表示提取分隔后的第1个分块
# /etc/passwd中的内容如：root:x:0:0:root:/root:/bin/bash
# 首先按冒号分隔，然后提取第1个，则得到了root

# 查看系统中有哪些可以登录系统的用户
cat /etc/passwd | grep -v /sbin/nologin | cut -d : -f 1
# 1、显示/etc/passwd的内容
# 2、在其中搜索不匹配 /sbin/nologin 的行
# 3、剩下的行按冒号分隔后提取第1块
```

- **id**

- 显示用户的UID、GID及所属的组信息
- 基本语法： **id 用户名**
- 案例演示：

```
# 显示root用户的信息
id root
```

- **su**

- switch user, 切换用户
- 基本语法： **su 用户名**
- 说明：root用户切换到其他普通用户可以任意切换，其他情况需要输入用户相应的密码。

- **groups**

- 查看用户所属的组信息
- 基本语法： **groups [用户名]**
- 案例演示：

```
# 查看当前用户的组信息
groups

# 查看用户lisi的组信息，显示结果为 用户名 : 组名
groups lisi
```

- **passwd**

- 管理指定用户的密码，root用户可以管理其他用户密码，普通用户只能管理自己的密码
- 基本语法： **passwd [选项] 用户名**
- 常用选项：
  - 无选项时：改变密码
  - **-d** : delete, 删除指定用户的密码，删除之后该用户登录时不需要密码
  - **-l** : lock, 锁定指定用户，给指定的用户加锁后，该用户不能登录，密码失效

- `-u` : unlock, 解锁指定用户, 恢复该用户登录

◦ 案例演示:

```
# 改变用户zhangsan的密码
passwd zhangsan

# 改变用户自己的密码
passwd

## 删除用户zhangsan的密码
passwd -d zhangsan
```

```
[root@localhost ~]# useradd zhangsan
```

```
[root@localhost ~]# passwd zhangsan
```

更改用户 **zhangsan** 的密码。

新的 密码:

无效的密码: 密码少于 8 个字符

重新输入新的 密码:

抱歉, 密码不匹配。

新的 密码:

无效的密码: 密码未通过字典检查 - 过于简单化/系统化

重新输入新的 密码:

```
[root@localhost zhangsan]# su zhangsan
```

```
[zhangsan@localhost ~]$ passwd
```

更改用户 **zhangsan** 的密码。

新的 密码:

• `userdel`

◦ 删除用户

◦ 基本语法: `userdel` [选项] 用户名

◦ 常用选项:

- `-r` : 删除用户的同时删除其家目录

◦ 案例演示:

```
# 删除用户张三
userdel zhangsan
```

```
[root@localhost zhangsan]# userdel zhangsan
```

```
userdel: user zhangsan is currently used by process 29739
```

若删除中出现上图提示, 说明之前通过 `su` 命令切换过用户, 需要使用 `CTRL` + `D` 快捷键或者 `exit` 命令退出当前用户, 可能会需要多退几次, 之后再删除即可。

```
[root@localhost ~]# userdel zhangsan
```

```
[root@localhost ~]# ll /home
```

总用量 0

```
drwx-----. 2 1000 1000 83 3月  5 09:01 zhangsan
```

• `usermod`

◦ 修改用户的信息

◦ 基本语法: `usermod` [选项] 用户名

◦ 常用选项:

- `-a`: 将用户添加到附加组，只能和 `-G` 一起使用
  - `-d`: 修改用户的主目录
  - `-g`: 修改用户的主组，用户主目录中，属于原来的主组的文件将转交新组所有。主目录之外的文件所属的组必须手动修改。
- 案例演示:

```
[root@localhost ~]# usermod -a -G lisi zhangsan
[root@localhost ~]# id zhangsan
uid=500(zhangsan) gid=500(zhangsan) 组=500(zhangsan),501(lisi)
[root@localhost ~]# groups zhangsan
zhangsan : zhangsan lisi

[root@localhost ~]# usermod -g lisi zhangsan
[root@localhost ~]# groups zhangsan
zhangsan : lisi
```

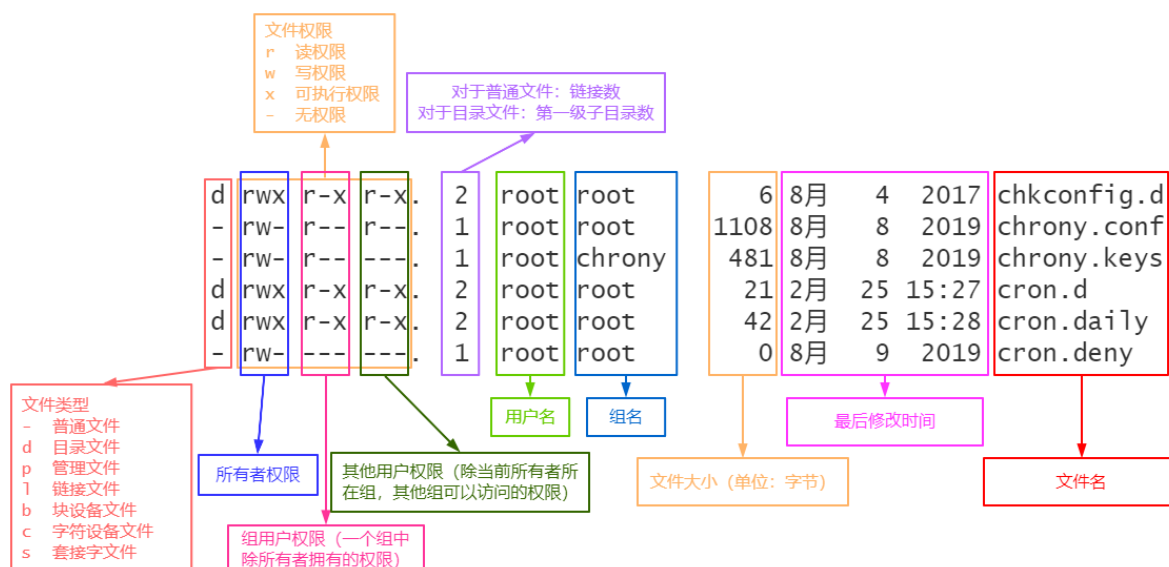
## 文件系统管理

### 文件属性信息

在基本操作命令中，通过 `ls -l` 或 `ll` 可以以列表形式显示文件的详细信息：

```
[root@localhost ~]# ll /etc
总用量 1048
-rw-r--r--. 1 root root      16 2月 25 15:30 adjtime
-rw-r--r--. 1 root root    1518 6月 7 2013 aliases
-rw-r--r--. 1 root root   12288 2月 25 15:39 aliases.db
drwxr-xr-x. 2 root root     236 2月 25 15:28 alternatives
-rw-----. 1 root root     541 8月 9 2019 anacrontab
```

这些信息中每个字段都有其含义：



### 关于权限

文件或目录的访问权限分为 **只读**，**只写** 和 **可执行** 三种。

- r: read; 对于文件, 表示可以阅读该文件的内容; 对于目录, 表示可以通过ls或ll命令查看目录列表
- w: write; 对于文件, 表示可以编辑该文件; 对于目录, 表示可以修改该目录 (在该目录中进行创建、删除子文件/目录, 以及可以修改该目录名称)
- x: execute; 对于文件, 表示可以通过./xxx来执行该文件; 对于目录, 表示可以cd命令进入该目录

以文件为例, **只读** 权限表示只允许读其内容, 而禁止对其做任何的更改操作。 **可执行** 权限表示允许将该文件作为一个程序执行。

文件被创建时, 文件所有者自动拥有对该文件的读、写和可执行权限, 以便于对文件的阅读和修改。用户也可根据需要把访问权限设置为需要的任何组合。

## 关于访问者

有三种不同类型的用户可对文件或目录进行访问: **文件所有者**、**同组用户**、**其他用户**。

所有者一般是文件的创建者。所有者可以允许同组用户有权访问文件, 还可以将文件的访问权限赋予系统中的其他用户。在这种情况下, 系统中每一位用户都能访问该用户拥有的文件或目录。

每一文件或目录的访问权限都有三组, 每组用三位表示, 分别为:

- 文件属主的读、写和执行权限;
- 与属主同组的用户的读、写和执行权限;
- 系统中其他用户的读、写和执行权限。

当用 **ls -l** 命令显示文件或目录的详细信息时, 最左边的一列为文件的访问权限

## 举例说明

对于 **-rw-r--r--** :

- 第1列 表示是否是目录或者文件, -代表是文件, d代表是目录
- 第2-4列 表示文件所属用户的权限user
- 第5-7列 表示所在的组的权限group
- 第8-10列 表示其他人的权限other
- r表示可读read, w表示可写write, x表示可执行execution (例如打开文件夹)

## 文件属性管理

- **chmod**
  - change mode, chmod命令是非常重要的, 用于改变文件或目录的访问权限。用户用它控制文件或目录的访问权限。该命令有两种用法。一种是包含字母和操作符表达式的文字设定法; 另一种是包含数字的数字设定法。
  - 基本语法: **chmod [选项] 文件名**
  - 文字设定法:
    - 语法: **chmod [who] [+ | - | =] [mode] 文件名**
    - 选项说明:
      - 操作对象who可是下述字母中的任一个或者它们的组合:
        - **u** 表示“用户 (user)”, 即文件或目录的所有者。
        - **g** 表示“同组 (group) 用户”, 即与文件属主有相同组ID的所有用户。
        - **o** 表示“其他 (others) 用户”。
        - **a** 表示“所有 (all) 用户”。它是系统默认值。
      - 操作符号可以是:

- **+** 添加某个权限。
- **-** 取消某个权限。
- **=** 赋予给定权限并取消其他所有权限（如果有的话）。
- 设置mode所表示的权限可用下述字母的任意组合：

