

Linux

一、Linux简介

- (一) 什么是Linux
- (二) 为什么要用Linux?
- (三) Linux 系统

二、远程连接Linux

- (一) Linux IP配置
- (二) SSH命令连接Linux
- (三) 远程连接工具

三、Linux 初识

- (一) Linux 关机
- (二) 系统目录结构

四、Linux基本命令

- (一) 目录操作
 - 1. 相对路径和绝对路径
 - 2. 切换目录
 - 3. 列出目录下的内容
 - 4. 创建目录
 - 5. 删除目录
 - 6. 创建文件
 - 7. 删除文件
 - 8. 拷贝
 - 9. 移动
 - 10. Linux 短语
- (二) 权限属性操作
 - 1. 查看权限属性
 - 2. 操作权限
- (三) 文件查看
- (四) 文件编辑
 - 1. vi/vim编辑器
 - 2. 文件编辑
- (五) 用户账号
 - 创建用户
 - 密码管理
 - 修改用户
 - 切换用户
 - 删除用户
- (六) 解压缩
- (七) 进程、端口
 - 查看进程
 - 结束进程
 - 端口号

五、软件安装

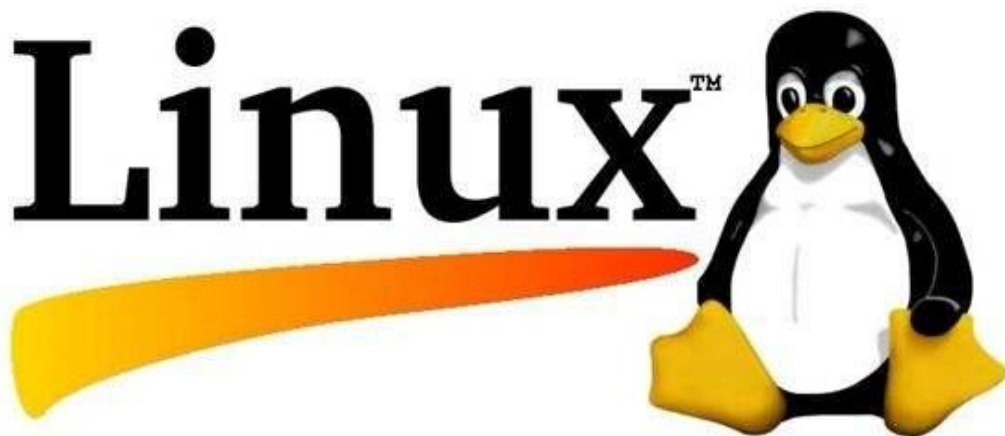
- (零) 软件准备
- (一) Linux安装JDK
 - 1. 卸载Linux默认的JDK
 - 2. 安装自定义JDK
- (二) Linux安装Tomcat
- (三) Linux安装MySQL8

Linux

一、Linux简介

(一) 什么是Linux

常用的操作系统：Windows（图形化）、**Linux（命令行）**、MacOS（图形化、命令行）、鸿蒙OS、优麒麟、安卓、IOS、塞班.....



Linux，全称GNU/Linux，是一种免费使用和自由传播的类UNIX操作系统，其内核由**林纳斯·本纳第克特·托瓦兹**于1991年10月5日首次发布，它主要受到Minix和Unix思想的启发，是一个基于POSIX的**多用户、多任务、支持多线程和多CPU**的操作系统。它能运行主要的Unix工具软件、应用程序和网络协议。它支持32位和64位硬件。Linux继承了Unix以**网络为核心**的设计思想，是一个性能稳定的**多用户网络操作系统**。Linux 有上百种不同的发行版，如基于社区开发的debian、archlinux，和基于商业开发的Red Hat Enterprise Linux、SUSE、Oracle Linux等。

(二) 为什么要用Linux?

- Linux优点：

- 1、**免费开源**。Linux是一款完全免费的操作系统，任何人都可以从网络上下载到它的源代码，并且可以根据自己的需求进行定制化的开发，而且没有版权限制。
- 2、**模块化程度高**。Linux的内核设计分成进程管理、内存管理、进程间通信、虚拟文件系统、网络五部分，其采用的模块机制使得用户可以根据实际需要，在内核中插入或移走模块，这使得内核可以被高度的剪裁定制，以方便在不同的场景下使用。
- 3、**Linux系统广泛的硬件支持**。得益于其免费开源的特点，有大批程序员不断地向Linux社区提供代码，使得Linux有着异常丰富的设备驱动资源，对主流硬件的支持极好，而且几乎能运行在所有流行的处理器上。
- 4、**安全稳定**。Linux采取了很多安全技术措施，包括读写权限控制、带保护的子系统、审计跟踪、核心授权等，这为网络环境中的用户提供了安全保障。实际上有很多运行 Linux 的服务器可以持续运行长达数年而无须重启，依然可以性能良好地提供服务，其安全稳定性已经在各个领域得到了广泛的证实。

5、多用户，多任务。多用户是指系统资源可以同时被不同的用户使用，每个用户对自己的资源有特定的权限，互不影响。多任务是现代化计算机的主要特点，指的是计算机能同时运行多个程序，且程序之间彼此独立，Linux内核负责调度每个进程，使之平等地访问处理器。由于CPU处理速度极快，从用户的角度来看所有的进程好像在并行运行。

6、良好的可移植性。Linux中95%以上的代码都是用C语言编写的，由于C语言是一种机器无关的高级语言，是可移植的，因此Linux系统也是可移植的。

- 基于以上优点，企业的服务器大多使用 Linux 系统，开发的很多项目也是发布在 Linux 系统上；
- Java 的很多技术（Redis、MQ、JDK、Maven）是基于 Linux 开发；

(三) Linux 系统

- 虚拟机安装（无成本、性能低）
- 物理机安装（无成本、性能高）
- 购买云服务器（成本、性能高）：阿里云、华为云、腾讯云...

二、远程连接Linux

作为服务器，服务器是统一管理放在机房，不建议直接操作物理机。另外现在很多情况下都是云服务器，更不可能找到物理机进行操作，所以需要远程操作。远程操作就需要知道 Linux 的 IP 地址。

(一) Linux IP配置

- 方式一：通过图形界面配置
- 方式二：通过命令行配置

1) 输入命令 `ifconfig` 查看默认配置，第一行可以看到配置网络的文件名称 `ens33`。有些系统是 `eth0` 可能不一样，注意区分；

```
[root@localhost ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.23.123  netmask 255.255.255.0  broadcast 192.168.23.255
    inet6 fe80::7c87:83b6:c218:862f  prefixlen 64  scopeid 0x20<link>
    ether 00:0c:29:b3:cf:34  txqueuelen 1000  (Ethernet)
    RX packets 282  bytes 26600 (25.9 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 243  bytes 34317 (33.5 KiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 68  bytes 5916 (5.7 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 68  bytes 5916 (5.7 KiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
    inet 192.168.122.1  netmask 255.255.255.0  broadcast 192.168.122.255
    ether 52:54:00:ea:10:12  txqueuelen 1000  (Ethernet)
    RX packets 0  bytes 0 (0.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2) 输入命令 `cd /etc/sysconfig/network-scripts` 查看实际文件名

```
[root@localhost ~]# cd /etc/sysconfig/network-scripts
[root@localhost network-scripts]# ll
总用量 252
-rw-r--r--. 1 root root 404 4月 10 11:11 ifcfg-ens33
-rw-r--r--. 1 root root 254 8月 24 2018 ifcfg-lo
lrwxrwxrwx. 1 root root 24 4月 10 10:07 ifdown -> ../../usr/sbin/ifdown
-rwxr-xr-x. 1 root root 654 8月 24 2018 ifdown-bnep
-rwxr-xr-x. 1 root root 6532 8月 24 2018 ifdown-eth
```

3) 通过vi或者vim命令查看并修改文件，按 **i** 键，进入输入模式，添加 **IPADDR** **NETMASK** **GATEWAY**
DNS1 **DNS2** 配置

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens33
UUID=0953eb94-0e4a-4cbd-b366-3369177790f1
DEVICE=ens33
ONBOOT=yes
DNS1=114.114.114.114
DNS2=8.8.8.8
IPADDR=192.168.23.123
NETMASK=255.255.255.0
GATEWAY=192.168.23.2
~
```

4) 输入命令 `service network restart` 重启网卡即可；

(二) SSH命令连接Linux

SSH 命令是 Windows 10/11 提供的用于连接 Linux 系统的命令，在 cmd 命令提示符窗口下输入 `ssh user@ip` 即可远程连接Linux。

例如： `ssh root@192.168.88.88`

```
root@localhost:~  
Microsoft Windows [版本 10.0.19043.1165]  
(c) Microsoft Corporation。保留所有权利。  
  
C:\Users\Riu>ssh root@192.168.88.88  
The authenticity of host '192.168.88.88 (192.168.88.88)' can't be established.  
ECDSA key fingerprint is SHA256:Cdm5VHQh/P115Qs6mX9K1WJHw0AE+yk00faHHGgz7Fs.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.88.88' (ECDSA) to the list of known hosts.  
root@192.168.88.88's password:  
Last login: Mon Aug 30 19:53:25 2021  
[root@localhost ~]#
```

