

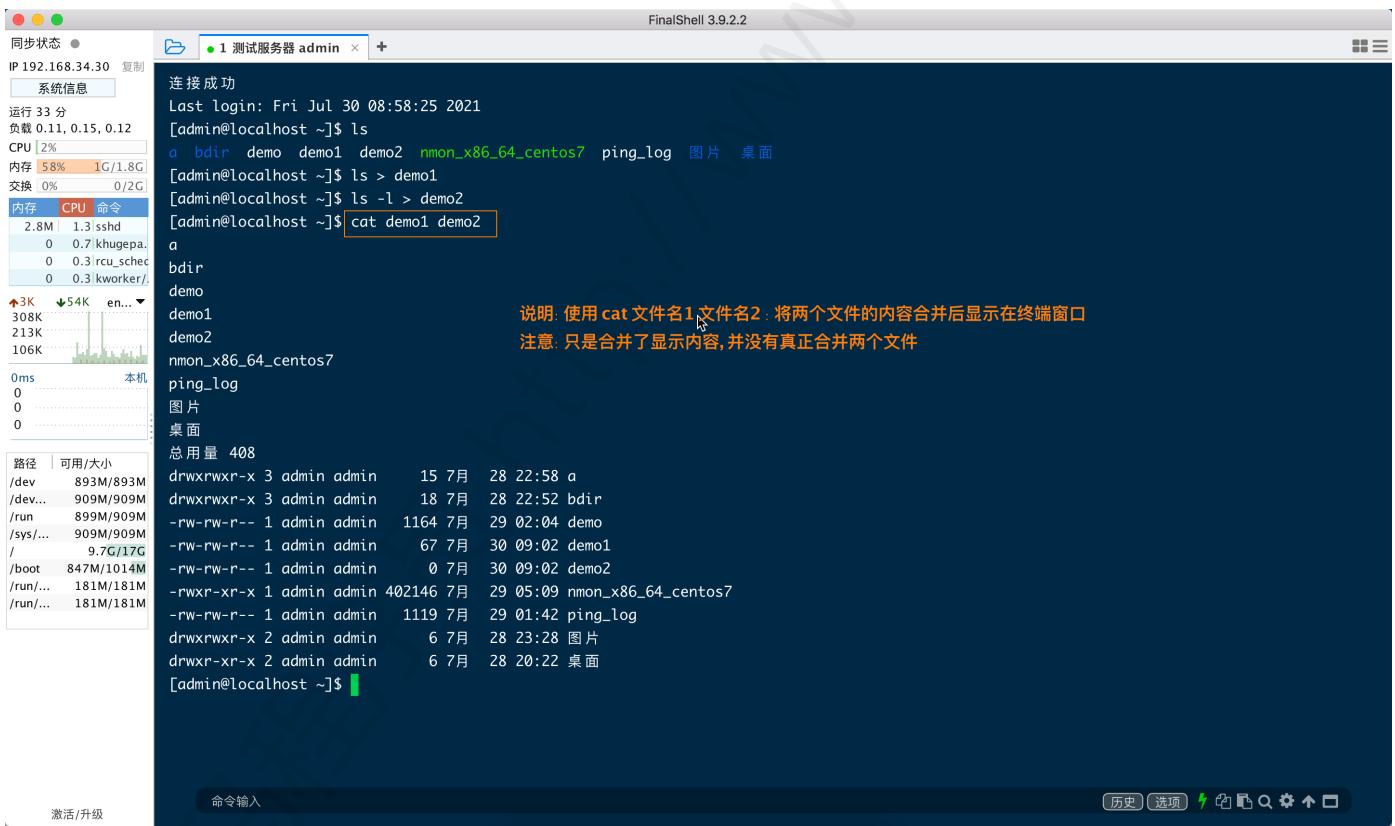
# Linux 命令 day03 随堂笔记

## 今日内容

- Linux 命令收尾
- 数据库的相关理论

## Linux 命令

### 补充: cat 合并查看多个文件内容



```
FinalShell 3.9.2.2
IP 192.168.34.30 复制
系统信息
运行 33 分
负载 0.11, 0.15, 0.12
CPU 2%
内存 58% 1G/1.8G
交换 0% 0/2G
内存 CPU 命令
2.8M 1.3 sshd
0 0.7 khugepaged
0 0.3 rCU_sche...
0 0.3 kworker...
a bdir demo demo1 demo2 nmon_x86_64_centos7 ping_log 图片 桌面
[admin@localhost ~]$ ls > demo1
[admin@localhost ~]$ ls -l > demo2
[admin@localhost ~]$ cat demo1 demo2
a
bdir
demo
demo1
demo2
nmon_x86_64_centos7
ping_log
图片
桌面
总用量 408
drwxrwxr-x 3 admin admin 15 7月 28 22:58 a
drwxrwxr-x 3 admin admin 18 7月 28 22:52 bdir
-rw-rw-r-- 1 admin admin 1164 7月 29 02:04 demo
-rw-rw-r-- 1 admin admin 67 7月 30 09:02 demo1
-rw-rw-r-- 1 admin admin 0 7月 30 09:02 demo2
-rwxr-xr-x 1 admin admin 402146 7月 29 05:09 nmon_x86_64_centos7