```
* `r` 可读。
* `w` 可写。
* `x` 可执行。
* `X` 只有目标文件对某些用户是可执行的或该目标文件是目录时才追加x 属性。
* `s` 在文件执行时把进程的属主或组ID置为该文件的文件属主。方式“u+s”设置文件的用户ID位，“g+s”设置组ID位。
* `t` 保存程序的文本到交换设备上。
* `u` 与文件属主拥有一样的权限。
* `g` 与和文件属主同组的用户拥有一样的权限。
* `o` 与其他用户拥有一样的权限。
```

- 文件名：以空格分开的要改变权限的文件列表，支持通配符。
- 在一个命令行中可给出多个权限方式，其间用逗号隔开。例如：`chmod g+r, o+r example` 使同组和其他用户对文件example 有读权限。
- 案例演示：

```
# 给所有人（ugo）添加可读可写可执行的权限
chmod a+rwX 1.java

# 给所在的组和其他人移除写和执行权限
chmod go-wX 1.txt
```

#### ○ 数字表示法

- **0** 表示没有权限，**1** 表示可执行权限，**2** 表示可写权限，**4** 表示可读权限，然后将其相加。其实就是对应了rwx设置为0、1时的8进制数的值，所以数字属性的格式应当为3个从0到7的八进制数，其顺序是（u）（g）（o）。
- 例如 **755**：表示设置user权限为1+2+4，即可执行、可写、可读，组group权限为1+4，即可执行、可读，其他人others权限为1+4

#### • `chown`

- change owner，改变文件的所有者。
- 基本语法：`chown [选项] 用户名 文件名`
- 常用选项：
  - **-R**：表示递归将目录下的所有文件的属主都改为指定用户
- 案例演示：

```
# 将指定的文件hello.txt的属主变更为user1:
chown user1 hello.txt

# 将目录下的所有文件的属主改为user1:
chown -R user1 homework
```

#### • `chgrp`

- change group，改变文件的所属组
- 基本语法：`chgrp [选项] 组名 文件名`
- 常用选项：
  - **-R**：表示递归将目录下的所有文件的所属用户组都改为指定组



- 案例演示:

```
# 将指定的文件的所属用户组更改为group1:  
chgrp group1 hello.txt
```

## 文件权限操作案例

- 1、创建一个sh文件，并输入一个简单的shell脚本内容：

```
[root@localhost ~]# touch myshell.sh  
[root@localhost ~]# echo "echo 'Hello world'" >> myshell.sh  
[root@localhost ~]# cat myshell.sh  
echo 'Hello world'  
[root@localhost ~]# ll  
总用量 8  
-rw-----. 1 root root 1201 2月 25 15:31 anaconda-ks.cfg  
-rw-r--r--. 1 root root 19 3月 5 11:01 myshell.sh
```

- 2、执行该脚本：

```
[root@localhost ~]# ./myshell.sh  
-bash: ./myshell.sh: 权限不够
```

为什么权限不够？因为该文件的所有者拥有的权限有r、w，却没有x，意味着此时用户可以对它进行读（cat）、或者修改（vim），但是执行则权限不够。

那么想要执行这个脚本怎么办呢？该文件myshell.sh创建完之后，默认其所有者不具有x权限，需要授权！

- 3、文件授权

```
[root@localhost ~]# chmod u=rwx,g=rx myshell.sh  
[root@localhost ~]#  
[root@localhost ~]# ll  
总用量 8  
-rw-----. 1 root root 1201 2月 25 15:31 anaconda-ks.cfg  
-rwxr-xr--. 1 root root 19 3月 5 11:01 myshell.sh
```

- 4、再次执行：

```
[root@localhost ~]# ./myshell.sh  
Hello world
```

- 5、切换用户验证：

```
[root@localhost ~]# useradd lisi  
[root@localhost ~]# su lisi  
[lisi@localhost root]$ groups  
lisi  
[lisi@localhost root]$ ./myshell.sh  
bash: ./myshell.sh: 权限不够
```

## 五、Linux系统管理（二）

## 网络基础

- 查看ip地址
  - 在centOS 6版本中使用 `ifconfig` 命令查看ip地址，而在7版本中使用了新命令 `ip addr`，可以通过安装net-tools让旧命令生效。

```
[root@localhost ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:b0:9b:3f brd ff:ff:ff:ff:ff:ff
    inet 192.168.126.134/24 brd 192.168.126.255 scope global noprefixroute dynamic ens33
        valid_lft 1687sec preferred_lft 1687sec
    inet6 fe80::41b1:66ff:ecb:30e2/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```
  - 设置自动连接（CentOS 6中网络连接需要手动进行，所以需要配置其总动连接网络）
    - 编辑系统网络配置文件：`vi /etc/sysconfig/network-scripts/ifcfg-网络名`，设置其内容：
      - `ONBOOT="yes"`
    - 保存退出后重启网络服务（`service network restart` / `systemctl restart network`）
- IP地址设置
  - 由于默认配置中，IP地址采用DHCP协议自动分配，所以会出现Linux系统IP地址经常变化，导致使用Xshell需要修改配置连接，比较麻烦。所以可以将系统的IP地址固定下来。
  - 编辑系统网络配置文件：`vi /etc/sysconfig/network-scripts/ifcfg-网络名`：
    - 将其中网卡启动类型由dhcp改为static：`BOOTPROTO="static"`
    - 补充以下IP配置：

```
IPADDR=192.168.126.137
GATEWAY=192.168.126.2
NETMASK=255.255.255.0
DNS1=192.168.126.2
```
    - 网关通过虚拟机配置查看



- 修改完成后重启网络服务: `systemctl restart network`

## 防火墙

在CentOS 6中使用 `iptables` 服务来管理防火墙, 包括开启、关闭以及查看状态, 而在CentOS 7中使用 `firewalld` 服务进行, 如果要使用老版本的服务, 可以用 `yum install iptables-services` 安装 `iptables` 服务。

- 查看防火墙状态: `systemctl status firewalld.service`
- 关闭防火墙: `systemctl stop firewalld.service`
- 启动防火墙: `systemctl start firewalld`

## 网卡设置

在CentOS 6中使用 `ifconfig` 进行网络操作, 在CentOS7中使用其他的命令:

- 开启网卡: `ifup ens33` -- ens33代表网卡名称
- 关闭网卡: `ifdown ens33`

或者使用系统命令开启或关闭网络服务

- 开启网络: `systemctl start network`
- 关闭网络: `systemctl stop network`
- 重启网络: `systemctl restart network`

## 主机名设置

编辑系统文件 `vi /etc/sysconfig/network` 输入主机名:

```
HOSTNAME=xxxx
```

## hosts设置

`vi /etc/hosts`

## 进程管理

---

- `ps`
  - process, 查询进程
  - 基本语法: `ps [选项]`
  - 常用选项:
    - 无: 查询在当前控制台上运行的进程
    - `-aux`: 查询系统中所有运行的进程, 包括后台进程, 其中参数 `a` 是所有进程, 参数 `x` 包括不占用控制台的进程, 参数 `u` 显示用户
    - `-ef`: 查询系统中所有运行的进程, 包括后台进程, 并显示出每个进程的父进程号
- `pstree`
  - process tree, 树状展示, 选择最小化安装时可能会没有安装这个命令, 需要安装: `yum install psmisc`
  - 基本语法: `pstree [选项]`
  - 常用选项:
    - 无: 树状格式显示进程列表
    - `-p`: 带进程号的树状格式显示进程列表
- `top`
  - 动态显示系统中的进程
- `kill`
  - 杀掉指定进程
  - 基本语法: `kill [选项] xx`
  - 常用选项:
    - 无: 杀掉指定进程
    - `-9`: 强制杀掉指定进程

## 服务管理

---

### 什么是服务

在Linux下, 服务(service) 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**。

### 服务管理指令

- `systemctl`
  - 在CentOS 7以上的版本, 系统服务相关的指令更换为 `systemctl`! 可以使用: `yum install initscripts` 安装service命令
  - 查看当前运行级别: `systemctl get-default`