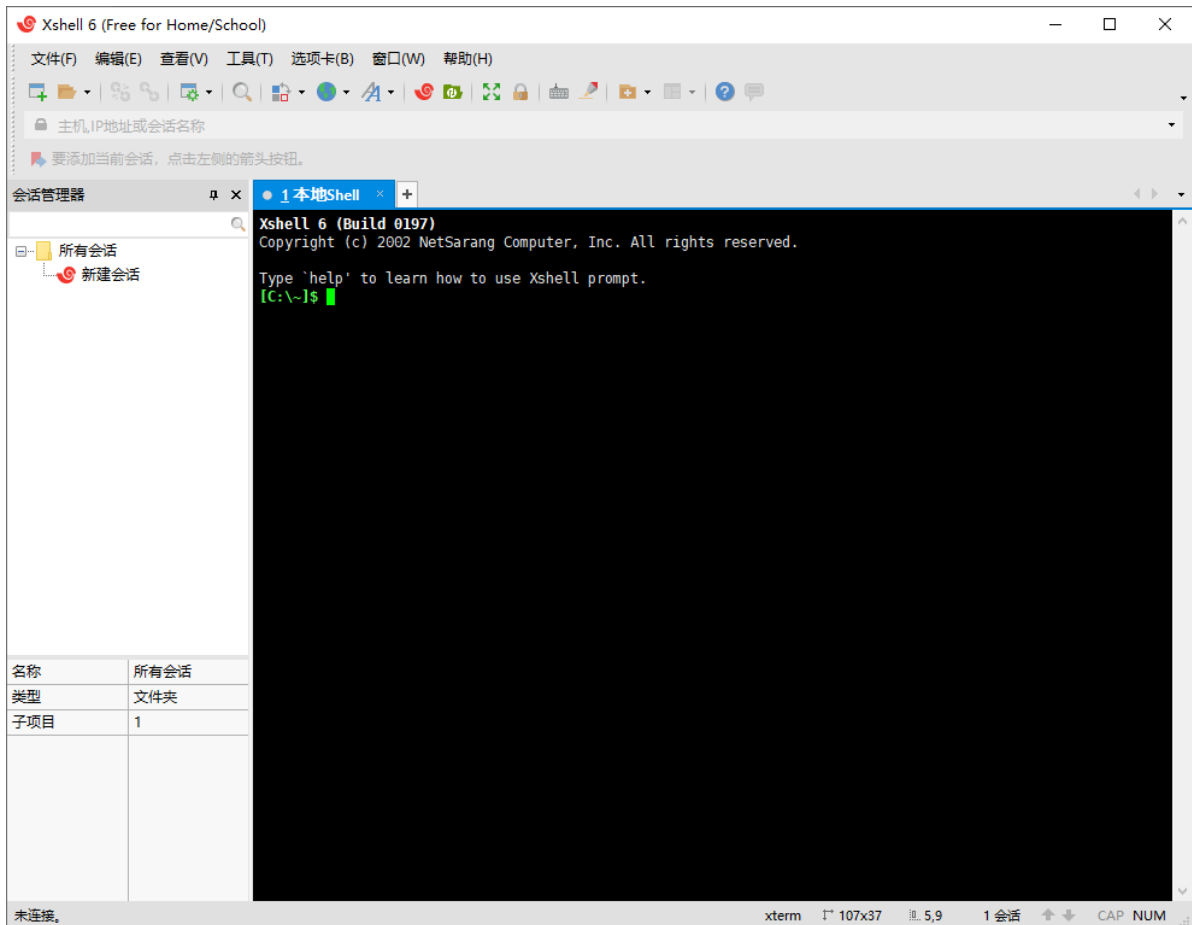
输入yes, 同意连接Linux

输入密码, Linux密码是不显示的, 一定要一次性输入正确

看着这个标识, 就说明连接成功了

(三) 远程连接工具

真实工作中服务器都是统一管理的, 不可能直接接触操作, 一般情况下都是通过远程访问的形式进行。常用的远程连接软件: SSH、XShell



- 软件配置

类别(C):

- 连接
 - 用户身份验证
 - 登录提示符
 - 登录脚本
 - SSH
 - 安全性
 - 隧道
 - SFTP
 - TELNET
 - RLOGIN
 - 串口
 - 代理
 - 保持活动状态
- 终端
 - 键盘
 - VT 模式
 - 高级
- 外观
 - 窗口
 - 突出
- 高级
 - 跟踪
 - 响铃
 - 日志记录
- 文件传输
 - X/YMODEM
 - ZMODEM

连接

常规

名称(N): 新建会话

协议(P): SSH

主机(H): 192.168.23.134

端口号(O): 22

说明(D):

重新连接

☐ 连接异常关闭时自动重新连接(A)

间隔(V): 0 秒

限制(L): 0 分钟

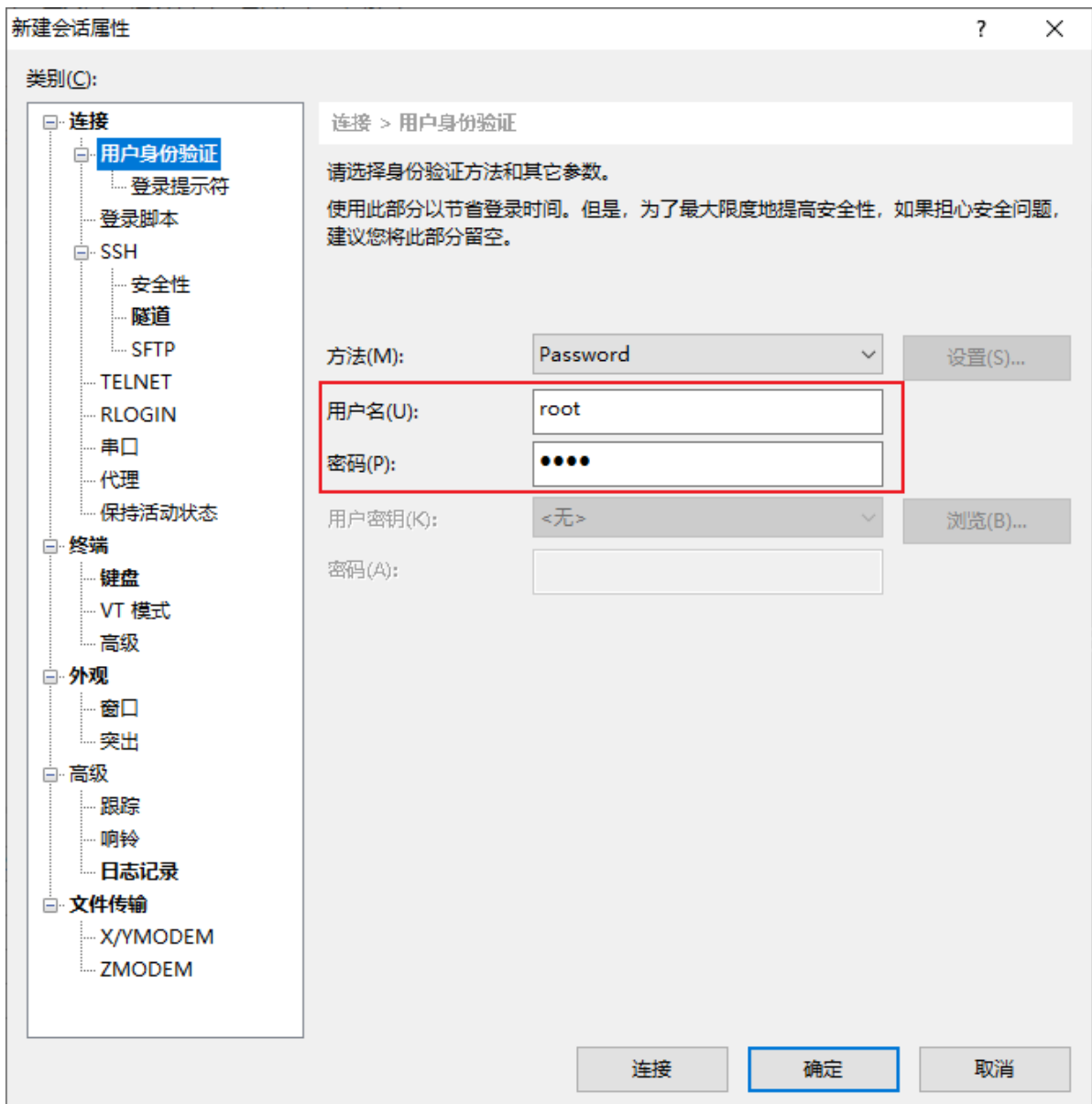
TCP选项

☐ 使用Nagle算法(U)

连接

确定

取消



三、Linux 初识

(一) Linux 关机

```
1 # 关机命令
2 shutdown
3 shutdown -h 5 # 5分钟后关机
4 shutdown -h now # 立刻关机
5 shutdown -h 12:00 # 在某个时间点关机
6
7 halt # 立刻关机
8 poweroff # 立刻关机
9
10 # 重启命令
11 shutdown -r now # 立刻重启
12 shutdown -r +5 # 5分钟后重启
13 reboot # 立刻重启
```

(二) 系统目录结构

Linux 下一切皆文件;

Linux 根目录 `/`，所有的文件都在这个目录下

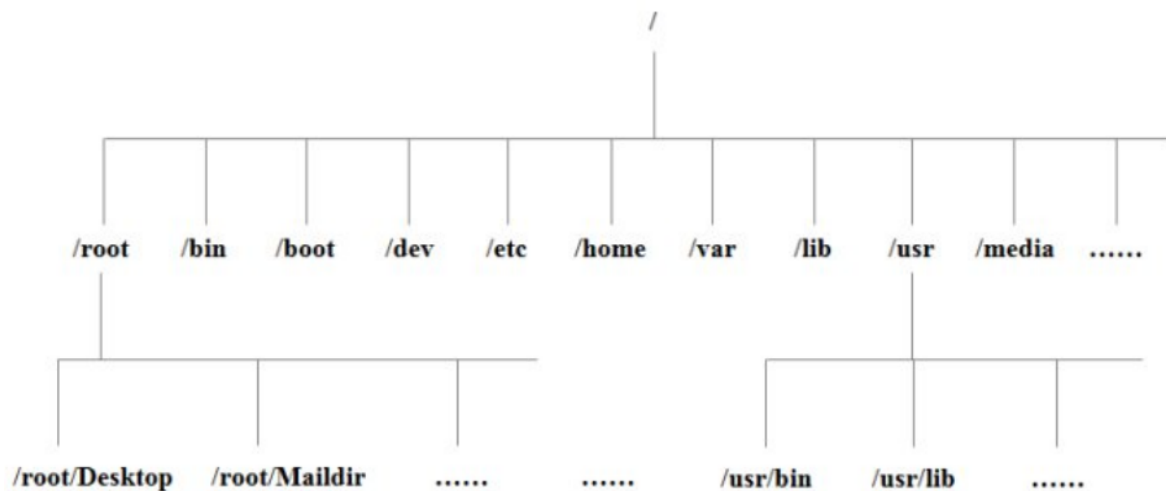
登录系统后，在命令窗口输入：

```
1 # 展示根目录下内容
2 ls /
```

输入命令后会看到下图所示内容：

```
[root@localhost ~]# ls /
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[root@localhost ~]#
```

树状结构：



目录解释：

- **/bin**：bin 是 Binaries (二进制文件) 的缩写, 这个目录存放着最经常使用的命令，例如：vi、vim 等。
- **/boot**：这里存放的是启动 Linux 时使用的一些核心文件，包括一些连接文件以及镜像文件。
- **/dev**：dev 是 Device(设备) 的缩写，该目录下存放的是 Linux 的外部设备（U盘），在 Linux 中访问设备的方式和访问文件的方式是相同的。
- **/etc**：etc 是 Etcetera(等等) 的缩写，这个目录用来存放所有的系统管理所需要的**配置文件**和子目录。
- **/home**：用户的主目录，在 Linux 中，每个用户都有一个自己的目录，一般该目录名是以用户的**账号命名的**，类似于Windows平台下 `C:\Users` 目录
- **/lib**：lib 是 Library(库) 的缩写这个目录里存放着系统最基本的动态连接共享库，其作用类似于 Windows 里的 DLL 文件。几乎所有的应用程序都需要用到这些共享库。
- **/lost+found**：这个目录一般情况下是空的，当系统非法关机后，这里就存放了一些文件。
- **/media**：Linux 系统会自动识别一些设备，例如U盘、光驱等等，当识别后，Linux 会把识别的设备挂载到这个目录下。
- **/mnt**：系统提供该目录是为了让用户临时挂载别的文件系统的，我们可以将光驱挂载在 `/mnt/` 上，然后进入该目录就可以查看光驱里的内容了。