-rw-rw-r-- 1 admin admin 1119 7月 29 01:42 ping_log
drwxrwxr-x 2 admin admin 6 7月 28 23:28 图片
drwxr-xr-x 2 admin admin 6 7月 28 20:22 桌面
[admin@localhost ~]$
```

说明: 使用 cat 文件名1 文件名2 . 将两个文件的内容合并后显示在终端窗口  
注意: 只是合并了显示内容, 并没有真正合并两个文件

# 案例7: chmod 修改文件权限

## 文件权限的解读方法



```
[admin@localhost ~]$ ls
a bdir demo demo1 demo2 nmon_x86_64_cents7 ping_log 图片 桌面
[admin@localhost ~]$ ls -l
总用量 412
drwxrwxr-x 3 admin admin 15 7月 28 22:58 a
drwxrwxr-x 3 admin admin 18 7月 28 22:52 bdir
-rw-rw-r-- 1 admin admin 1164 7月 29 02:04 demo
-rw-rw-r-- 1 admin admin 67 7月 30 09:02 demo1
-rw-rw-r-- 1 admin admin 504 7月 30 09:02 demo2
-rwxr-xr-x 1 admin admin 402146 7月 29 05:09 nmon_x86_64_cents7
-rw-rw-r-- 1 admin admin 1119 7月 29 01:42 ping_log
drwxrwxr-x 2 admin admin 6 7月 28 23:28 图片
drwxr-xr-x 2 admin admin 6 7月 28 20:22 桌面
[admin@localhost ~]$
```