- 运行级别 0: 系统停机状态, 系统默认运行级别不能设为 0, 否则不能正常启动
- 运行级别 1: 单用户工作状态, root 权限, 用于系统维护, 禁止远程登陆
- 运行级别 2: 多用户状态(没有 NFS, 即没有网络), 不支持网络
- 运行级别 3: 完全的多用户状态(有 NFS, 即有网络), 登陆后进入控制台命令行模式
- 运行级别 4: 系统未使用, 保留
- 运行级别 5: X11 控制台, 登陆后进入图形 GUI 模式
- 运行级别 6: 系统正常关闭并重启, 默认运行级别不能设为 6, 否则不能正常启动
- 服务操作: `systemctl status/start/stop/restart 服务名`
  - 关于操作:
    - `status`: 查看服务运行状态
    - `start`: 启动服务
    - `stop`: 停止服务
    - `restart`: 重启服务
    - `reload`: 如果这个服务有配置文件, 则会重新加载配置文件
- 列出所有服务: `systemctl list-unit-files`
- 查看服务是否为开机自启: `systemctl is-enabled 服务名`
- 设置服务开机自启: `systemctl enable 服务名`
- 设置服务开机不自启: `systemctl disable 服务名`
- 案例演示:

```
# 查看系统所有服务
systemctl list-unit-files

# 在列举出的所有系统服务中, 筛选出sshd服务的情况
systemctl list-unit-files | grep sshd
```

- `service *`
  - Linux中与服务有关的命令, 通过该命令可以查看服务的状态、停止服务、启动服务、重启服务等
  - 基本语法: `service 服务名 status/start/stop/restart/reload`
  - 案例演示:

```
[root@localhost ~]# service sshd status
Redirecting to /bin/systemctl status sshd.service
● sshd.service - OpenSSH server daemon
Loaded: loaded (/usr/lib/systemd/system/ssh.service);
```
- `chkconfig *`
  - Linux `chkconfig` 命令用于检查, 设置系统的各种服务。这是Red Hat公司遵循GPL规则所开发的程序, 它可查询操作系统在每一个执行等级中会执行哪些系统服务, 其中包括各类常驻服务。
  - 基本语法: `chkconfig [--add][--del][--list][系统服务] 或 chkconfig [--level <等级代号>][系统服务][on/off/reset]`
  - 常用选项:
    - `--list`: 列出chkconfig所知道的所有的服务的情况
    - `--add`: 增加所指定的系统服务, 让chkconfig指令得以管理它, 并同时在系统启动的叙述文件内增加相关数据。
    - `--del`: 删除所指定的系统服务, 不再由chkconfig指令管理, 并同时在系统启动的叙述文件内删除相关数据。

- `--level<等级代号> [on/off/reset]` : 指定该系统服务要在哪一个执行等级中开启或关闭。注意：服务器重启后才会生效。

## 任务调度

### 什么是任务调度

指定某个任务在特定的时间执行，或者在特定的周期内反复执行

### 如何配置任务调度

所有需要调度的任务都配置在系统文件 `/etc/crontab` 中。

crontab有2种编辑方式：直接编辑 `/etc/crontab` 文件或者使用命令 `crontab -e`，其中 `/etc/crontab` 里的计划任务是系统中的计划任务，而用户的计划任务需要通过 `crontab -e` 来编辑，会自动在 `/var/spool/cron` 下生成一个与此用户同名的文件。

### 案例演示

通过命令 `crontab -e` 进入任务编辑模式，添加以下任务：

```
*/1 * * * * date >> /home/mydate.log
```

其中：

- 任务内容：`date >> /home/mydate.log`
- 任务执行规则：`*/1 * * * *`

任务调度的规则，总共有5个\*，分别代表的意思如下：

- 第1个\* 表示一个小时中的第几分，取值范围（0-59）
- 第2个\* 表示一天当中的第几个小时，取值范围（0-23）
- 第3个\* 表示一个月当中的第几日，取值范围（1-31）
- 第4个\* 表示一年当中的第几个月，取值范围（1-12）
- 第5个\* 表示一周当中的星期几，取值范围（0-7），其中0和7都表示星期天

特殊符号的含义：

- \* 代表任何时间。比如第一个“\*”就代表一小时中每分钟都执行一次的意思。
- , 代表不连续的时间。比如“0 8,12,16 \*\*\* 命令”，就代表在每天的8点0分，12点0分，16点0分都执行一次命令
- - 代表连续的时间范围。比如“0 5 \*\* 1-6 命令”，代表在周一到周六的凌晨5点0分执行命令
- \*/n 代表每隔多久执行一次。比如“\*/10 \*\*\* 命令”，代表每隔10分钟就执行一次命令。

### 其他相关命令

- `crontab -l` : 查看当前所有已经配置的任务
- `crontab -r` : 移除所有已经配置的任务

## 六、vi和vim的使用

# 是什么

---

**vi (visual editor)** 编辑器是所有Unix及Linux系统下标准的编辑器，他就相当于Windows系统中的记事本一样，它的强大不逊色于任何最新的文本编辑器。它是我们使用Linux系统不能缺少的工具。

由于对Unix及Linux系统的任何版本，vi编辑器是完全相同的，学会它后，就可以在Linux的世界里畅行无阻。

**vim (visual editor improved)** 具有程序编辑的能力，可以以字体颜色辨别语法的正确性，方便程序设计；因为程序简单，编辑速度相当快速。

vim可以当作vi的升级版本，可以用多种颜色的方式来显示一些特殊的信息。并且vim会依据文件扩展名或者是文件内的开头信息，判断该文件的内容而自动的执行该程序的语法判断式，再以颜色来显示程序代码与一般信息。

另外，vim里面加入了很多额外的功能，例如支持正则表达式的搜索、多文件编辑、块复制等等。

## 安装vim

---

不过默认最小化安装的系统中可能是没有vim的，查看本机已经存在的包：

```
rpm -qa | grep vim
```

可能没有或者只有个别的输出，不够完整，输入vim指令也会提示不存在：

```
[root@localhost ~]# rpm -qa | grep vim
vim-minimal-7.4.629-6.el7.x86_64
[root@localhost ~]#
[root@localhost ~]# vim
-bash: vim: 未找到命令
```

通过命令进行安装：`yum -y install vim*`

更新完毕：

```
vim-minimal.x86_64 2:7.4.629-8.el7_9
```

完毕！

```
[root@localhost ~]# rpm -qa | grep vim
vim-minimal-7.4.629-8.el7_9.x86_64
vim-filesystem-7.4.629-8.el7_9.x86_64
vim-enhanced-7.4.629-8.el7_9.x86_64
vim-common-7.4.629-8.el7_9.x86_64
vim-X11-7.4.629-8.el7_9.x86_64
```

## 模式

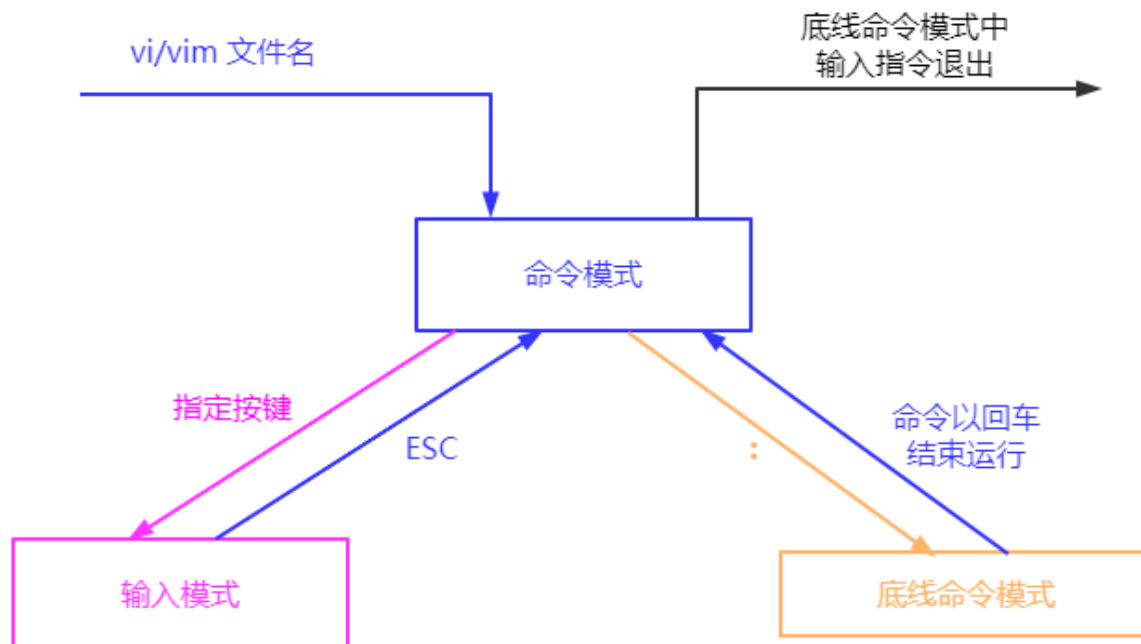
---

**vi/vim** 指令在编辑指定文件时，如果文件不存在，将会直接创建并打开。

使用 `vi/vim` 打开一个文件时，处于普通模式，也叫 **命令模式**，此时不能编辑内容，但是可以通过按键（即输入命令）进入 **输入模式/编辑模式**，或者通过按键进行复制、剪切、粘贴、撤销的操作，所以才会被称为命令模式。