- **/opt**: opt 是 optional (可选) 的缩写, 这是给主机**额外安装软件**所摆放的目录。比如你安装一个ORACLE数据库则就可以放到这个目录下。默认是空的。
- **/proc**: proc 是 Processes(进程) 的缩写, /proc 是一种伪文件系统 (也即虚拟文件系统), 存储的是当前内核运行状态的一系列特殊文件, 这个目录是一个虚拟的目录, 它是系统内存的映射, 我们可以通过直接访问这个目录来获取系统信息。这个目录的内容不在硬盘上而是在内存里, 我们也可以直接修改里面的某些文件, 比如可以通过下面的命令来屏蔽主机的ping命令, 使别人无法ping你的机器:

```
1 | echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

- **/root**: 该目录为系统管理员, 也称作超级权限者的用户主目录。
- **/sbin**: s 就是 Super User 的意思, 是 Superuser Binaries (超级用户的二进制文件) 的缩写, 这里存放的是系统管理员使用的系统管理程序。
- **/selinux**: 这个目录是 Redhat/CentOS 所特有的目录, Selinux 是一个安全机制, 类似于 windows 的防火墙, 但是这套机制比较复杂, 这个目录就是存放selinux相关的文件的。
- **/srv**: 该目录存放一些服务启动之后需要提取的数据。
- **/sys**:

这是 Linux2.6 内核的一个很大的变化。该目录下安装了 2.6 内核中新出现的一个文件系统 sysfs。sysfs 文件系统集成了下面3种文件系统的信息: 针对进程信息的 proc 文件系统、针对设备的 devfs 文件系统以及针对伪终端的 devpts 文件系统。

该文件系统是内核设备树的一个直观反映。

当一个内核对象被创建的时候, 对应的文件和目录也在内核对象子系统中被创建。

- **/tmp**: tmp 是 temporary(临时) 的缩写这个目录是用来存放一些临时文件的。
- **/usr**: **usr 是 unix shared resources(共享资源) 的缩写, 这是一个非常重要的目录, 用户的很多应用程序和文件都放在这个目录下, 类似于 windows 下的 program files 目录。**
- **/usr/bin**: 系统用户使用的应用程序。
- **/usr/sbin**: 超级用户使用的比较高级的管理程序和系统守护程序。
- **/usr/src**: 内核源代码默认的放置目录。
- **/var**: var 是 variable(变量) 的缩写, 这个目录中存放着在不断扩充着的东西, 我们习惯将那些经常被修改的目录放在这个目录下。包括各种日志文件。
- **/run**: 是一个临时文件系统, 存储系统启动以来的信息。当系统重启时, 这个目录下的文件应该被删掉或清除。如果你的系统上有 /var/run 目录, 应该让它指向 run。

四、Linux基本命令

(一) 目录操作

常用命令:

```
1 ls # 列出目录
2 cd # 切换目录
3 pwd # 显示目前的目录
4 mkdir # 创建一个新的目录
5 rmdir # 删除一个空的目录
6 cp # 复制文件或目录
7 rm # 移除文件或目录
8 mv # 移动文件与目录，或修改文件与目录的名称
```

1. 相对路径和绝对路径

- **绝对路径：**

基于根，某一个文件在某个目录下的绝对位置。比如：`D:\863\a.txt`、`/etc/xx.conf`。可以理解为邮寄包裹时填写的路径，就是绝对路径。

- **相对路径：**

某一个文件相对另一个目录的位置。比如：`863\a.txt`，在863目录下的a.txt。可以理解为住在楼上楼下的邻居，是一个相对的位置，不是一个具体的位置。

- **目录操作：** 目录操作时，命令后面可以跟相对路，也可以跟绝对路径

2. 切换目录

```
1 # cd (change directory) 命令，用于切换目录
2 # 语法结构：
3 cd [相对路径或绝对路径]
4 # 常用标识：
5 ../ # 上一层目录
6 ./ # 当前目录
7 / # 根目录
8 ~ # 用户目录
9
10 eg:
11 cd /home # 切换到home目录；
12 cd ../etc # 切换到上一级目录下的 etc 目录；
13 cd ../ # 切换到上一层目录
14 cd / # 切换到根目录
15 cd ~ # 切换到home目录
```

```
[root@localhost ~]# cd home
[root@localhost home]# cd ..
[root@localhost ~]# cd ~
[root@localhost ~]# cd /
[root@localhost /]#
```

3. 列出目录下的内容

通俗点说就是打开文件夹，查看里面的内容。

```
1 ls # 查看当前目录下的所有文件夹以及文件，不显示文件的属性等信息，只展示名字。
2 ls -a # 查看当前目录下的所有文件夹以及文件，包含隐藏文件（.开头的文件），不显示文件的属性等信息，只展示名字。
3 ls -la # 查看当前目录下的所有文件夹以及文件，包含隐藏文件，以及属性等信息
4 ls -l 和 ll # 查看当前目录下的所有文件夹以及文件，以及属性等信息，不包含隐藏文件
```

```
[root@localhost /]# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
[root@localhost /]# ls -a
. .. bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
[root@localhost /]#
```

```
[root@localhost /]# ls -la
总用量 28
dr-xr-xr-x. 17 root root 224 4月 10 10:22 .
dr-xr-xr-x. 17 root root 224 4月 10 10:22 ..
lrwxrwxrwx. 1 root root 7 4月 10 10:04 bin -> usr/bin
dr-xr-xr-x. 5 root root 4096 4月 10 10:24 boot
drwxr-xr-x. 20 root root 3320 4月 10 11:12 dev
drwxr-xr-x. 143 root root 8192 4月 10 11:28 etc
drwxr-xr-x. 3 root root 17 4月 10 10:32 home
lrwxrwxrwx. 1 root root 7 4月 10 10:04 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 4月 10 10:04 lib64 -> usr/lib64
drwxr-xr-x. 2 root root 6 4月 11 2018 media
drwxr-xr-x. 2 root root 6 4月 11 2018 mnt
drwxr-xr-x. 3 root root 16 4月 10 10:12 opt
dr-xr-xr-x. 220 root root 0 4月 10 11:12 proc
dr-xr-xr-x. 15 root root 4096 4月 10 11:17 root
drwxr-xr-x. 41 root root 1240 4月 10 11:18 run
lrwxrwxrwx. 1 root root 8 4月 10 10:04 sbin -> usr/sbin
drwxr-xr-x. 2 root root 6 4月 11 2018 srv
dr-xr-xr-x. 13 root root 0 4月 10 11:12 sys
drwxrwxrwt. 24 root root 4096 4月 10 12:54 tmp
drwxr-xr-x. 13 root root 155 4月 10 10:04 usr
drwxr-xr-x. 21 root root 4096 4月 10 10:30 var
[root@localhost /]#
```

```
[root@localhost /]# ll
总用量 28
lrwxrwxrwx. 1 root root 7 4月 10 10:04 bin -> usr/bin
dr-xr-xr-x. 5 root root 4096 4月 10 10:24 boot
drwxr-xr-x. 20 root root 3320 4月 10 11:12 dev
drwxr-xr-x. 143 root root 8192 4月 10 11:28 etc
drwxr-xr-x. 3 root root 17 4月 10 10:32 home
lrwxrwxrwx. 1 root root 7 4月 10 10:04 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 4月 10 10:04 lib64 -> usr/lib64
drwxr-xr-x. 2 root root 6 4月 11 2018 media
drwxr-xr-x. 2 root root 6 4月 11 2018 mnt
drwxr-xr-x. 3 root root 16 4月 10 10:12 opt
dr-xr-xr-x. 220 root root 0 4月 10 11:12 proc
dr-xr-xr-x. 15 root root 4096 4月 10 11:17 root
drwxr-xr-x. 41 root root 1240 4月 10 11:18 run
lrwxrwxrwx. 1 root root 8 4月 10 10:04 sbin -> usr/sbin
drwxr-xr-x. 2 root root 6 4月 11 2018 srv
dr-xr-xr-x. 13 root root 0 4月 10 11:12 sys
drwxrwxrwt. 24 root root 4096 4月 10 12:54 tmp
drwxr-xr-x. 13 root root 155 4月 10 10:04 usr
drwxr-xr-x. 21 root root 4096 4月 10 10:30 var
[root@localhost /]#
```

- 1 | `pwd` # **Print Working Directory**的缩写，显示当前所在目录
- 2 | `pwd -P` # 如果是连接（快捷方式），则显示出完成路径

```
[root@localhost bin]# pwd
/bin
[root@localhost bin]# pwd -P
/usr/bin
[root@localhost bin]#
```

4. 创建目录

建议不要在系统文件夹下做操作。

- 1 | # 语法结构: **make directory** 的缩写
- 2 | `mkdir [-mp]` 目录名称
- 3 |
- 4 | # `mkdir` 不带参数时，只支持创建一级目录
- 5 | # 参数含义:
- 6 | `-m` : 创建文件同时配置文件的权限。例如: `mkdir -m 777 a`
- 7 | `-p` : 递归创建目录。例如: `mkdir -p object/src/main/java`

```
[root@localhost riu]# mkdir test
[root@localhost riu]# ll
总用量 0
drwxr-xr-x. 2 root root 6 4月 13 16:16 test
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 公共
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 模板
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 视频
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 图片
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 文档
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 下载
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 音乐
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 桌面
[root@localhost riu]#
```

```
[root@localhost riu]# mkdir test1/test2/test3
mkdir: 无法创建目录“test1/test2/test3”: 没有那个文件或目录
[root@localhost riu]# mkdir -p test1/test2/test3
[root@localhost riu]#
```

```
[root@localhost riu]# mkdir -m 777 test2
[root@localhost riu]# ll
总用量 0
drwxr-xr-x. 2 root root 6 4月 13 16:16 test
drwxr-xr-x. 3 root root 19 4月 13 16:19 test1
drwxrwxrwx. 2 root root 6 4月 13 16:21 test2
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 公共
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 模板
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 视频
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 图片
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 文档
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 下载
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 音乐
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 桌面
[root@localhost riu]#
```

5. 删除目录

- 1 # 语法结构: **remove directory** 的缩写
- 2 **rmdir** [-p] 目录名称
- 3 # 参数含义:
- 4 **-p** : 连同上一级『空的』目录也一起删除。例如: **rmdir -p object/src/main/java**

```
[root@localhost riu]# ll
总用量 0
drwxr-xr-x. 2 root root 6 4月 13 16:16 test
drwxr-xr-x. 3 root root 19 4月 13 16:19 test1
drwxrwxrwx. 2 root root 6 4月 13 16:21 test2
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 公共
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 模板
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 视频
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 图片
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 文档
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 下载
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 音乐
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 桌面
[root@localhost riu]# rmdir test
[root@localhost riu]# rmdir -p test1/test2/test3/
[root@localhost riu]# ll
总用量 0
drwxrwxrwx. 2 root root 6 4月 13 16:21 test2
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 公共
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 模板
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 视频
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 图片
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 文档
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 下载
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 音乐
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 桌面
[root@localhost riu]#
```

- 1 # 语法结构
- 2 **rm** [-fir] 文件或目录
- 3 # 参数含义
- 4 **-f** : 就是 **force** 的意思, 忽略不存在的文件, 不会出现警告信息;
- 5 **-i** : 互动模式, 在删除前会询问使用者是否动作。默认情况下 **Linux** 会自动添加当前参数, 为了保障用户误操作。
- 6 **-r** : 递归删除! 最常用在目录的删除了! 这是非常危险的选项!!!