## 对照表

### 文件权限示例

### 目录权限示例

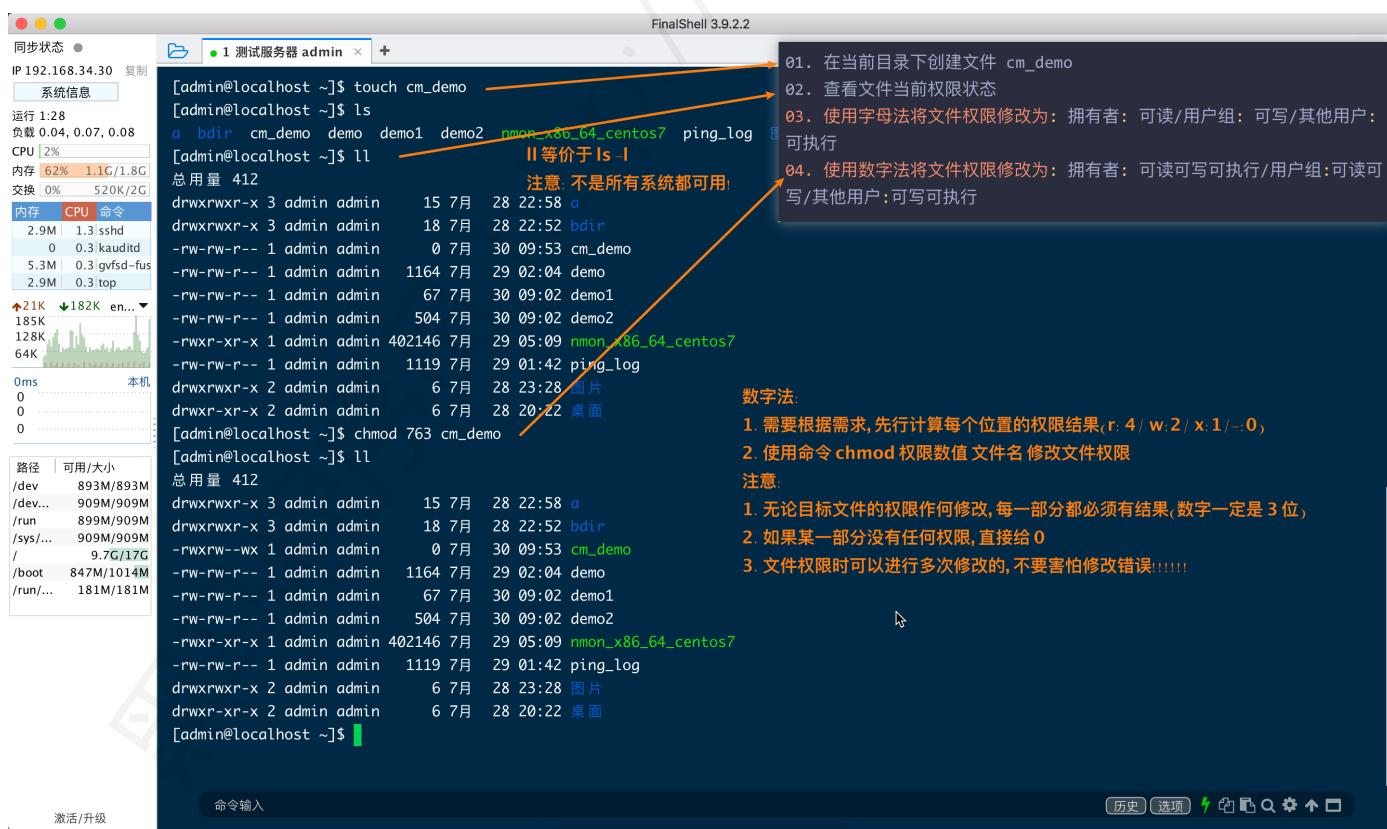
目录	拥有者权限			组权限			其他用户权限		
-	r	w	-	r	w	-	r	-	-
d	r	w	x	r	w	x	r	-	x

# 修改文件权限: 数字法[掌握]

## 权限对应数值

权限	英文	缩写	数字代号
读	read	r	4
写	write	w	2
执行	execute	x	1
无权限		-	0

## 案例实现



01. 在当前目录下创建文件 cm\_demo  
02. 查看文件当前权限状态  
03. 使用字母法将文件权限修改为：拥有者：可读/用户组：可写/其他用户：可执行  
注意：不是所有系统都可用  
04. 使用数字法将文件权限修改为：拥有者：可读可写可执行/用户组：可读可写/其他用户：可写可执行

数字法：  
1. 需要根据需求,先行计算每个位置的权限结果,r:4 / w:2 / x:1 / -:0,  
2. 使用命令 chmod 权限数值 文件名修改文件权限  
注意：  
1. 无论目标文件的权限作何修改,每一部分都必须有结果,数字一定是3位,  
2. 如果某一部分没有任何权限,直接给0  
3. 文件权限时可以进行多次修改的,不要害怕修改错误!!!!