另外，在这种模式下可以通过输入 **冒号** 来进入特殊的命令模式，也被称为 **底线命令模式**，因为光标会跳转到底部进行特殊命令的输入。

在编辑模式下可以按ESC回到普通模式，是不可以直接从输入模式进入底线命令模式的。



## 常用命令

### 模式转换

- 命令模式 --> 输入模式: `i/I`、`o/O`、`a/A`、`r/R`、`s/S`、`c/C`
  - `i`: 在光标处插入【最常用】
  - `I`: 在光标所在行第一个非空格字符处插入，也就是行首插入【常用】
  - `o`: 在光标所在行的下一行插入新的一行【常用】
  - `O`: 在光标所在行的上一行插入新的一行
  - `a`: 在光标所在位置的下一个字符处开始插入
  - `A`: 在光标所在行的最后一个位置开始插入，也就是行尾【比较常用】
  - `r`: 替换光标所在位置的字符，只替换一次
  - `R`: 一直替换光标所在位置的字符，也就是覆盖，直到按esc为止
  - `s`: 删除光标所在的字符并进行插入
  - `S`: 删除光标所在行的内容并在行首进行插入
  - `c`: 修改命令，和其他命令组合和执行
  - `C`: 删除光标所在位置到行尾的内容并进行插入
- 输入模式-->命令模式: `ESC`
- 命令模式-->底线命令模式: `:`
- 底线命令模式-->命令模式: `ESC, ESC`

### 保存和退出



- `:w` 保存文本
- `:q` 退出，如果文件发生修改将会提示失败
- `:q!` 不保存并强制退出【常用】
- `:wq` 保存文本并退出【常用】
- `:x` 保存文本并退出，等效于 `wq`
- 在正常模式下按组合键 `shift+zz` 可以保存并退出

## 光标移动（命令模式下）

- 按字符个数移动
  - `h`: 左;
  - `l`: 右;
  - `j`: 下;
  - `k`: 上;
  - `3h`: 左移动3个字符，同理 `6l` 表示向右移动6个字符
- 按单词移动
  - `w`: 移至下一个单词的词首;
  - `e`: 跳至当前或下一个单词的词尾;
  - `b`: 跳至当前或前一个单词的词首;
  - `2w`: 移动2个单词
- 行内跳转
  - `0`: 行首;
  - `^`: 行首的第一个非空白字符;
  - `$`: 行尾
- 行间跳转
  - 行号显示/隐藏: 底线命令模式下输入 `set number`、
  - `#G`: 跳转至第#行;
  - `gg`: 第一行;
  - `G`: 最后一行
  - 底线命令模式下，通过输入指令跳转到特定行:
    - `.`: 表示当前行;
    - `$`: 最后一行;
    - `#`: 第#行;
    - `+#`: 向下的#行

## 翻页（命令模式）

- `Ctrl+f`: 向下翻一屏;
- `Ctrl+b`: 向上翻一屏;
- `Ctrl+d`: 向下翻半屏;
- `Ctrl+u`: 向上翻半屏

## 文本编辑（命令模式）

- `yy`: 复制当前行
- `#yy`: 复制当前行开始往下的#行
- `p`: 将已复制的数据在光标下一行粘贴
- `P`: 将已复制的数据在光标上一行粘贴

- `dd` : 剪切（删除）光标所在行
- `#dd` : 剪切（删除）光标所在的向下#行，例如 5dd
- `x` : 删除光标所在位置的单个字符
- `#x` : 删除光标所在位置及向后的#个字符
- `d$` 或 `D` : 从光标所在位置处删除至行尾
- `d^` : 从光标所在位置处删除至行首
- `u` : 撤销前一次的编辑操作
- `#u` : 撤消最近#次编辑操作
- `CTRL+r` : 重做

## 其他配置

- `set ts=4` : 制表符tab的缩进默认是8个空格，修改为4个
- `set expandtb` : 制表符转换为空格存储
- `set autoindent` : 换行后保持缩进
- `syntax on` / `syntax off` : 启用/关闭语法高亮
- `vs xxx.x` / `sp xxx.x` : 将另一个文件在当前窗口中垂直分屏/水平分屏显示
  - 使用 `CTRL` + `w` 后再按 `上下左右` 按键可切换目标窗口

这些配置命令如果是在vi或vim的普通模式下输入，只会影响当前编辑环境，如果要长久生效，可以写入到 `vi ~/.vimrc` 中

# 七、软件安装

## rpm (Redhat Package Manager)

rpm文件类似于Windows中的exe安装包，各种文件已经编译好并打包，内部哪些内容安装到哪个文件夹都指定好了，安装很方便，在图形化界面中可以双击自动安装。

rpm是红帽系列的软件安装命令，如RedHat、CentOS，而Ubuntu、Suse等则没有这个指令。但是其原始设计理念是开放式的，现在包括OpenLinux、S.u.S.E.以及Turbo Linux等Linux的分发版本都有采用，可以算是公认的行业标准了。Ubuntu也可以通过安装rpm支持来使用这个命令。

## 安装软件

```
rpm -ivh xxxx.rpm
```

- `i` : 表示安装
- `v` : 表示显示安装过程
- `h` : 表示显示安装的细节

## 查询所有已安装的软件包的包名

```
rpm -qa
```

- `q` : 表示查询
- `a` : 表示所有

```
[root@localhost ~]# rpm -qa
libestr-0.1.9-2.el7.x86_64
setup-2.8.71-10.el7.noarch
tuned-2.11.0-5.el7.noarch
kbd-misc-1.15.5-15.el7.noarch
lvm2-2.02.185-2.el7.x86_64
nss-softokn-freebl-3.44.0-5.el7.x86_64
openssh-server-7.4p1-21.el7.x86_64
```

如果想要查询指定的软件包，可以在查询结果上使用 `grep` 命令筛选：

```
rpm -qa | grep xxxx
```

它的作用是列出所有包含xxx字段的软件包包名：

```
[root@localhost ~]# rpm -qa | grep python
libselinux-python-2.5-14.1.el7.x86_64
python-linux-procfs-0.4.11-4.el7.noarch
python-pycurl-7.19.0-19.el7.x86_64
python-slip-dbus-0.4.0-4.el7.noarch
python-2.7.5-86.el7.x86_64
python-decorator-3.4.0-3.el7.noarch
python-gobject-base-3.22.0-1.el7_4.1.x86_64
```

## 查询软件包的安装位置

```
rpm -ql xxx
```

- `q` : 表示查询
- `l` : 表示位置
- `xxx` : 为要查询的软件包

```
[root@localhost ~]# rpm -ql python-pyudev-0.15-9.el7.noarch
/usr/lib/python2.7/site-packages/pyudev
/usr/lib/python2.7/site-packages/pyudev-0.15-py2.7.egg-info
/usr/lib/python2.7/site-packages/pyudev-0.15-py2.7.egg-info/PKG-INFO
/usr/lib/python2.7/site-packages/pyudev-0.15-py2.7.egg-info/SOURCES.txt
/usr/lib/python2.7/site-packages/pyudev-0.15-py2.7.egg-info/dependency_links.txt
/usr/lib/python2.7/site-packages/pyudev-0.15-py2.7.egg-info/top_level.txt
```

## 查询已安装的软件的信息

```
rpm -qi xxx
```

- `q` : 表示查询
- `i` : 表示信息

```
[root@localhost ~]# rpm -qi python-pyudev-0.15-9.el7.noarch
Name       : python-pyudev
Version    : 0.15
Release    : 9.el7
Architecture: noarch
Install Date: 2021年02月25日 星期四 15时28分02秒
Group      : Development/Languages
```

## 卸载软件

```
rpm -e xxx
```

- `e` : 表示卸载

## tar ( Tape Archive )

---

tar并不是做软件安装的指令，而是用来压缩和解压的。

参数有很多，通常解压能用到的有

- `c` : 压缩文件
- `x` : 解压文件
- `z` : 格式为gzip
- `v` : 显式执行文件列表
- `f` : 要操作的文件

## 直接解压

```
tar -zxvf xxx.tar.gz
```

注意：`zxv` 三个顺序可以调整，但 `f` 必须是最后一个

## yum ( Yellow dog Updater, Modified )

---

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，可以自动处理依赖性关系，并且一次安装所有依赖的软件包，是一个在Fedora和RedHat以及SUSE中的Shell前端软件包管理器。