```
[root@localhost riu]# rm -rf test
[root@localhost riu]# ll
总用量 0
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 公共
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 模板
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 视频
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 图片
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 文档
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 下载
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 音乐
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 桌面
[root@localhost riu]#
```

```
[root@localhost opt]# ll
总用量 0
-rw-r--r--. 1 root root  0 11月  9 09:35 Hello.java
-rw-r--r--. 1 root root  0 11月  9 09:35 index.html
drwxr-xr-x. 3 root root 17 11月  9 09:39 object
drwxr-xr-x. 2 root root  6 3月  26 2015 rh
[root@localhost opt]# rm -r object
rm: 是否进入目录"object"? y
rm: 是否进入目录"object/src"? y
rm: 是否进入目录"object/src/main"? y
rm: 是否进入目录"object/src/main/java"? y
rm: 是否进入目录"object/src/main/java/com"? y
rm: 是否进入目录"object/src/main/java/com/soft"? y
rm: 是否进入目录"object/src/main/java/com/soft/controller"? y
rm: 是否删除普通空文件 "object/src/main/java/com/soft/controller/HelloController.java"? y
rm: 是否删除目录 "object/src/main/java/com/soft/controller"? y
rm: 是否删除目录 "object/src/main/java/com/soft"? y
rm: 是否删除目录 "object/src/main/java/com"? y
rm: 是否删除目录 "object/src/main/java"? y
rm: 是否删除目录 "object/src/main"? y
rm: 是否删除目录 "object/src"? y
rm: 是否删除目录 "object"? y
[root@localhost opt]# ls
Hello.java  index.html  rh
[root@localhost opt]#
```

6. 创建文件

```
1 # 语法结构:
2 touch 文件名
3
4 # 实例:
5 touch a.txt
6 touch Hello.java
7 touch index.html
```

7. 删除文件

```
1 # 语法结构
2 rm [-fir] 文件或目录
3 # 参数含义
4 -f : 就是 force 的意思, 忽略不存在的文件, 不会出现警告信息;
5 -i : 互动模式, 在删除前会询问使用者是否动作
6 -r : 递归删除! 最常用在目录的删除了! 这是非常危险的选项!!!
```

```
[root@localhost opt]# ll
总用量 0
-rw-r--r--. 1 root root  0 11月  9 09:35 a.txt
-rw-r--r--. 1 root root  0 11月  9 09:35 Hello.java
-rw-r--r--. 1 root root  0 11月  9 09:35 index.html
drwxr-xr-x. 3 root root 17 11月  9 09:39 object
drwxr-xr-x. 2 root root  6 3月  26 2015 rh
[root@localhost opt]# rm a.txt
rm: 是否删除普通空文件 "a.txt"? y
[root@localhost opt]# ll
总用量 0
-rw-r--r--. 1 root root  0 11月  9 09:35 Hello.java
-rw-r--r--. 1 root root  0 11月  9 09:35 index.html
drwxr-xr-x. 3 root root 17 11月  9 09:39 object
drwxr-xr-x. 2 root root  6 3月  26 2015 rh
[root@localhost opt]#
```

8. 拷贝

```

1  # 语法结构: copy 的缩写
2  cp [-adfilprsu] 源文档(source) 目标路径(destination)
3
4  # 参数含义:
5  -a: 相当於 -pdr 的意思, 至於 pdr 请参考下列说明: (常用)
6  -p: 连同文件的属性一起复制过去, 而非使用默认属性(备份常用);
7  -d: 若来源档为连结档的属性(link file), 则复制连结档属性而非文件本身;
8  -r: 递归持续复制, 用于目录的复制行为: (常用)
9  -f: 为强制(force)的意思, 不提示, 直接覆盖文件;
10 -i: 若目标档(destination)已经存在时, 在覆盖时会先询问动作的进行(常用)
11 -l: 进行硬式连结(hard link)的连结档创建, 而非复制文件本身。
12 -s: 复制成为符号连结档 (symbolic link), 亦即『捷径』文件;
13 -u: 若 destination 比 source 旧才升级 destination !

```

```

[root@localhost riu]# ll
总用量 0
-rw-r--r--. 1 root root 0 4月 13 16:29 a.txt
drwxr-xr-x. 2 root root 6 4月 13 16:30 test
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 公共
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 模板
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 视频
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 图片
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 文档
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 下载
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 音乐
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 桌面
[root@localhost riu]# cp a.txt test
[root@localhost riu]# cd test
[root@localhost test]# ll
总用量 0
-rw-r--r--. 1 root root 0 4月 13 16:30 a.txt
[root@localhost test]# cd ..
[root@localhost riu]# cp -i a.txt test
cp: 是否覆盖"test/a.txt"? y
[root@localhost riu]# cp -i a.txt test
cp: 是否覆盖"test/a.txt"? n
[root@localhost riu]#

```

9. 移动

```

1  # 语法结构: move 的缩写
2  mv [-fui] 源文档/目录(source) 目标路径(destination)
3
4  # 参数含义:
5  -f : force 强制的意思, 如果目标文件已经存在, 不会询问而直接覆盖;
6  -i : 若目标文件 (destination) 已经存在时, 就会询问是否覆盖!
7  -u : 若目标文件已经存在, 且 source 比较新, 才会升级 (update)
8
9  # 移动到当前文件夹可以修改文件名

```

```

[root@localhost riu]# ll
总用量 0
-rw-r--r--. 1 root root 0 4月 13 16:36 a.txt
drwxr-xr-x. 2 root root 6 4月 13 16:36 test
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 公共
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 模板
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 视频
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 图片
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 文档
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 下载
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 音乐
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 桌面
[root@localhost riu]# mv a.txt test
[root@localhost riu]# ll
总用量 0
drwxr-xr-x. 2 root root 19 4月 13 16:36 test
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 公共
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 模板
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 视频
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 图片
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 文档
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 下载
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 音乐
drwxr-xr-x. 2 riu riu 6 4月 10 10:32 桌面
[root@localhost riu]# cd test/
[root@localhost test]# ll
总用量 0
-rw-r--r--. 1 root root 0 4月 13 16:36 a.txt
[root@localhost test]#

```

10. Linux 短语

为什么命令后面添加了 `-f` 依旧有提示信息呢？那是因为 Linux 对文件删除的一种保护机制，避免误删除文件。所以会给一些命令自动绑定 `-i` 显示操作提示。绑定是通过 `alias` 命令实现，给一串命令顶一个短语，调用短语即可执行对应的命令。

如果不希望有这样的保护机制，可以使用 `unalias` 命令去掉短语绑定，例如：`unalias cp`。

除了内置的一些短语，还可以添加自定义的短语，项目中会将常用命令绑定短语，以提高工作效率，减少命令的错误率。

```
1 alias cp='cp -i'
2 alias egrep='egrep --color=auto'
3 alias fgrep='fgrep --color=auto'
4 alias grep='grep --color=auto'
5 alias l.='ls -d .* --color=auto'
6 alias ll='ls -l --color=auto'
7 alias ls='ls --color=auto'
8 alias mv='mv -i'
9 alias rm='rm -i'
10 alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
```

(二) 权限属性操作

1. 查看权限属性

Linux 系统是一种典型的多用户系统，不同的用户处于不同的角色，拥有不同的权限。为了保护系统的安全性，Linux 系统对不同的用户访问同一文件（包括目录文件）的权限做了不同的规定。

Linux 中文件属性共由10位组成；在Linux中我们可以使用 `ll` 或者 `ls -l` 命令来显示一个文件的属性以及文件所属的用户和组。

```
[root@localhost ~]# ll
总用量 28
lrwxrwxrwx. 1 root root 7 4月 10 10:04 bin -> usr/bin
dr-xr-xr-x. 5 root root 4096 4月 10 10:24 boot
drwxr-xr-x. 20 root root 3320 4月 13 14:02 dev
drwxr-xr-x. 143 root root 8192 4月 13 14:02 etc
drwxr-xr-x. 3 root root 17 4月 10 10:32 home
lrwxrwxrwx. 1 root root 7 4月 10 10:04 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 4月 10 10:04 lib64 -> usr/lib64
drwxr-xr-x. 2 root root 6 4月 11 2018 media
drwxr-xr-x. 2 root root 6 4月 11 2018 mnt
drwxr-xr-x. 3 root root 16 4月 10 10:12 opt
dr-xr-xr-x. 178 root root 0 4月 13 14:02 proc
dr-xr-x---. 15 root root 4096 4月 13 15:55 root
drwxr-xr-x. 41 root root 1240 4月 13 14:02 run
lrwxrwxrwx. 1 root root 8 4月 10 10:04 sbin -> usr/sbin
drwxr-xr-x. 2 root root 6 4月 11 2018 srv
dr-xr-xr-x. 13 root root 0 4月 13 14:02 sys
drwxrwxrwt. 30 root root 4096 4月 13 16:49 tmp
drwxr-xr-x. 13 root root 155 4月 10 10:04 usr
drwxr-xr-x. 21 root root 4096 4月 10 10:30 var
[root@localhost ~]#
```

文件类型：

第1位字符代表当前文件的类型：

- 当为[**d**]，Directory的缩写，是目录
- 当为[-]，是文件；
- 若是[**l**]，Link，表示为链接文档 (link file)；
- 若是[**b**]，表示为装置文件里面的可供储存的接口设备 (可随机存取装置)；
- 若是[**c**]，表示为装置文件里面的串行端口设备，例如键盘、鼠标 (一次性读取装置)。

文件权限：

剩余9位上字符代表这个文件的权限；可以分为三组：【当前用户拥有的权限】【当前用户所在组拥有的权限】【其他用户拥有的权限】

- r: 读取，可以用数字 4 表示
- w: 写，可以用数字 2 表示
- x: 执行，可以用数字 1 表示
- -: 没有权限，可以用数字 0 表示

文件 类型	属主 权限			属组 权限			其他用户 权限		
0	1	2	3	4	5	6	7	8	9
d	rwX			r-X			r-X		
目录 文件	读	写	执行	读	写	执行	读	写	执行

R的值等于 4 W值等于 2 X值等于 1
完全权限: 4+2+1=7 读写权限: 4+2=6

2. 操作权限

操作权限

Linux权限操作有两种设置方法，一种是数字，一种是符号。分别可以设置三种身份：owner（所有者）、group（所属组）、others（其他）；

三种身份分别有各自的权限：读、写、执行；

```
1 # 语法结构：其中权限分为三组 owner（所有者）、group（所属组）、others（其他）
2 chmod 权限 文件名
3 chmod 412 hello.java # owner: 可读      group: 可执行      others: 可写
4 chmod 777 hello.java # owner: 读写执行   group: 读写执行   others: 读写执行
5 chmod 000 hello.java # owner: 无权限     group: 无权限     others: 无权限
6
7 # 语法结构：
8 # u: 当前文件所属的用户
9 # g: 当前文件所属的用户组
10 # o: 其他
11 chmod u/g/o +/- r/w/x
12
13 chmod u+r
14 chmod o-x
```

操作所有者

更改文件属主，也可以同时更改文件属组

```
1 # 语法结构：
2 chown 属主名 文件名
3
4 chown riu a.txt
```


操作组

更改文件属组

```
1 # 语法结构:
2 chgrp 属组名 文件名
3
4 chgrp g1 a.txt
```

使用 `su` 命令可以切换用户，验证权限配置。

(三) 文件查看

常用命令：