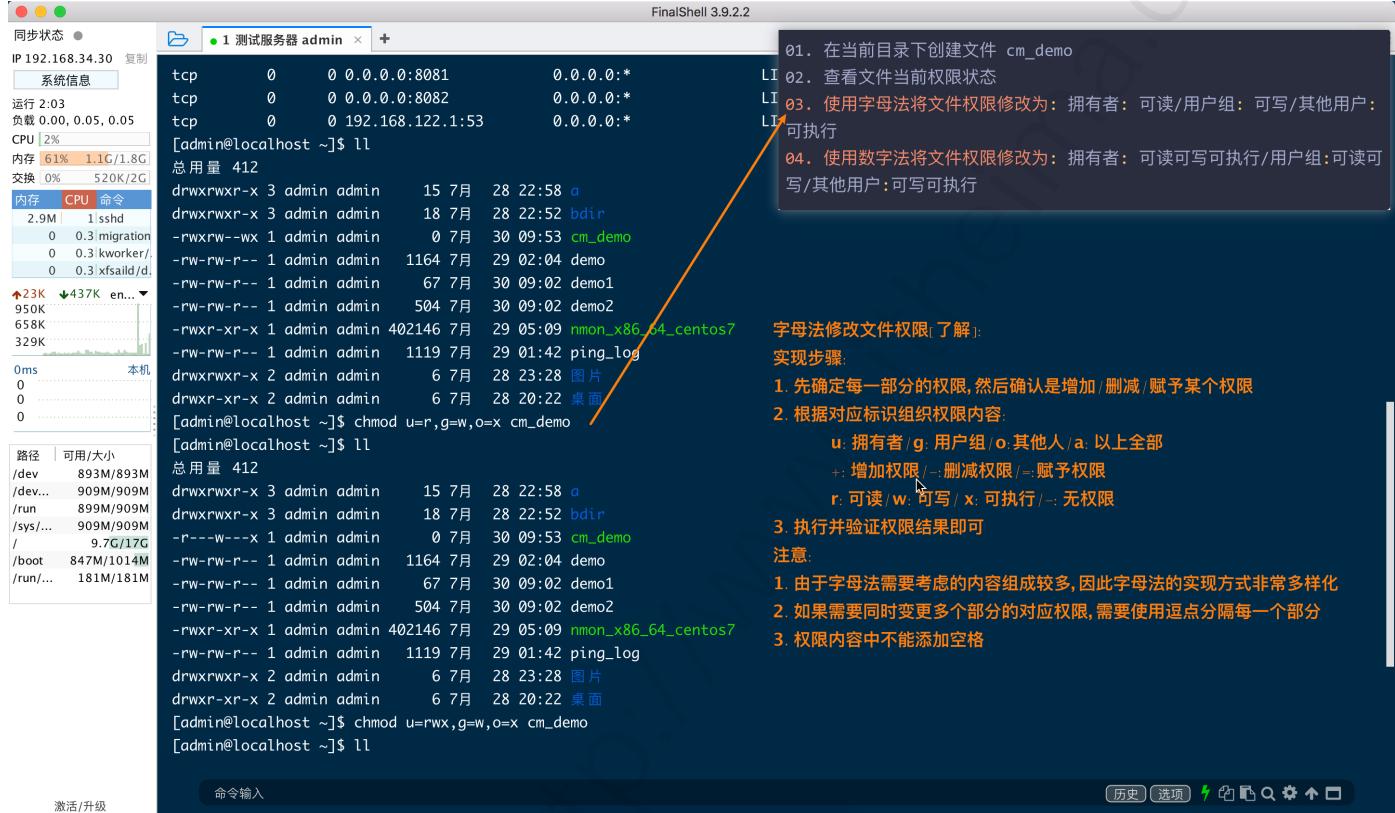
```
[admin@localhost ~]$ touch cm_demo
[admin@localhost ~]$ ls
a bdir cm_demo demo demo1 demo2 rmon_x86_64_centos7 ping_log
[admin@localhost ~]$ ll
II等价于ls -l
注意：不是所有系统都可用
总用量 412
drwxrwxr-x 3 admin admin 15 7月 28 22:58 a
drwxrwxr-x 3 admin admin 18 7月 28 22:52 bdir
-rw-rw-r-- 1 admin admin 0 7月 30 09:53 cm_demo
-rw-rw-r-- 1 admin admin 1164 7月 29 02:04 demo
-rw-rw-r-- 1 admin admin 67 7月 30 09:02 demo1
-rw-rw-r-- 1 admin admin 504 7月 30 09:02 demo2
-rwxr-xr-x 1 admin admin 402146 7月 29 05:09 rmon_x86_64_centos7
-rw-rw-r-- 1 admin admin 1119 7月 29 01:42 ping_log
drwxrwxr-x 2 admin admin 6 7月 28 23:28 图片
drwxr-xr-x 2 admin admin 6 7月 28 20:22 桌面
[admin@localhost ~]$ chmod 763 cm_demo
[admin@localhost ~]$ ll
总用量 412
drwxrwxr-x 3 admin admin 15 7月 28 22:58 a
drwxrwxr-x 3 admin admin 18 7月 28 22:52 bdir
-rwxrwxr--wx 1 admin admin 0 7月 30 09:53 cm_demo
-rw-rw-r-- 1 admin admin 1164 7月 29 02:04 demo
-rw-rw-r-- 1 admin admin 67 7月 30 09:02 demo1
-rw-rw-r-- 1 admin admin 504 7月 30 09:02 demo2
-rwxr-xr-x 1 admin admin 402146 7月 29 05:09 rmon_x86_64_centos7
-rw-rw-r-- 1 admin admin 1119 7月 29 01:42 ping_log
drwxrwxr-x 2 admin admin 6 7月 28 23:28 图片
drwxr-xr-x 2 admin admin 6 7月 28 20:22 桌面
[admin@localhost ~]$
```