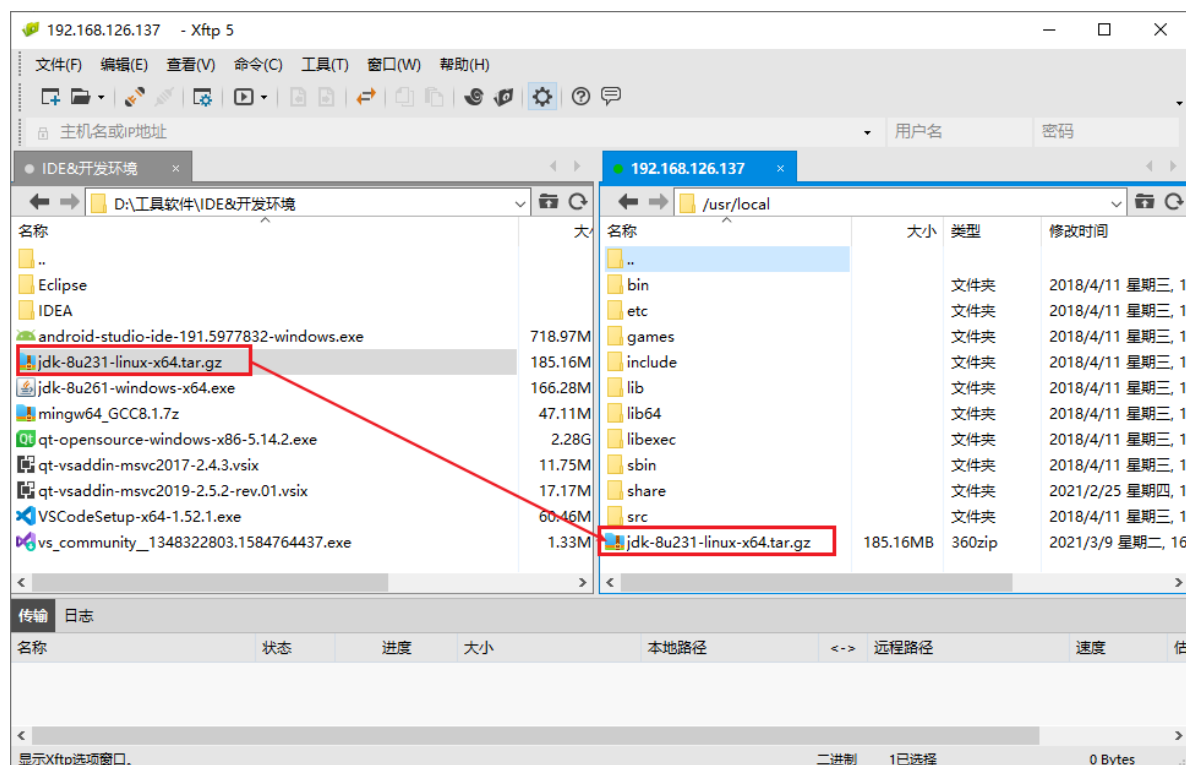
基本语法：`yum [options] [command] [package ...]`

- `[options]` 是可选的，包括
  - `-h` (帮助)
  - `-y` (当安装过程提示选择全部为“yes”)，
  - `-q` (不显示安装的过程) 等等。
- `[command]` 为所要进行的操作，包括
  - `install` 安装
  - `update` 更新，`yum update` 表示全部更新，`yum update xxx` 表示更新指定的包
- `[package ...]` 是操作的对象

## 安装JDK

---

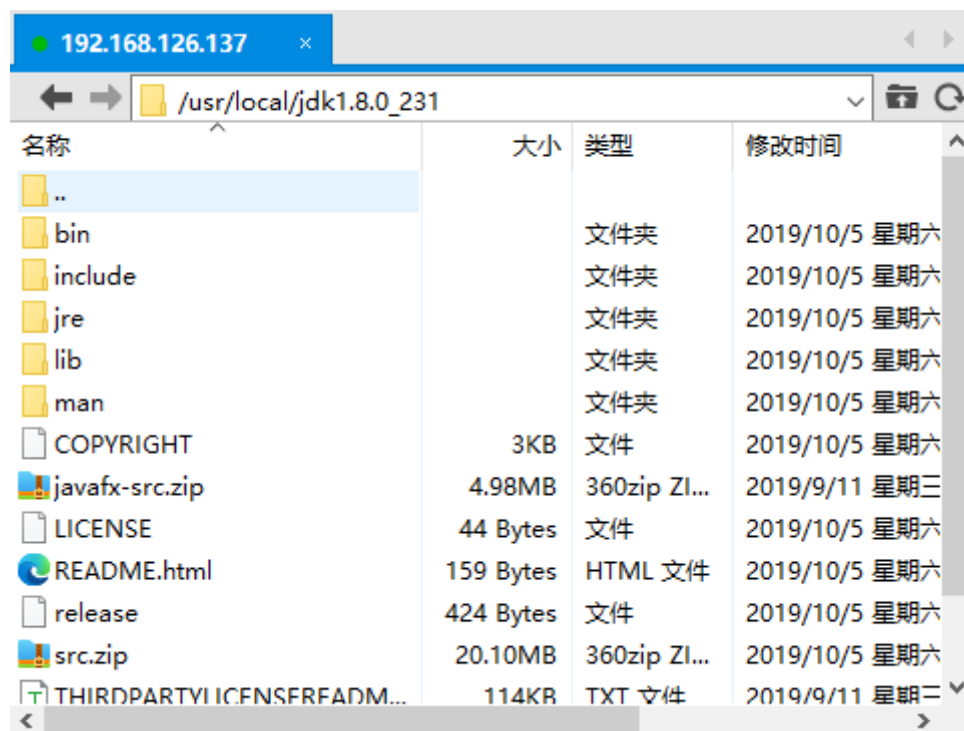
## 1、拷贝jdk的gz包到Linux中，使用XFtp



## 2、解压缩

```
[root@localhost ~]# cd /usr/local/
[root@localhost local]# ls
bin  etc  games  include  jdk-8u231-linux-x64.tar.gz  lib  lib64  libexec  sbin  share  src
[root@localhost local]# tar -xzf jdk-8u231-linux-x64.tar.gz
[root@localhost local]# ls
bin  games  jdk1.8.0_231  lib  libexec  share
etc  include  jdk-8u231-linux-x64.tar.gz  lib64  sbin  src
[root@localhost local]#
```

在Xftp中刷新可以看到解压后的文件夹



## 3、配置环境变量

编辑系统配置文件： `vi /etc/profile` ，在最后输入以下内容：

```
export JAVA_HOME=/usr/local/jdk1.8.0_231
export PATH=$PATH:$JAVA_HOME/bin

:x
```

## 4、激活配置

使用命令 `source /etc/profile` 来激活配置文件，之后输入 `java` 命令测试环境变量的配置是否生效。

```
[root@localhost local]# source /etc/profile
[root@localhost local]#
[root@localhost local]# java
用法: java [-options] class [args...]
        (执行类)
    或   java [-options] -jar jarfile [args...]
        (执行 jar 文件)
```

其中选项包括：

|                      |                   |
|----------------------|-------------------|
| <code>-d32</code>    | 使用 32 位数据模型（如果可用） |
| <code>-d64</code>    | 使用 64 位数据模型（如果可用） |
| <code>-server</code> | 选择 "server" VM    |

## 5、其他说明

编辑 `/etc/profile` 的方式是首选，当本机仅作为开发使用时推荐使用这种方法，因为此种配置时所有用户的 `shell` 都有权使用这些环境变量，可能会给系统带来安全性问题。

除此之外，还有其他方式：

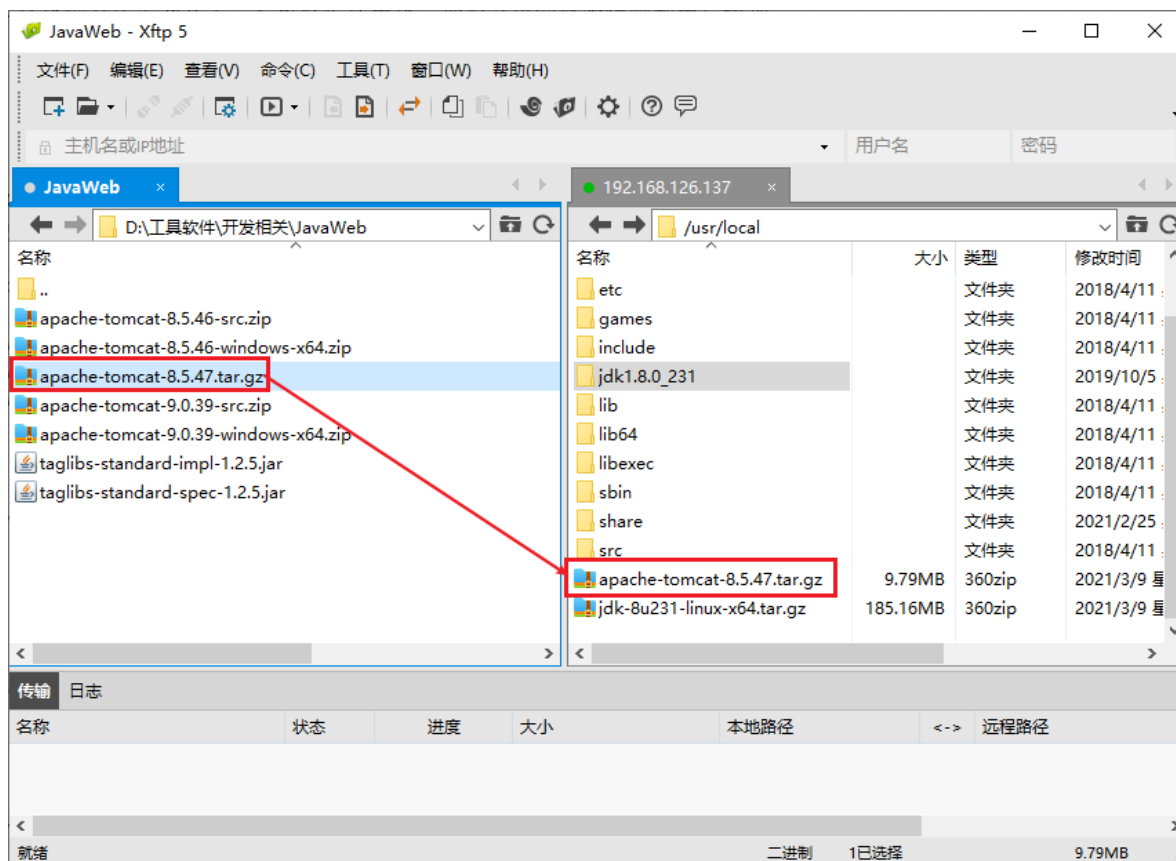
- 修改 `.bashrc` 文件这种方法更为安全，它可以把使用这些环境变量的权限控制到用户级别，如果需要给某个用户权限使用这些环境变量，只需要修改其个人用户主目录下的 `.bashrc` 文件就可以了。
- 直接在 `shell` 下设置变量。不推荐使用这种方法，因为换个 `shell`，该设置就无效了。这种方法仅仅是临时使用，以后要使用的时候又要重新设置，比较麻烦。