```
1 cat # 由第一行开始显示文件内容
2 tac # 从最后一行开始显示，可以看出 tac 是 cat 的倒着写！
3 nl # 显示的时候，显示输出行号！
4 more # 一页一页的显示文件内容
5 less # 与 more 类似，但是比 more 更好的是，他可以往前翻页！
6 head # 只看头几行
7 tail # 只看尾部几行
```

正序查看文件内容：

```
1 # 语法结构:
2 cat [-AbEnTv] 文件名
3
4 # 参数含义:
5 -A : 相当於 -vET 的整合选项，可列出一些特殊字符而不是空白而已；
6 -v : 列出一些看不出来的特殊字符
7 -E : 将结尾的断行字节 $ 显示出来；
8 -T : 将 [tab] 按键以 ^I 显示出来；
9
10 -b : 列出行号，仅针对非空白行做行号显示，空白行不标行号！
11 -n : 列印出行号，连同空白行也会有行号，与 -b 的选项不同；
```

```
[root@localhost network-scripts]# cat ifcfg-ens33
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens33
UUID=0953eb94-0e4a-4cbd-b366-3369177790f1
DEVICE=ens33
ONBOOT=yes
DNS1=114.114.114.114
DNS2=8.8.8.8
IPADDR=192.168.23.123
NETMASK=255.255.255.0
GATEWAY=192.168.23.2
[root@localhost network-scripts]#
```

带参数，显示行号。

```
[root@localhost network-scripts]# cat ifcfg-ens33 -n
1 TYPE=Ethernet
2 PROXY_METHOD=none
3 BROWSER_ONLY=no
4 BOOTPROTO=static
5 DEFROUTE=yes
6 IPV4_FAILURE_FATAL=no
7 IPV6INIT=yes
8 IPV6_AUTOCONF=yes
9 IPV6_DEFROUTE=yes
10 IPV6_FAILURE_FATAL=no
11 IPV6_ADDR_GEN_MODE=stable-privacy
12 NAME=ens33
13 UUID=0953eb94-0e4a-4cbd-b366-3369177790f1
14 DEVICE=ens33
15 ONBOOT=yes
16 DNS1=114.114.114.114
17 DNS2=8.8.8.8
18 IPADDR=192.168.23.123
19 NETMASK=255.255.255.0
20 GATEWAY=192.168.23.2
[root@localhost network-scripts]#
```

倒序查看文件内容：

- 1 # 语法结构
- 2 tac 文件名

```
[root@localhost network-scripts]# tac ifcfg-ens33
GATEWAY=192.168.23.2
NETMASK=255.255.255.0
IPADDR=192.168.23.123
DNS2=8.8.8.8
DNS1=114.114.114.114
ONBOOT=yes
DEVICE=ens33
UUID=0953eb94-0e4a-4cbd-b366-3369177790f1
NAME=ens33
IPV6_ADDR_GEN_MODE=stable-privacy
IPV6_FAILURE_FATAL=no
IPV6_DEFROUTE=yes
IPV6_AUTOCONF=yes
IPV6INIT=yes
IPV4_FAILURE_FATAL=no
DEFROUTE=yes
BOOTPROTO=static
BROWSER_ONLY=no
PROXY_METHOD=none
TYPE=Ethernet
[root@localhost network-scripts]#
```

- 1 # 语法结构
- 2 nl [-bnw] 文件
- 3
- 4 # 参数含义：
- 5 **-b** ：指定行号指定的方式，主要有两种：**-b a** ：表示不论是否为空行，也同样列出行号(类似 **cat -n**)；**-b t** ：如果有空行，空的那一行不要列出行号(默认值)；
- 6 **-n** ：列出行号表示的方法，主要有三种：**-n ln** ：行号在荧幕的最左方显示；**-n rn** ：行号在自己栏位的最右方显示，且不加 **0** ；**-n rz** ：行号在自己栏位的最右方显示，且加 **0** ；
- 7 **-w** ：行号栏位的占用的位数。

```
[root@localhost network-scripts]# nl ifcfg-ens33
1 TYPE=Ethernet
2 PROXY_METHOD=none
3 BROWSER_ONLY=no
4 BOOTPROTO=static
5 DEFROUTE=yes
6 IPV4_FAILURE_FATAL=no
7 IPV6INIT=yes
8 IPV6_AUTOCONF=yes
9 IPV6_DEFROUTE=yes
10 IPV6_FAILURE_FATAL=no
11 IPV6_ADDR_GEN_MODE=stable-privacy
12 NAME=ens33
13 UUID=0953eb94-0e4a-4cbd-b366-3369177790f1
14 DEVICE=ens33
15 ONBOOT=yes
16 DNS1=114.114.114.114
17 DNS2=8.8.8.8
18 IPADDR=192.168.23.123
19 NETMASK=255.255.255.0
20 GATEWAY=192.168.23.2
[root@localhost network-scripts]#
```

(四) 文件编辑

1. vi/vim编辑器

Vim是从vi发展出来的一个文本编辑器。代码补完、编译及错误跳转等方便编程的功能特别丰富，在程序员中被广泛使用。

Vim分为三种模式：**命令模式 (Command mode)**，**输入模式 (Insert mode)** 和**底线命令模式 (Last line mode)**

命令模式：

刚启动vi/vim时即为命令模式，此时用户敲击键盘的内容不会被记录在文件中，而是被识别为命令。

- **i/a/o** 切换到**输入模式**，可以输入字符。
- **:** 切换到底线命令模式，以在最底一行输入命令。冒号一定是英文的
- **x** 删除当前光标所在处的字符。
- **/** 查找关键字，从上往下找
- **?** 查找关键字，从下往上找
- **G** 回到文件最后一行
- **gg** 回到文件第一行

输入模式：

用户敲击“i/a/o”时进入编辑模式

- **字符按键以及Shift组合**，输入字符
- **ENTER**，回车键，换行
- **BACK SPACE**，退格键，删除光标前一个字符
- **DEL**，删除键，删除光标后一个字符
- **方向键**，在文本中移动光标
- **HOME/END**，移动光标到行首/行尾
- **Page Up/Page Down**，上/下翻页
- **Insert**，切换光标为输入/替换模式，光标将变成竖线/下划线
- **ESC**，退出输入模式，切换到命令模式

底线命令模式：

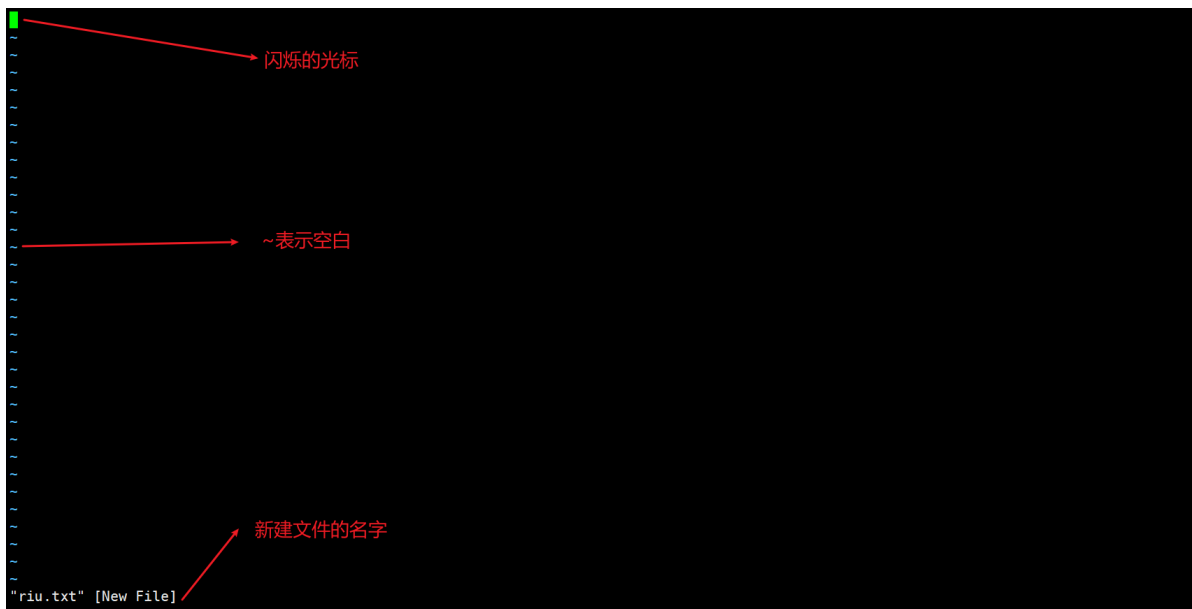
在命令模式下按下【: (英文冒号)】就进入了底线命令模式。按ESC可以退出底线命令模式。

- **【q】** 退出
- **【q!】** 强制退出文件
- **【w】** 保存文件
- **【wq】** 保存并退出

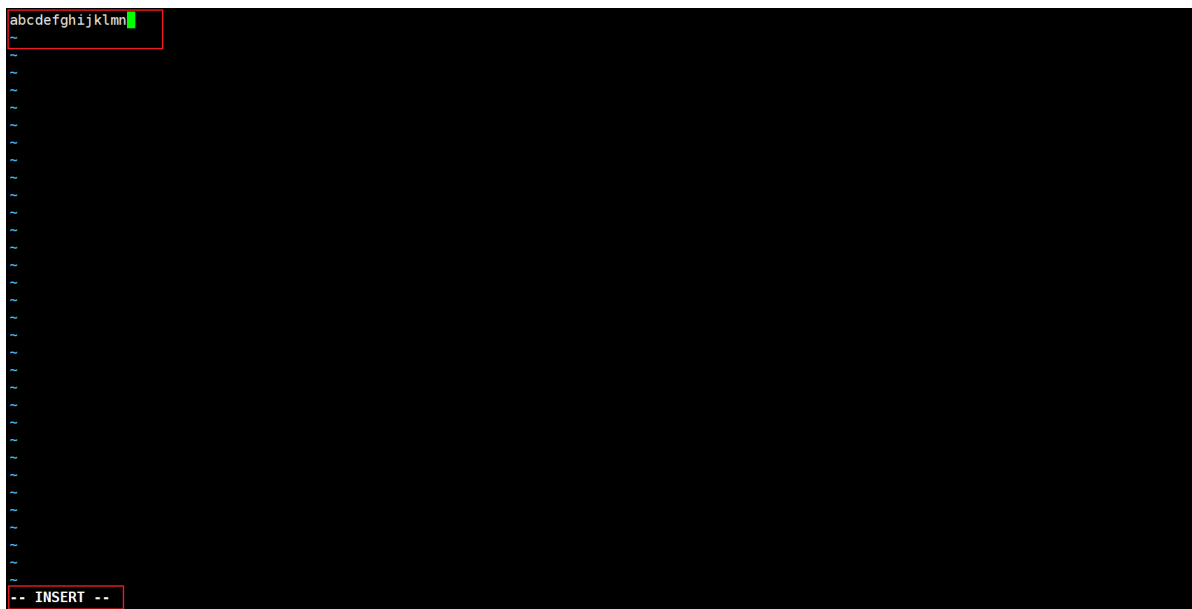
2. 文件编辑