# 修改文件权限: 字母法[了解]

组别: u: 拥有者/ g: 用户组/ o: 其他人/ a: 以上全部

修改: +: 增加权限/ -: 删减权限/ =: 赋予权限

权限: r: 可读/ w: 可写/ x: 可执行/ -: 无权限



## 文件权限修改注意事项

- 必须明确文件权限的含义(读懂)
- 使用数字法还是字母法全凭个人喜好
- 尽量注意不要随意赋予文件最高权限: 数字法(777)/字母法(rwxrwxrwx)
- 建议适当修改文件权限: 数字法(755)/字母法(rwxr-xr-x), 除拥有者外其余无可写权限, 以保证文件内容安全

## 案例9: find 查找文件

FinalShell 3.9.2.2

```
[admin@localhost ~]$ ls
[admin@localhost ~]$ mkdir adir bdir
[admin@localhost ~]$ ls
adir bdir
[admin@localhost ~]$ touch adir/f_demo
[admin@localhost ~]$ ls adir/
f_demo
[admin@localhost ~]$ cd bdir/
[admin@localhost bdir]$ pwd
/home/admin/bdir
[admin@localhost bdir]$ find /home/admin/ -name *mo
/home/admin/adir/f_demo
[admin@localhost bdir]$ find /home/admin/ -name '*mo'
/home/admin/adir/f_demo
[admin@localhost bdir]$ find /root -name '*mo'
find: '/root': 权限不够
[admin@localhost bdir]$
```

01. 在路径下创建 adir,bdir 两个文件夹  
02. 在 adir 目录下创建文件 f\_demo  
03. 切换路径到 bdir 目录下  
04. 在当前目录下从 **用户** 目录中查找 f\_demo 文件

命令格式: find 查找路径信息 -name 目标文件名  
注意:  
1. 目标文件名可以省略引号  
2. 文件名部分支持使用\*实现模糊匹配  
3. 如果当前用户对目标路径没有访问权限, 则无法执行查找文件动作!

## 案例10: ln -s 链接文件

注意: 需要使用 -s 创建软链接(类似 Windows 系统下的快捷方式)