# 安装Tomcat

---

前提：安装Tomcat前必须安装好jdk，配置好 `JAVA_HOME`

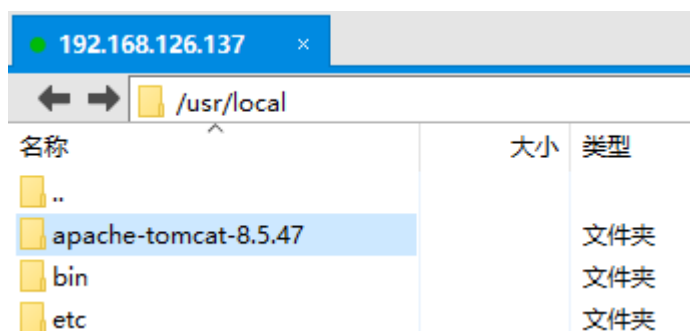
## 1、发送文件



## 2、解压缩

```
[root@localhost ~]# cd /usr/local
[root@localhost local]# ls
apache-tomcat-8.5.47.tar.gz  games          jdk-8u231-linux-x64.tar.gz  libexec  src
bin                          include        lib                          sbin
etc                          jdk1.8.0_231  lib64                       share
```

```
[root@localhost local]# tar -zxvf apache-tomcat-8.5.47.tar.gz
apache-tomcat-8.5.47/conf/
apache-tomcat-8.5.47/conf/catalina.policy
```



## 3、启动Tomcat

在tomcat文件夹下有一个 `bin` 目录，其中有一个 `startup.sh`，这是Linux下的脚本文件

| 192.168.126.137                     |       |              |       |            |      |  |
|-------------------------------------|-------|--------------|-------|------------|------|--|
| /usr/local/apache-tomcat-8.5.47/bin |       |              |       |            |      |  |
| 名称                                  | 大小    | 类型           | 修...  | 属性         | 所有者  |  |
| shutdown.sh                         | 2KB   | Shell Script | 20... | -rwxr-x--- | root |  |
| startup.bat                         | 2KB   | Windows ...  | 20... | -rw-r----- | root |  |
| startup.sh                          | 2KB   | Shell Script | 20... | -rwxr-x--- | root |  |
| tomcat-juli.jar                     | 49KB  | Executabl... | 20... | -rw-r----- | root |  |
| tomcat-native.tar.gz                | 410KB | 360zip       | 20... | -rw-r----- | root |  |
| tool-wrapper.bat                    | 4KB   | Windows ...  | 20... | -rw-r----- | root |  |

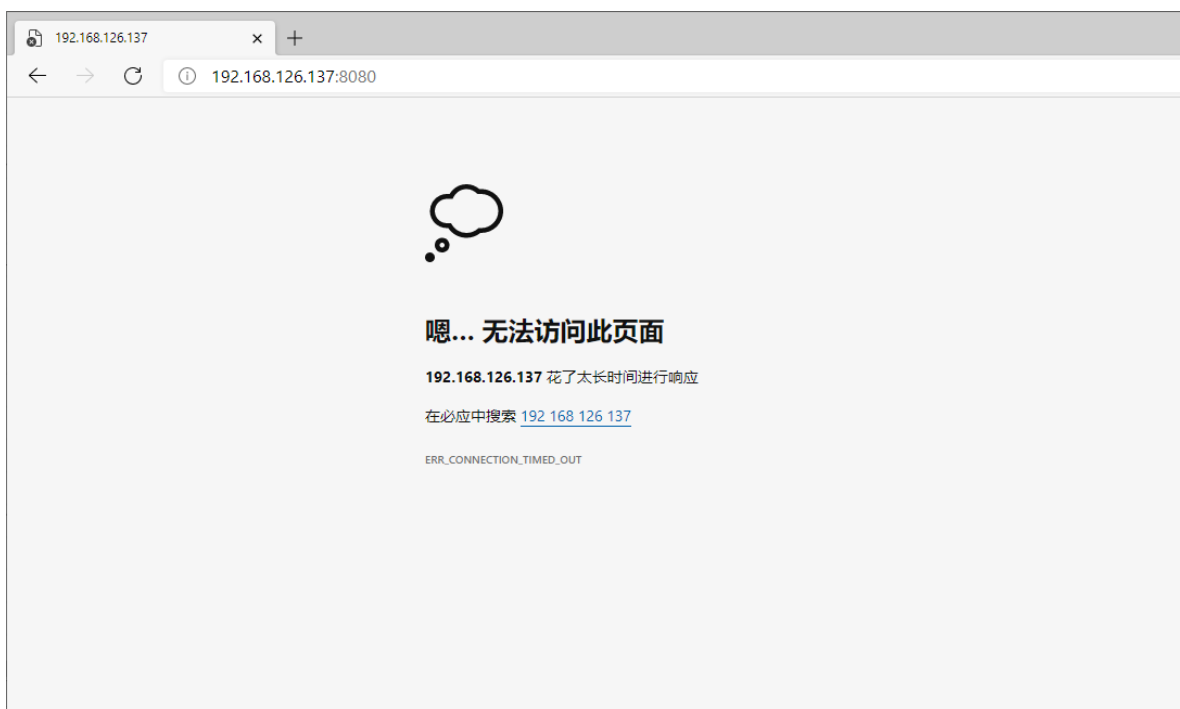
通过命令启动Tomcat:

```
[root@localhost local]# ls
apache-tomcat-8.5.47      bin  games  jdk1.8.0_231  lib  libexec  share
apache-tomcat-8.5.47.tar.gz  etc  include  jdk-8u231-linux-x64.tar.gz  lib64  sbin  src
[root@localhost local]# cd apache-tomcat-8.5.47
[root@localhost apache-tomcat-8.5.47]# cd bin/
[root@localhost bin]# ./startup.sh
Using CATALINA_BASE:   /usr/local/apache-tomcat-8.5.47
Using CATALINA_HOME:   /usr/local/apache-tomcat-8.5.47
Using CATALINA_TMPDIR: /usr/local/apache-tomcat-8.5.47/temp
Using JRE_HOME:        /usr/local/jdk1.8.0_231
Using CLASSPATH:       /usr/local/apache-tomcat-8.5.47/bin/bootstrap.jar:/usr/local/apache-tomcat-8.5.47/bin/tomcat-juli.jar
Tomcat started.
[root@localhost bin]#
```

可以到tomcat目录下找 `logs/catalina.out` 来查看输出日志确定是否启动了

## 4、访问Tomcat主页

在本机浏览器中输入 `http://Linux系统地址:8080` 进行访问:



原因在于 **防火墙**，Linux默认只开放了22端口（远程连接），所以其他端口的连接被阻拦了。

解决方法有二：关闭防火墙，或者开放8080端口：

- 关闭防火墙： `systemctl stop firewalld`

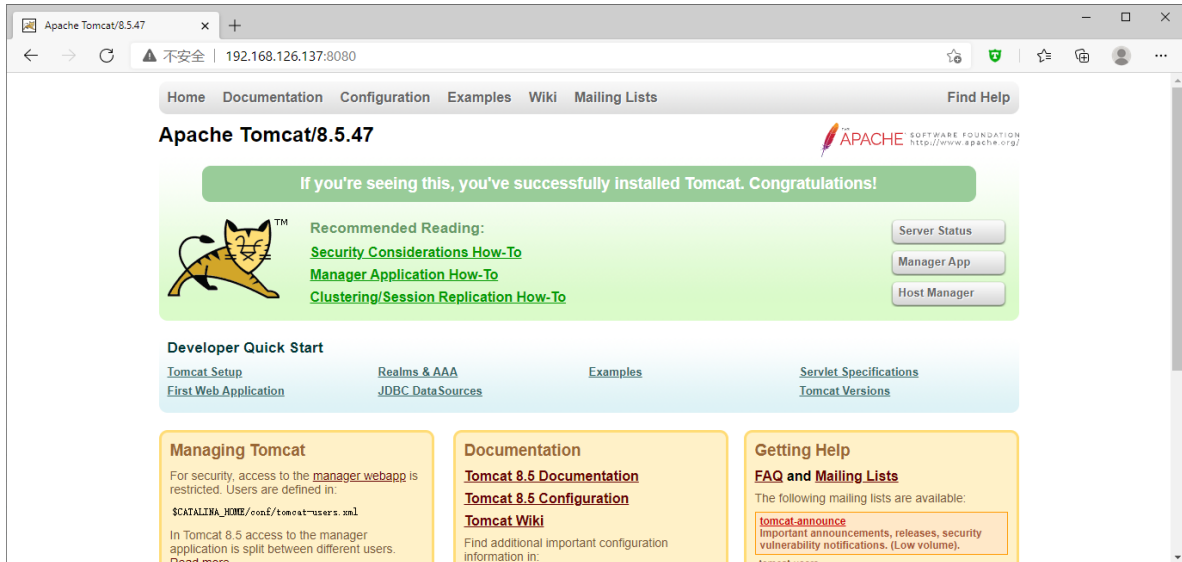


```
[root@localhost logs]# systemctl stop firewalld
```

- 开放8080端口:

```
[root@localhost ~]# firewall-cmd --zone=public --add-port=8080/tcp --permanent
success
[root@localhost ~]# firewall-cmd --reload
success
```

- 开启端口: `firewall-cmd --zone=public --add-port=8080/tcp --permanent`
- 关闭端口: `firewall-cmd --zone=public --remove-port=8080/tcp --permanent`
- 重启防火墙: `firewall-cmd --reload`



## 5、关闭Tomcat

```
./shutdown.sh
```

```
[root@localhost bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/local/apache-tomcat-8.5.47
Using CATALINA_HOME:   /usr/local/apache-tomcat-8.5.47
Using CATALINA_TMPDIR: /usr/local/apache-tomcat-8.5.47/temp
Using JRE_HOME:        /usr/local/jdk1.8.0_231
Using CLASSPATH:       /usr/local/apache-tomcat-8.5.47/bin/bootstrap.jar:/usr/local/apache-tomcat-8.5.47/bin/tomcat-juli.jar
```

# 八、服务端应用程序环境搭建及项目部署

## 安装MySQL

### 1、版本清理

在CentOS7中，预装了一个MariaDB的数据库组件，它是MySQL的一个分支，所以安装MySQL前先把它卸载掉

```
[root@localhost ~]# rpm -qa | grep mariadb
mariadb-libs-5.5.64-1.el7.x86_64
[root@localhost ~]#
```

- 1) 卸载: `yum remove mariadb`

```
[root@localhost ~]# yum remove mariadb
已加载插件：fastestmirror
参数 mariadb 没有匹配
不删除任何软件包
```

2) 删除配置文件: `rm -f /etc/my.cnf`

```
[root@localhost ~]# cat /etc/my.cnf
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock

[root@localhost ~]# rm -f /etc/my.cnf
```

3) 删除数据目录 (可能没有): `rm -rf /var/lib/mysql`

```
[root@localhost ~]# rm -rf /var/lib/mysql
```

## 2、安装方式

- RPM安装包: 将rpm文件发送到Linux, 通过rpm命令安装, 需要安装MySQL相关的依赖组件
- yum安装: 需要连接网络, 另外为了防止镜像站点连接不上, 一般需要配置国内镜像站点, 安装过程中自动配置, 比较简单
- 压缩包解压: 官网下载压缩包到Linux中, 需要配置

安装前建议使用虚拟机的快照功能建立快照, 如果安装失败可以通过快照恢复。

## 3、yum安装

yum安装会查找一个镜像站点来下载, 可能选择不是国内的源, 导致下载速度比较慢, 可以更改源为国内的镜像站点, 例如阿里、163的源:

### 阿里云yum源配置:

1) 备份当前yum源防止出现意外还可以还原回来

```
cd /etc/yum.repos.d/
```

```
cp CentOS-Base.repo CentOS-Base-repo.bak
```

```
[root@localhost ~]# cd /etc/yum.repos.d/
[root@localhost yum.repos.d]# ls
CentOS-Base.repo  CentOS-Debuginfo.repo  CentOS-Media.repo  CentOS-Vault.repo
CentOS-CR.repo   CentOS-fasttrack.repo  CentOS-Sources.repo
[root@localhost yum.repos.d]# cp CentOS-Base.repo CentOS-Base-repo.bak
[root@localhost yum.repos.d]# ls
CentOS-Base.repo  CentOS-CR.repo  CentOS-fasttrack.repo  CentOS-Sources.repo
CentOS-Base-repo.bak  CentOS-Debuginfo.repo  CentOS-Media.repo  CentOS-Vault.repo
```

2) 使用wget下载阿里yum源repo文件

```
wget http://mirrors.aliyun.com/repo/Centos-7.repo
```

注意: 最小化安装没有wget工具, 需要先使用yum安装: `yum -y install wget`

3) 清理旧包

```
yum clean all
```

4) 把下载下来阿里云repo文件设置为默认源

```
mv Centos-7.repo CentOS-Base.repo
```

```
[root@localhost yum.repos.d]# ls
Centos-7.repo          CentOS-CR.repo          CentOS-Media.repo
CentOS-Base.repo       CentOS-Debuginfo.repo   CentOS-Sources.repo
CentOS-Base.repo.bak   CentOS-fasttrack.repo   CentOS-Vault.repo
[root@localhost yum.repos.d]# mv Centos-7.repo CentOS-Base.repo
mv: 是否覆盖"CentOS-Base.repo"? y
[root@localhost yum.repos.d]# ls
CentOS-Base.repo        CentOS-CR.repo           CentOS-fasttrack.repo   CentOS-Sources.repo
CentOS-Base.repo.bak    CentOS-Debuginfo.repo    CentOS-Media.repo       CentOS-Vault.repo
```

#### 5) 生成阿里云yum源缓存并更新yum源

```
yum makecache
```

```
yum update
```

### 安装过程

#### 1) 下载并安装MySQL官方的 Yum Repository:

```
wget -i -c http://dev.mysql.com/get/mysql57-community-release-el7-10.noarch.rpm
```

#### 2) 安装下载的包:

```
yum -y install mysql57-community-release-el7-10.noarch.rpm
```

#### 3) 正式安装mysql (联网下载, 速度较慢) :

```
yum -y install mysql-community-server
```

#### 4) 启动服务

```
systemctl start mysqld
```

## 4、压缩包解压安装

#### 1) 清理

- Step 1.

检查是否已经安装过mysql, 执行命令: `rpm -qa | grep mysql`

如果已经安装过, 卸载掉: `rpm -e --nodeps mysql-libs-5.1.73-5.el6_6.x86_64`

卸载后再次检查

- Step 2.

查询系统现有MySQL对应的文件夹

```
whereis mysql
```

```
find -name mysql
```

```
[root@localhost ~]# whereis mysql
mysql: /usr/lib64/mysql /usr/share/mysql
[root@localhost ~]# find -name mysql
```

- Step 3.