```
1 vi riu.txt
2
3 1、刚打开文件处于【命令模式】
4 2、按i/a/o进入【编辑模式】
5 3、按ESC退出【编辑模式】，进入【命令模式】
6 4、按:【底线命令模式】，输入w或者q或者wq完成文件的编辑
```

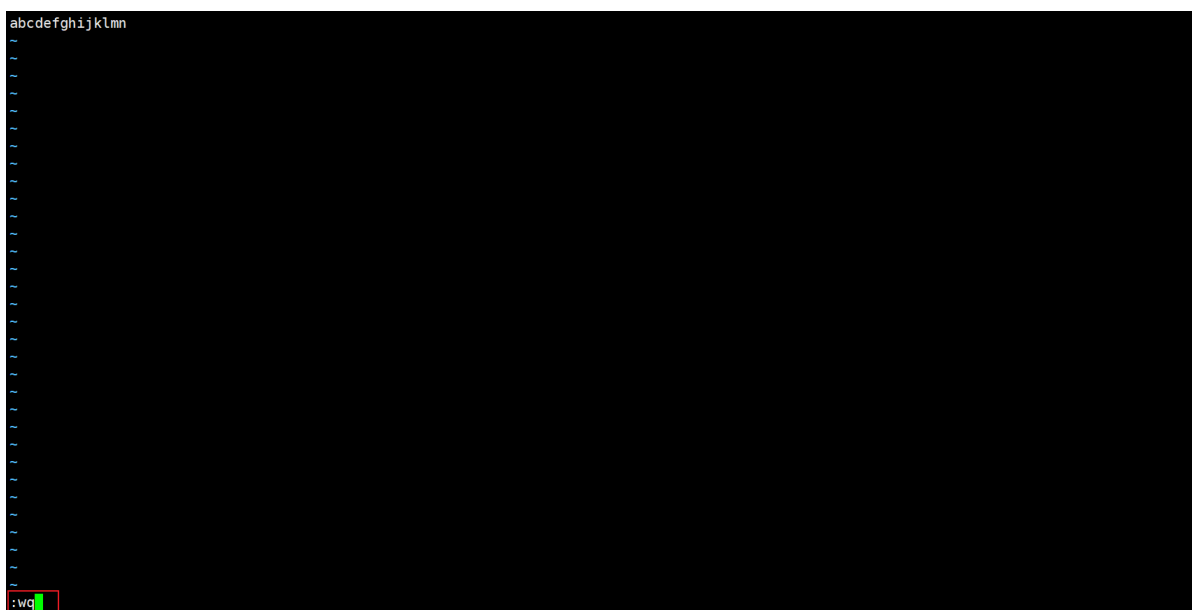
文件初始化的状态 (命令模式)



敲击键盘 `i`、`o`、`a` 进入编辑模式，此时可以输入内容



按ESC，退出编辑模式，进入命令模式。此时可以输入 `:wq`，回车，保存并退出文件



(五) 用户账号

Linux系统是一个多用户多任务的分时操作系统，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（root）申请一个账号，然后以这个账号的身份进入系统。

用户的账号一方面可以帮助系统管理员对使用系统的用户进行跟踪，并控制他们对系统资源的访问；另一方面也可以帮助用户组织文件，并为用户提供安全性保护。

每个用户账号都拥有一个唯一的用户名和各自的口令。

用户在登录时键入正确的用户名和口令后，就能够进入系统和自己的主目录。

创建用户

```
1  # 语法结构
2  useradd 参数 用户名
3
4  # 参数含义
5  -c comment 指定一段注释性描述。
6  -d 目录 指定用户主目录，如果此目录不存在，则同时使用-m选项，可以创建主目录。
7  -g 用户组 指定用户所属的用户组。
8  -G 用户组, 用户组 指定用户所属的附加组。
9  -m 使用者目录如不存在则自动建立。
10 -s shell文件 指定用户的登录Shell。
11 -u 用户号 指定用户的用户号，如果同时有-o选项，则可以重复使用其他用户的标识号。
```

```
[root@localhost home]# ll
总用量 4
drwx----- 15 riu riu 4096 4月 27 16:58 riu
[root@localhost home]# useradd -m user1
[root@localhost home]# ll
总用量 4
drwx----- 15 riu riu 4096 4月 27 16:58 riu
drwx-----  3 user1 user1 78 4月 27 17:07 user1
[root@localhost home]#
```

密码管理

```
1  # 语法结构
2  passwd 参数 用户名
3
4  # 参数含义
5  -l 锁定(lock)口令，即禁用账号。
6  -u 口令解锁(unlock)。
7
8  -d 使账号无口令。
9  -f 强制修改用户口令。
```

```
[root@localhost home]# passwd user1
更改用户 user1 的密码。
新的 密码:
无效的密码: 密码未通过字典检查 - 过于简化/系统化
重新输入新的 密码:
passwd: 所有的身份验证令牌已经成功更新。
[root@localhost home]# passwd -d user1
清除用户的密码 user1。
passwd: 操作成功
[root@localhost home]#
```

修改用户

```

1 # 语法结构
2 usermod 参数 用户名
3
4 # 参数含义
5 -c comment 指定一段注释性描述。
6 -d 目录 指定用户主目录，如果此目录不存在，则同时使用-m选项，可以创建主目录。
7 -g 用户组 指定用户所属的用户组。
8 -G 用户组，用户组 指定用户所属的附加组。
9 -m 使用者目录如不存在则自动建立。
10 -s shell文件 指定用户的登录Shell。
11 -u 用户号 指定用户的用户号，如果同时有-o选项，则可以重复使用其他用户的标识号。

```

切换用户

```

1 # 语法结构，
2 # 通过root用户切换其他用户时不需要输入密码。其他用户之间切换账号必须输入密码。
3 su 用户名
4
5 $: 普通用户
6 #: 超级用户，root用户

```

删除用户

```

1 # 语法结构
2 userdel 参数 用户名
3
4 # 参数含义
5 -f 强制删除用户，即使用户当前已登录
6 -r 删除用户的同时，删除与用户相关的所有文件

```

(六) 解压缩

```

1 # 语法结构
2 tar 参数 文件名
3
4 # 参数含义
5 -A或--catenate: 新增文件到已存在的备份文件；
6 -B: 设置区块大小；
7 -C或--create: 建立新的备份文件；
8 -C < 目录>: 这个选项用在解压缩，若要在特定目录解压缩，可以使用这个选项。
9 -d: 记录文件的差别；
10 -x或--extract或--get: 从备份文件中还原文件；
11 -t或--list: 列出备份文件的内容；
12 -z或--gzip或--ungzip: 通过gzip指令处理备份文件；
13 -Z或--compress或--uncompress: 通过compress指令处理备份文件；
14 -f< 备份文件>或--file=< 备份文件>: 指定备份文件；
15 -v或--verbose: 显示指令执行过程；
16 -r: 添加文件到已经压缩的文件；
17 -u: 添加改变了和现有的文件到已经存在的压缩文件；
18 -j: 支持bzip2解压文件；
19 -l: 文件系统边界设置；
20 -k: 保留原有文件不覆盖；
21 -m: 保留文件不被覆盖；

```

```
22 -w: 确认压缩文件的正确性;
23 -p或--same-permissions: 用原来的文件权限还原文件;
24 -P或--absolute-names: 文件名使用绝对名称, 不移除文件名前的“/”号;
25 -N < 日期格式> 或 --newer=< 日期时间>; 只将较指定日期更新的文件保存到备份文件里;
26 --exclude=< 范本样式>; 排除符合范本样式的文件。
```

```
1 # 打包文件
2 # 单个文件压缩打包
3 tar -czvf 包名.tar 文件
4 # 多个文件压缩打包
5 tar -czvf 包名.tar 文件1 文件2
6 # 单个目录压缩打包
7 tar -czvf 包名.tar 目录1
8 # 多个目录压缩打包
9 tar -czvf 包名.tar 目录1 目录2
```

```
1 # 解压文件
2 # 展示解压过程
3 tar -zxvf redis.tar.gz
4 # 不展示解压过程
5 tar -zxf redis.tar.gz
```

(七) 进程、端口

每一个应用的开启都需要一个进程, 每个进程都有自己的 ID 标识, 称为 PID; 系统中进程可以在前台执行也可在后台执行。对于服务相关的进程都要求在后台实时运行。

查看进程

```
1 # ps命令用于查看进程相关的所有信息
2 # 语法结构
3 ps 参数 | grep 过滤字符
4
5 # 参数含义
6 -a: 显示当前系统当前的进程
7 -u: 当前用户下
8 -x: 显示后台进程相关信息
9 |: 管道符, 用于过滤进程相关信息
10 grep: 查找进程中符合条件的进程信息
11
12 # 实例
13 # 查看进程中 redis 相关的进程
14 ps -aux | grep reids
```

```
1 # 以树的形式展示进程信息
2 pstree -pu
3 -p: 显示父进程
4 -u: 显示当前用户进程
```

结束进程

```
1 # 杀掉进程
2 kill -9 进程id
```

端口号

默认情况下，Linux 只开放了必要的端口号。例如：22、80。有需要的端口号，比如 Tomcat (8080)、MySQL (3306)、Oracle (1521)，需要在防火墙中放行。

五、软件安装

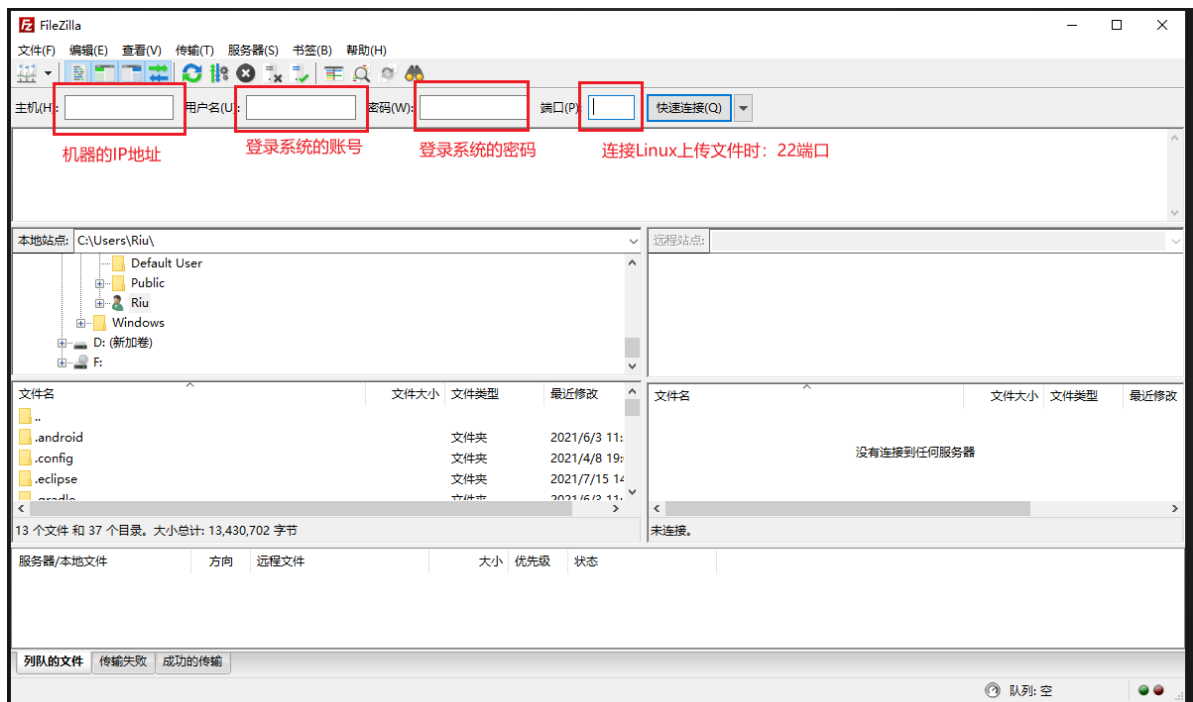
软件安装有三种方式：

离线安装：rpm、解压缩

在线安装：yum 在线安装

(零) 软件准备

安装 FileZilla 软件输入机器相关信息，连接之后将本地软件上传。



(一) Linux安装JDK

1、卸载Linux默认的JDK

(1) 查看默认JDK，如图