FinalShell 3.9.2.2

同步状态 ● IP 192.168.34.30 复制 系统信息 运行 3:13 负载 0.15, 0.21, 0.14 CPU 2% 内存 76% 1.3G/1.8G 交换 1% 12M/2G 内存 CPU 命令 2.9M 1 sshd 0 0.3 ksoftirqd, 760K 0.3 audit 840K 0.3 audispd ↑1K ↓51K en... 444K 308K 154K 0ms 本机 路径 可用/大小 /dev 893M/893M /dev... 909M/909M /run 899M/909M /sys... 909M/909M / 9.7G/17G /boot 847M/1014M /run/... 181M/181M 激活/升级 命令输入

```
[admin@localhost bdir]$ ls
[admin@localhost bdir]$ touch demo
[admin@localhost bdir]$ ls
demo
[admin@localhost bdir]$ ln -s demo ldemo
[admin@localhost bdir]$ ll
总用量 0
-rw-rw-r-- 1 admin admin 0 7月 30 11:33 demo
lrwxrwxrwx 1 admin admin 4 7月 30 11:34 ldemo -> demo
[admin@localhost bdir]$ ls > ldemo
[admin@localhost bdir]$ cat demo
demo
ldemo
[admin@localhost bdir]$ ls -l > demo
[admin@localhost bdir]$ cat ldemo
总用量 0
-rw-rw-r-- 1 admin admin 0 7月 30 11:36 demo
lrwxrwxrwx 1 admin admin 4 7月 30 11:34 ldemo -> demo
[admin@localhost bdir]$ rm demo
[admin@localhost bdir]$ ll
总用量 0
lrwxrwxrwx 1 admin admin 4 7月 30 11:34 ldemo -> demo
[admin@localhost bdir]$
```

01. 在当前路径下创建文件 demo  
02. 给 demo 文件创建链接文件名为 ldemo  
03. 修改 ldemo 链接文件的内容  
04. 查看 demo 文件的内容是否同样变化  
05. 修改 demo 文件内容, 查看 ldemo 链接文件内容是否同样变化

在 Linux 系统中存在两种链接文件方式:  
ln -s 原文件名链接文件名: 软链接, 类似 Windows 下的快捷方式,  
ln 原文件名链接文件名: 硬链接, 类似复制文件,

注意: 删除原文件, 链接文件会失效

## 扩展: 硬链接

FinalShell 3.9.2.2

同步状态 ● IP 192.168.34.30 复制 系统信息 运行 3:38 负载 0.01, 0.07, 0.12 CPU 2% 内存 59% 1G/1.8G 交换 1% 12M/2G 内存 CPU 命令 2.9M 1 sshd 0 0.3 kworker/ 3M 0.3 top 4M 0 systemd ↑1K ↓62K en... 434K 301K 150K 0ms 本机 路径 可用/大小 /dev 893M/893M /dev... 909M/909M /run 899M/909M /sys... 909M/909M / 9.7G/17G /boot 847M/1014M /run/... 181M/181M 激活/升级 命令输入

```
[admin@localhost ~]$ touch demo 创建文件
[admin@localhost ~]$ ls
demo
[admin@localhost ~]$ ln demo hdemo 创建硬链接文件
[admin@localhost ~]$ ll
总用量 0
-rw-rw-r-- 2 admin admin 0 7月 30 12:03 demo
-rw-rw-r-- 2 admin admin 0 7月 30 12:03 hdemo
[admin@localhost ~]$ ls -l > hdemo
[admin@localhost ~]$ cat demo
demo
hdemo
[admin@localhost ~]$ ls -l > demo
[admin@localhost ~]$ cat demo
总用量 0
-rw-rw-r-- 2 admin admin 0 7月 30 12:05 demo
-rw-rw-r-- 2 admin admin 0 7月 30 12:05 hdemo
[admin@localhost ~]$ rm demo
[admin@localhost ~]$ ll
总用量 4
-rw-rw-r-- 1 admin admin 107 7月 30 12:05 hdemo
[admin@localhost ~]$ cat hdemo
总用量 0
-rw-rw-r-- 2 admin admin 0 7月 30 12:05 demo
-rw-rw-r-- 2 admin admin 0 7月 30 12:05 hdemo
[admin@localhost ~]$ ll
总用量 4
-rw-rw-r-- 1 admin admin 107 7月 30 12:05 hdemo
[admin@localhost ~]$
```