删除相关目录和文件

```
rm -rf /usr/lib64/mysql /usr/share/mysql
```

```
[root@localhost ~]# rm -rf /usr/lib64/mysql /usr/share/mysql
[root@localhost ~]# whereis mysql
mysql:[root@localhost ~]# find -name mysql
[root@localhost ~]#
```

2) 检查mysql用户组和用户是否存在，如果没有，则创建

```
[root@localhost ~]# cat /etc/group | grep mysql
[root@localhost ~]# cat /etc/passwd | grep mysql
[root@localhost ~]#
```

```
[root@localhost ~]# groupadd mysql
[root@localhost ~]# useradd -r -g mysql mysql
```

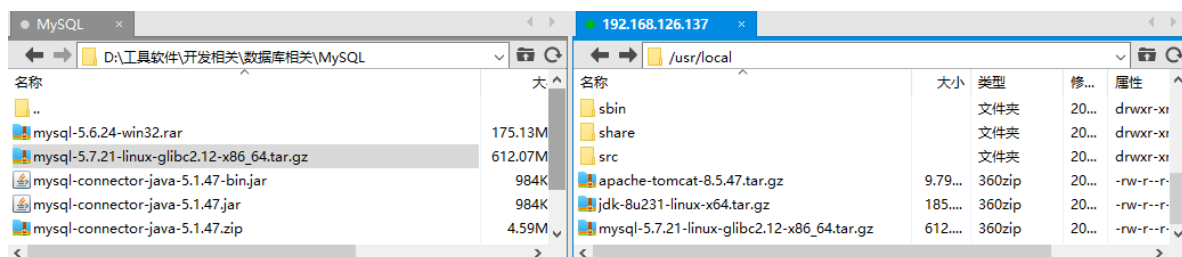
`-r` 表示是一个系统用户。

3) 从官网下载压缩包tar.gz

<https://downloads.mysql.com/archives/community/>

|  |                 |              |
|--|-----------------|--------------|
| Product Version:   | 5.7.21          |              |
| Operating System:  | Linux - Generic |              |
| OS Version:  | All             |              |
|  |                 |              |
| Linux - Generic (glibc 2.12) (x86, 32-bit), Compressed TAR Archive |                 | Dec 28, 2017 |
| (mysql-5.7.21-linux-glibc2.12-i686.tar.gz)                         |                 |              |
| Linux - Generic (glibc 2.12) (x86, 64-bit), Compressed TAR Archive |                 | Dec 28, 2017 |
| (mysql-5.7.21-linux-glibc2.12-x86_64.tar.gz)                       |                 |              |

4) 通过Xftp传输到Linux系统中，之后解压缩并改名为/usr/local/mysql中



```
[root@localhost ~]# cd /usr/local/
[root@localhost local]# ls
apache-tomcat-8.5.47      lib
apache-tomcat-8.5.47.tar.gz  lib64
bin                       libexec
etc                       mysql-5.7.21-linux-glibc2.12-x86_64.tar.gz
games                    sbin
include                 share
jdk1.8.0_231            src
jdk-8u231-linux-x64.tar.gz
[root@localhost local]# tar -zxvf mysql-5.7.21-linux-glibc2.12-x86_64.tar.gz
mysql-5.7.21-linux-glibc2.12-x86_64/bin/mysam_ftdump
mysql-5.7.21-linux-glibc2.12-x86_64/bin/mysamchk
mysql-5.7.21-linux-glibc2.12-x86_64/bin/mysamlog
```

```
[root@localhost local]# ls
apache-tomcat-8.5.47      lib
apache-tomcat-8.5.47.tar.gz lib64
bin                       libexec
etc                       mysql-5.7.21-linux-glibc2.12-x86_64
games                    mysql-5.7.21-linux-glibc2.12-x86_64.tar.gz
include                  sbin
jdk1.8.0_231             share
jdk-8u231-linux-x64.tar.gz src
[root@localhost local]# mv mysql-5.7.21-linux-glibc2.12-x86_64 mysql
[root@localhost local]# ls
apache-tomcat-8.5.47      lib
apache-tomcat-8.5.47.tar.gz lib64
bin                       libexec
etc                       mysql
games                    mysql-5.7.21-linux-glibc2.12-x86_64.tar.gz
include                  sbin
jdk1.8.0_231             share
jdk-8u231-linux-x64.tar.gz src
[root@localhost local]#
```

##### 5) 在/usr/local/mysql目录下创建data目录

```
[root@localhost local]# mkdir /usr/local/mysql/data
[root@localhost local]# cd mysql
[root@localhost mysql]# ls
bin  COPYING  data  docs  include  lib  man  README  share  support-files
```

##### 6) 更改mysql目录下所有的目录及文件夹所属的用户组和用户，以及权限

```
[root@localhost mysql]# chown -R mysql:mysql /usr/local/mysql
[root@localhost mysql]# chmod -R 755 /usr/local/mysql
```

##### 7) 编译安装并初始化mysql, 务必记住初始化输出日志末尾的密码 (数据库管理员临时密码)

```
./mysqld --initialize --user=mysql --datadir=/usr/local/mysql/data --
basedir=/usr/local/mysql
```

正确运行后，将会输出很多内容日志，其中将会生成一个临时的密码，需要记住！

```
[root@localhost bin]# ./mysqld --initialize --user=mysql --datadir=/usr/local/mysql/data --basedir=/usr/local/mysql
2021-03-09T13:12:33.819251Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp server option (see documentation for more details).
2021-03-09T13:12:34.053455Z 0 [Warning] InnoDB: New log files created, LSN=45790
2021-03-09T13:12:34.108161Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.
2021-03-09T13:12:34.181545Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has been started. Generating a new UUID: 1ca5c028-80d9-11eb-9c64-000c29b09b3f.
2021-03-09T13:12:34.182859Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid_executed' cannot be opened.
2021-03-09T13:12:34.183784Z 1 [Note] A temporary password is generated for root@localhost: QI+g!7WeS8di
[root@localhost bin]#
```

##### 8) 编辑配置文件my.cnf，添加配置信息

```
vi /etc/my.cnf
```

```
[mysqld]
basedir=/usr/local/mysql
datadir=/usr/local/mysql/data
socket=/usr/local/mysql/data/mysql.sock
port=3306
log-error=/usr/local/mysql/data/error.log
```

```
[client]
socket=/usr/local/mysql/data/mysql.sock
```

## 9) 移动文件

```
cp ./support-files/mysql.server /etc/init.d/mysqld
```

```
[root@localhost bin]# cd ..
[root@localhost mysql]# ls
bin  COPYING  data  docs  include  lib  man  README  share  support-files
[root@localhost mysql]#
[root@localhost mysql]# cp ./support-files/mysql.server /etc/init.d/mysqld
```

## 10) 启动MySQL服务

进入 `/etc/init.d`，通过拷贝过来的文件启动服务

```
[root@localhost mysql]# cd /etc/init.d/
[root@localhost init.d]# ls
functions  mysqld  netconsole  network  README
[root@localhost init.d]# ./mysqld start
Starting MySQL.Logging to '/usr/local/mysql/data/error.log'.
SUCCESS!
[root@localhost init.d]#
```

此时，通过 `systemctl` 命令查看服务，可能是查看不到的：

```
[root@localhost ~]# systemctl status mysqld
Unit mysqld.service could not be found.
```

但是通过 `service` 命令可以看到其正在运行中：

```
[root@localhost ~]# service mysqld status
SUCCESS! MySQL running (5326)
```

为了后续方便启动，可以做一个软链接到环境变量中：

```
ln -s /usr/local/mysql/bin/mysql /usr/bin/
```

# 进入数据库

## 使用临时密码进入数据库：

输入密码时是看不见的，小心

```
[root@localhost ~]# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.21
```

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> █
```

使用临时密码进入数据库，此时不能做任何操作，系统会提示必须要重新设置密码：

```
mysql> show databases;
ERROR 1820 (HY000): You must reset your password using ALTER USER statement before executing this statement.
mysql> █
```

## 修改密码：

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'root';
Query OK, 0 rows affected (0.00 sec)
```

如果已经有密码存在，需要修改的话：

- 方法1：用SET PASSWORD命令

```
mysql -u root
mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('newpass');
```

- 方法2：用mysqladmin

```
mysqladmin -u root password "newpass"
-- 如果root已经设置过密码，采用如下方法
mysqladmin -u root password oldpass "newpass"
```

- 方法3：用UPDATE直接编辑user表

```
mysql -u root
mysql> use mysql;
mysql> UPDATE user SET Password = PASSWORD('newpass') WHERE user = 'root';
mysql> FLUSH PRIVILEGES;
```

- 在丢失root密码的时候，可以这样

```
mysqld_safe --skip-grant-tables&
mysql -u root mysql
mysql> UPDATE user SET password = PASSWORD("new password") WHERE user='root';
mysql> FLUSH PRIVILEGES;
```

## 远程权限

数据库默认是不允许远程访问的，是为了保护数据。



如果想要通过本机的数据库管理工具方便的管理远程服务器上的数据库数据，或者服务端需要连接远程服务器上的数据库，则需要开放相应的远程访问权限。

## 1. 修改用户权限

1) 到MySQL库的user表中

```
update user set host='',password=password('123456') where host='127.0.0.1';
flush privileges;
```

## 2. 开放远程访问端口

关闭Linux的防火墙或者开放3306端口保证可以访问

## 服务端本地编码

为了防止本地客户端数据乱码，在 `/etc/my.cnf` 中添加如下配置：

```
[client]
default-character-set=utf8
```

这个配置只作用于客户端，要保证程序中的数据不乱码，需要建库、建表时指定统一的字符集。

## 卸载

`rpm -e mysql-server`：只需要卸载主服务就行了

`rm -rf /var/lib/mysql` 删除所有mysql的数据，不同安装方式路径不一样

## \*root密码找回（了解）

到 `/etc/my.cnf` 下添加配置

```
[mysqld]
...
skip-grant-tables #注意，建议在断网的情况下添加，添加后需要重启mysqld服务）
```

然后，`mysql -uroot -p` 即可登录mysql，登录后修改密码：

```
update user set pass.....修改密码
flush privileges;
```

之后删除 `my.cnf` 中的添加的内容

## 项目部署



使用Xftp传输项目输出的war包到tomcat的 `webapps` 目录下，重启tomcat后进行访问测试。