```
1 # -q: 查询; -a: 查询所有已经安装的软件包; grep: 查找内容包含指定的范本样式的文件;
2 rpm -qa | grep java
```

```
[root@localhost ~]# rpm -qa | grep java
java-1.7.0-openjdk-1.7.0.191-2.6.15.5.el7.x86_64
python-javapackages-3.4.1-11.el7.noarch
java-1.7.0-openjdk-headless-1.7.0.191-2.6.15.5.el7.x86_64
java-1.8.0-openjdk-headless-1.8.0.181-7.b13.el7.x86_64
tzdata-java-2018e-3.el7.noarch
javapackages-tools-3.4.1-11.el7.noarch
java-1.8.0-openjdk-1.8.0.181-7.b13.el7.x86_64
```

(2) 卸载JDK，参照以下命令


```

1 # -e 删除指定的套件
2 # --nodeps 不验证套件档的相互关联性
3 rpm -e --nodeps java-1.7.0-openjdk-1.7.0.191-2.6.15.5.el7.x86_64
4 rpm -e --nodeps java-1.7.0-openjdk-headless-1.7.0.191-2.6.15.5.el7.x86_64
5 rpm -e --nodeps java-1.8.0-openjdk-headless-1.8.0.181-7.b13.el7.x86_64
6 rpm -e --nodeps java-1.8.0-openjdk-1.8.0.181-7.b13.el7.x86_64

```

(3) 卸载完成之后再执行java命令会出现错误或者找不到目录或者命令

```

[root@localhost /]# rpm -e --nodeps java-1.7.0-openjdk-1.7.0.191-2.6.15.5.el7.x86_64
[root@localhost /]# rpm -e --nodeps java-1.7.0-openjdk-headless-1.7.0.191-2.6.15.5.el7.x86_64
[root@localhost /]# rpm -e --nodeps java-1.8.0-openjdk-headless-1.8.0.181-7.b13.el7.x86_64
[root@localhost /]# rpm -e --nodeps java-1.8.0-openjdk-1.8.0.181-7.b13.el7.x86_64
[root@localhost /]# javac
bash: javac: 未找到命令...
[root@localhost /]# java
-bash: /usr/bin/java: 没有那个文件或目录

```

2、安装自定义JDK

(1) 上传下载好的安装包 `jdk-8u60-linux-x64.tar.gz` 到Linux中，目录自定义，比如：`/opt/jdk`

```

[root@localhost /]# cd /opt
[root@localhost opt]# ll
总用量 189664
-rw-r--r--. 1 root root 191753373 5月 17 10:43 jdk-8u191-linux-x64.tar.gz
drwxrwxr-x. 7 root root 4096 5月 8 15:21 redis-6.2.3
-rw-r--r--. 1 root root 2456050 5月 8 14:29 redis-6.2.3.tar.gz
drwxr-xr-x. 2 root root 6 10月 31 2018 rh

```

(2) 解压：`tar -zxvf jdk-8u60-linux-x64.tar.gz`

```

[root@localhost opt]# ll
总用量 189664
drwxr-xr-x. 7 10 143 245 10月 6 2018 jdk1.8.0_191
-rw-r--r--. 1 root root 191753373 5月 17 10:43 jdk-8u191-linux-x64.tar.gz
drwxrwxr-x. 7 root root 4096 5月 8 15:21 redis-6.2.3
-rw-r--r--. 1 root root 2456050 5月 8 14:29 redis-6.2.3.tar.gz
drwxr-xr-x. 2 root root 6 10月 31 2018 rh
[root@localhost opt]#

```

(3) 配置环境变量

打开文件：`vim /etc/profile` 在文件的默认添加以下信息

```

1 # jdk config
2 # JDK的home目录，bin目录的上一级，每个人的环境目录不一样
3 export JAVA_HOME=/opt/jdk/jdk1.8.0_60
4 # 将JDK添加到$PATH环境变量中
5 export PATH=$JAVA_HOME/bin/:$PATH
6 export CLASSPATH=:$CLASSPATH:$JAVA_HOME/lib/

```

```

# jdk config
export JAVA_HOME=/opt/jdk1.8.0_191
export PATH=$JAVA_HOME/bin/:$PATH

```

81,0-1 底端

刷新环境变量：`source /etc/profile`

(4) 输入 `java -version` 测试

(5) 在 `~/.bashrc` 文件中配置环境变量，解决每次启动需要重新刷新环境变量才能使用java命令的问题

```
[root@localhost ~]# ls -la
总用量 72
dr-xr-x---. 16 root root 4096 12月 25 23:35 .
dr-xr-xr-x. 17 root root 240 6月 30 23:46 ..
-rw-----. 1 root root 1583 6月 30 21:00 anaconda-ks.cfg
-rw-----. 1 root root 11007 12月 25 23:37 .bash_history
-rw-r--r--. 1 root root 18 12月 29 2013 .bash_logout
-rw-r--r--. 1 root root 176 12月 29 2013 .bash_profile
-rw-r--r--. 1 root root 197 12月 25 23:35 .bashrc
drwx-----. 15 root root 4096 7月 27 21:15 .cache
drwxr-xr-x. 16 root root 4096 6月 30 22:04 .config
-rw-r--r--. 1 root root 100 12月 29 2013 .cshrc
drwx-----. 3 root root 25 6月 30 21:01 .dbus
-rw-----. 1 root root 16 6月 30 21:14 .esd_auth
-rw-----. 1 root root 1244 12月 25 23:00 .ICEauthority
-rw-r--r--. 1 root root 1631 6月 30 21:01 initial-setup-ks.cfg
drwx-----. 3 root root 19 6月 30 21:14 .local
drwx-----. 5 root root 66 7月 27 21:16 .mozilla
drwxr-xr-x. 2 root root 40 12月 25 23:19 .oracle_jre_usage
-rw-----. 1 root root 774 7月 1 00:41 .rediscli_history
-rw-r--r--. 1 root root 129 12月 29 2013 .tcshrc
-rw-----. 1 root root 6112 12月 25 23:30 .viminfo
drwxr-xr-x. 2 root root 6 6月 30 21:14 公共
drwxr-xr-x. 2 root root 6 6月 30 21:14 模板
drwxr-xr-x. 2 root root 6 6月 30 21:14 视频
drwxr-xr-x. 2 root root 6 6月 30 21:14 图片
drwxr-xr-x. 2 root root 6 6月 30 21:14 文档
drwxr-xr-x. 2 root root 6 6月 30 21:14 下载
drwxr-xr-x. 2 root root 6 6月 30 21:14 音乐
drwxr-xr-x. 2 root root 6 6月 30 21:14 桌面
```

```
选择root@localhost:~
# . bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

source /etc/profile
```

(二) Linux安装Tomcat

- 1、在 `/opt` 目录下创建tomcat9文件夹，并把下载好的安装包（apache-tomcat-9.0.30.tar.gz）上传到该目录下
- 2、解压安装包 `tar -zxvf apache-tomcat-9.0.30.tar.gz`，解压后会有一个同名的文件夹

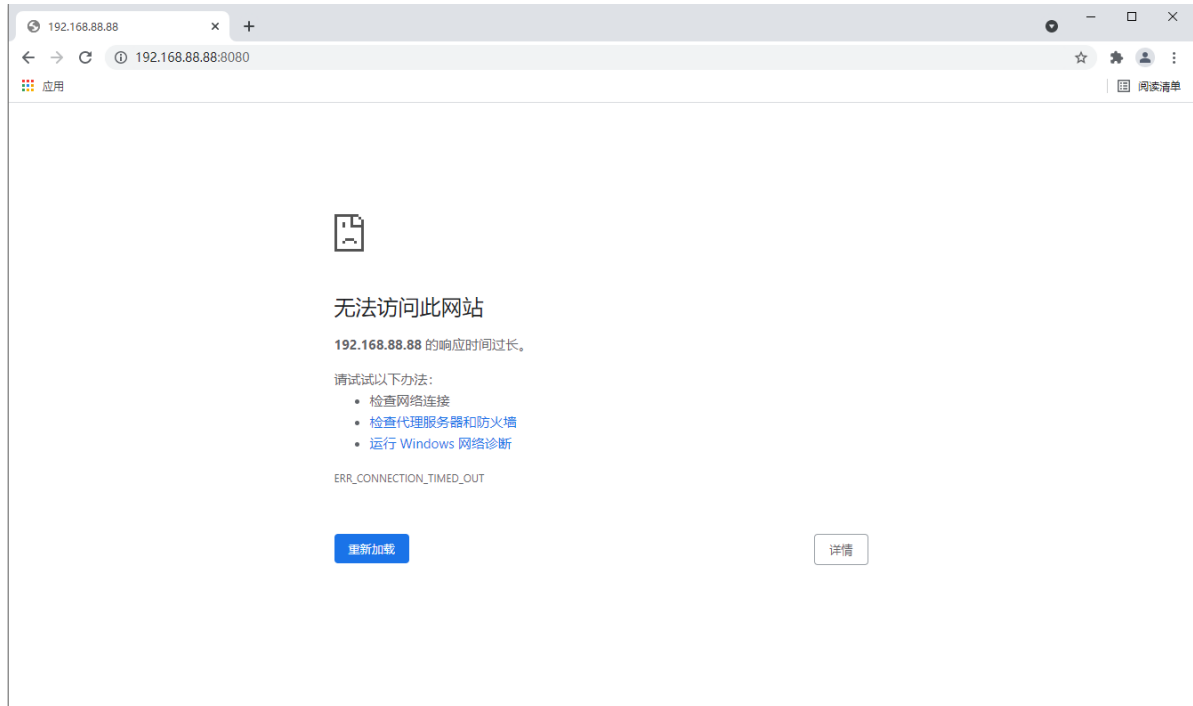
```
drwxr-xr-x. 9 root root      220 12月 25 20:49 apache-tomcat-9.0.30
-rwxr--r--. 1 peter peter 11026056 12月 25 20:00 apache-tomcat-9.0.30.tar.gz
```

3、进入解压后的目录 `cd apache-tomcat-9.0.30/bin`

4、输入命令 `./startup.sh` 启动tomcat