软链接

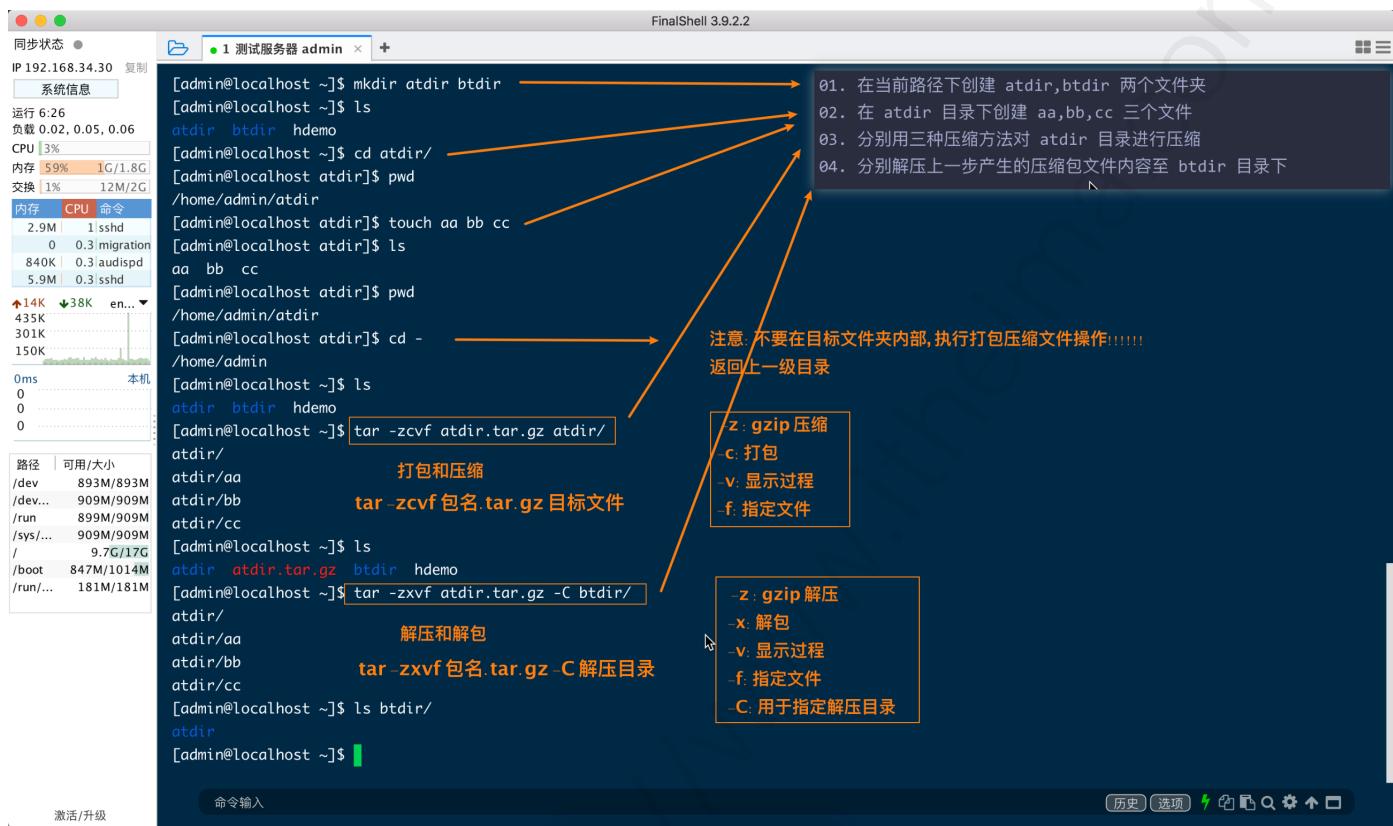
如果文件被删除, 则软链接文件失去指向, 变为不可用!