```
root@localhost bin]# ./startup.sh
Using CATALINA_BASE:   /opt/tomcat9/apache-tomcat-9.0.30
Using CATALINA_HOME:   /opt/tomcat9/apache-tomcat-9.0.30
Using CATALINA_TMPDIR: /opt/tomcat9/apache-tomcat-9.0.30/temp
Using JRE_HOME:        /usr
Using CLASSPATH:        /opt/tomcat9/apache-tomcat-9.0.30/bin/bootstrap.jar:/opt/tomcat9/apache-tomcat-9.0.30/bin/tomcat-juli.jar
Tomcat started.
root@localhost bin]#
```

5、访问 Tomcat 服务器 `http://ip:8080`，能看到 Tomcat 页面表示启动成功。如果启动成功，但是访问不到，有可能是linux防火墙没有对8080端口放行。



6、设置linux防火墙放行8080端口

默认情况下，Linux 只放行了 80 端口。其余的端口在需要的情况下自行放行。这样也是保证了 Linux 服务器的安全性。

有些情况下为了保证 Linux 的安全，不会放行其他端口，可以通过 Nginx 动态代理的方式，将外网的路径转发至内网路径。

例如：

项目1：8080:/obj1

项目2：8084:/obj2

`http:// 192.168.88.88/obj1 -> http://localhost:8080/obj1`

`http:// 192.168.88.88/obj2 -> http://localhost:8084/obj2`

- 1 -- 开启8080端口命令，注意命令中间的空格
- 2 `firewall-cmd --zone=public --add-port=8080/tcp --permanent`
- 3 -- 重新加载防火墙
- 4 `firewall-cmd --reload`

```

1  -- 关闭8080端口命令，注意命令中间的空格
2  firewall-cmd --zone=public --remove-port=8080/tcp --permanent
3  -- 重新加载防火墙
4  firewall-cmd --reload

```

7、发布项目：把 war 包上传到 tomcat 的 `webapps` 目录中，启动 Tomcat，会自动解压 war 包发布

(三) Linux安装MySQL8

1、查看是否已经安装MySQL

```

[root@localhost ~]# rpm -aq | grep mysql
[root@localhost ~]#

```

2、将MySQL安装包上传至Linux

```

[root@localhost mysql]# ll
总用量 775380
-rw-r--r--. 1 root root 793989120 5月 18 22:24 mysql-8.0.25-1.el7.x86_64.rpm-bundle.tar
[root@localhost mysql]#

```

3、卸载CenterOS自带Mariadb数据库，会和MySQL冲突

```

1  # 查找
2  rpm -qa | grep mariadb
3  # 卸载
4  rpm -e --nodeps mariadb-libs-5.5.60-1.el7_5.x86_64

```

```

[root@localhost ~]# rpm -aq | grep mariadb
mariadb-libs-5.5.60-1.el7_5.x86_64
[root@localhost ~]# rpm -e --nodeps mariadb-libs-5.5.60-1.el7_5.x86_64
[root@localhost ~]# rpm -aq | grep mariadb
[root@localhost ~]#

```

4、解压MySQL安装包：`tar -xvf mysql-8.0.25-1.el7.x86_64.rpm-bundle.tar`

```

[root@localhost mysql]# ll
总用量 775380
-rw-r--r--. 1 root root 793989120 5月 18 22:24 mysql-8.0.25-1.el7.x86_64.rpm-bundle.tar
[root@localhost mysql]# tar -xvf mysql-8.0.25-1.el7.x86_64.rpm-bundle.tar
mysql-community-client-8.0.25-1.el7.x86_64.rpm
mysql-community-client-plugins-8.0.25-1.el7.x86_64.rpm
mysql-community-common-8.0.25-1.el7.x86_64.rpm
mysql-community-devel-8.0.25-1.el7.x86_64.rpm
mysql-community-embedded-compat-8.0.25-1.el7.x86_64.rpm
mysql-community-libs-8.0.25-1.el7.x86_64.rpm
mysql-community-libs-compat-8.0.25-1.el7.x86_64.rpm
mysql-community-server-8.0.25-1.el7.x86_64.rpm
mysql-community-test-8.0.25-1.el7.x86_64.rpm
[root@localhost mysql]# ll
总用量 1550772
-rw-r--r--. 1 root root 793989120 5月 18 22:24 mysql-8.0.25-1.el7.x86_64.rpm-bundle.tar
-rw-r--r--. 1 7155 31415 47810444 4月 26 15:36 mysql-community-client-8.0.25-1.el7.x86_64.rpm
-rw-r--r--. 1 7155 31415 193616 4月 26 15:36 mysql-community-client-plugins-8.0.25-1.el7.x86_64.rpm
-rw-r--r--. 1 7155 31415 628904 4月 26 15:36 mysql-community-common-8.0.25-1.el7.x86_64.rpm
-rw-r--r--. 1 7155 31415 6800820 4月 26 15:36 mysql-community-devel-8.0.25-1.el7.x86_64.rpm
-rw-r--r--. 1 7155 31415 23586756 4月 26 15:37 mysql-community-embedded-compat-8.0.25-1.el7.x86_64.rpm
-rw-r--r--. 1 7155 31415 4240320 4月 26 15:37 mysql-community-libs-8.0.25-1.el7.x86_64.rpm
-rw-r--r--. 1 7155 31415 1262876 4月 26 15:37 mysql-community-libs-compat-8.0.25-1.el7.x86_64.rpm
-rw-r--r--. 1 7155 31415 448614076 4月 26 15:38 mysql-community-server-8.0.25-1.el7.x86_64.rpm
-rw-r--r--. 1 7155 31415 260841936 4月 26 15:40 mysql-community-test-8.0.25-1.el7.x86_64.rpm
[root@localhost mysql]#

```

5、通过 `rpm -ivh mysql-community-common-8.0.25-1.el7.x86_64.rpm --nodeps --force` 命令安装 common

```

[root@localhost mysql]# rpm -ivh mysql-community-common-8.0.25-1.el7.x86_64.rpm --nodeps --force
警告: mysql-community-common-8.0.25-1.el7.x86_64.rpm: 头V3 DSA/SHA1 Signature, 密钥 ID 5072elf5: NOKEY
准备中... ##### [100%]
正在升级/安装...
 1:mysql-community-common-8.0.25-1.el7##### [100%]
[root@localhost mysql]#

```

6、通过 `rpm -ivh mysql-community-libs-8.0.25-1.el7.x86_64.rpm --nodeps --force` 命令安装 libs

```

[root@localhost mysql]# rpm -ivh mysql-community-libs-8.0.25-1.el7.x86_64.rpm --nodeps --force
警告: mysql-community-libs-8.0.25-1.el7.x86_64.rpm: 头V3 DSA/SHA1 Signature, 密钥 ID 5072elf5: NOKEY
准备中... ##### [100%]
正在升级/安装...
 1:mysql-community-libs-8.0.25-1.el7##### [100%]
[root@localhost mysql]#

```

7、通过 rpm -ivh mysql-community-client-8.0.25-1.el7.x86_64.rpm --nodeps --force 命令安装 client

```
[root@localhost mysql]# rpm -ivh mysql-community-client-8.0.25-1.el7.x86_64.rpm --nodeps --force
警告: mysql-community-client-8.0.25-1.el7.x86_64.rpm: 头V3 DSA/SHA1 Signature, 密钥 ID 5072e1f5: NOKEY
准备中...##### [100%]
正在升级/安装...
  1:mysql-community-client-8.0.25-1.el7.x86_64.rpm##### [100%]
[root@localhost mysql]#
```

8、通过 rpm -ivh mysql-community-server-8.0.25-1.el7.x86_64.rpm --nodeps --force 命令安装 server

```
[root@localhost mysql]# rpm -ivh mysql-community-server-8.0.25-1.el7.x86_64.rpm --nodeps --force
警告: mysql-community-server-8.0.25-1.el7.x86_64.rpm: 头V3 DSA/SHA1 Signature, 密钥 ID 5072e1f5: NOKEY
准备中...##### [100%]
正在升级/安装...
  1:mysql-community-server-8.0.25-1.el7.x86_64.rpm##### [100%]
[root@localhost mysql]#
```

9、查看已经安装的MySQL rpm -qa | grep mysql

```
[root@localhost mysql]# rpm -qa | grep mysql
mysql-community-server-8.0.25-1.el7.x86_64
mysql-community-common-8.0.25-1.el7.x86_64
mysql-community-libs-8.0.25-1.el7.x86_64
mysql-community-client-8.0.25-1.el7.x86_64
[root@localhost mysql]#
```

10、设置mysql8不区分表名大小写，打开 vim /etc/my.cnf，在末行添加： lower_case_table_names=1

```
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock

log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid

lower_case_table_names=1
```

11、执行一下命令初始化MySQL和基本配置

1	mysqld --initialize	# 初始化MySQL
2	chown mysql:mysql /var/lib/mysql -R	# 将mysql目录划分给mysql用户和mysql组
3	systemctl start mysqld.service	# 启动MySQL服务
4	systemctl enable mysqld.service	# 设置MySQL服务开机启动

```
[root@localhost mysql]# mysqld --initialize
[root@localhost mysql]# chown mysql:mysql /var/lib/mysql -R
[root@localhost mysql]# systemctl start mysqld.service
[root@localhost mysql]# systemctl enable mysqld.service
```

12、查看初始密码 cat /var/log/mysql.log | grep password

```
[root@localhost mysql]# cat /var/log/mysqld.log | grep password
2021-05-18T15:15:14.240385Z 6 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: x,q-sy8+g&gG
[root@localhost mysql]#
```

如图所示，密码是：x,q-sy8+g&gG

13、使用初始密码登录数据库：mysql -user root -p，Linux下输入密码是不显示的

```
[root@localhost mysql]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.25

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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>
```

14、修改初始密码：ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'root';

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'root';
Query OK, 0 rows affected (0.04 sec)

mysql>
```

15、输入 `exit` 退出，再次使用新密码登录

```
mysql> exit
Bye
[root@localhost mysql]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.25 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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>
```

16、配置MySQL可以远程访问

```
1  -- 创建root账号，%表示可以远程连接
2  create user 'root'@'%' identified with mysql_native_password by 'root';
3  -- 给'root'@'%'用户添加权限
4  grant all privileges on *.* to 'root'@'%' with grant option;
5  -- 刷新权限
6  flush privileges;
```

```
mysql> create user 'root'@'%' identified with mysql_native_password by 'root';
Query OK, 0 rows affected (0.04 sec)

mysql> grant all privileges on *.* to 'root'@'%' with grant option;
Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.03 sec)

mysql>
```

17、配置防火墙放行

```
1  # 开启8080端口命令，注意命令中间的空格 firewall-cmd --zone=public --add-
    port=3306/tcp --permanent# 重新加载防火墙 firewall-cmd --reload
```

```
[root@localhost mysql]# firewall-cmd --zone=public --add-port=3306/tcp --permanent
success
[root@localhost mysql]# firewall-cmd --reload
success
[root@localhost mysql]#
```