硬链接

如果文件被删除, 由于硬链接文件直接指向内容, 因此不受影响

# 案例11: tar/gzip/zip/unzip

## Part1: tar/gzip 打包压缩和解压解包



FinalShell 3.9.2.2

```
[admin@localhost ~]$ mkdir atdir btdir
[admin@localhost ~]$ ls
atdir btdir hdemo
[admin@localhost ~]$ cd atdir/
[admin@localhost atdir]$ pwd
/home/admin/atdir
[admin@localhost atdir]$ touch aa bb cc
[admin@localhost atdir]$ ls
aa bb cc
[admin@localhost atdir]$ pwd
/home/admin/atdir
[admin@localhost atdir]$ cd -
[admin@localhost ~]$ ls
atdir btdir hdemo
[admin@localhost ~]$ tar -zcvf atdir.tar.gz atdir/
[admin@localhost ~]$ ls
atdir/
atdir/aa
atdir/bb
atdir/cc
[admin@localhost ~]$ tar -zcvf atdir.tar.gz -C btdir/
[admin@localhost ~]$ ls btdir/
atdir/
atdir/aa
atdir/bb
atdir/cc
[admin@localhost ~]$ ls btdir/
atdir
```

01. 在当前路径下创建 atdir,btdir 两个文件夹  
02. 在 atdir 目录下创建 aa,bb,cc 三个文件  
03. 分别用三种压缩方法对 atdir 目录进行压缩  
04. 分别解压上一步产生的压缩包文件内容至 btdir 目录下

注意 不要在目标文件夹内部, 执行打包压缩文件操作!!!!

返回上一级目录

**z gzip 压缩**  
c: 打包  
-v: 显示过程  
-f: 指定文件

**z gzip 解压**  
x: 解包  
-v: 显示过程  
-f: 指定文件  
C: 用于指定解压目录

## Part2: zip 压缩/unzip 解压

FinalShell 3.9.2.2

同步状态 ●  
IP 192.168.34.30 复制  
系统信息  
运行 6:49  
负载 0.06, 0.07, 0.05  
CPU 3%  
内存 60% 1.1G/1.8G  
交换 1% 12M/2G  
内存 CPU 命令  
2.9M 1.3 sshd  
0 0.3 ksoftirqd  
0 0.3 rcu\_sched  
760K 0.3 auditd  
↑912B ↓49K en...  
171K  
118K  
59K  
0ms 本机  
0  
0  
0  
路径 可用/大小  
/dev 893M/893M  
/dev... 909M/909M  
/run 899M/909M  
/sys... 909M/909M  
/ 9.7G/17G  
/boot 847M/1014M  
/run... 181M/181M

```
[admin@localhost ~]$ ls
atdir atdir.tar.gz btdir hdemo
[admin@localhost ~]$ zip atdir atdir
adding: atdir/ (stored 0%)
[admin@localhost ~]$ ls
atdir atdir.tar.gz atdir.zip btdir hdemo 桌面
[admin@localhost ~]$ rm atdir.zip
[admin@localhost ~]$ zip -r atdir atdir
adding: atdir/ (stored 0%)
adding: atdir/aa (stored 0%)
adding: atdir/bb (stored 0%)
adding: atdir/cc (stored 0%)
[admin@localhost ~]$ unzip -d btdir/ atdir.zip
Archive: atdir.zip
replace btdir/atdir/aa? [y]es, [n]o, [A]ll, [N]one, [r]ename: a
error: invalid response [a]
replace btdir/atdir/aa? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
extracting: btdir/atdir/aa
extracting: btdir/atdir/bb
extracting: btdir/atdir/cc
[admin@localhost ~]$ ls
atdir atdir.tar.gz atdir.zip btdir hdemo 桌面
[admin@localhost ~]$ unzip atdir.zip
Archive: atdir.zip
replace atdir/aa? [y]es, [n]o, [A]ll, [N]one, [r]ename: N
[admin@localhost ~]$
```

01. 在当前路径下创建 atdir, btdir 两个文件夹  
02. 在 atdir 目录下创建 aa, bb, cc 三个文件  
03. 分别用三种压缩方法对 atdir 目录进行压缩  
04. 分别解压上一步产生的压缩包文件内容至 btdir 目录下

注意: 如果目标文件是文件夹, 需要使用 -r 选项处理文件夹内部的所有文件

正确样例

语法: zip -r 压缩包名 目标文件  
注意: 压缩包名的后缀 .zip 可以省略

替换 不替换 全部替换 什么也不做 改名

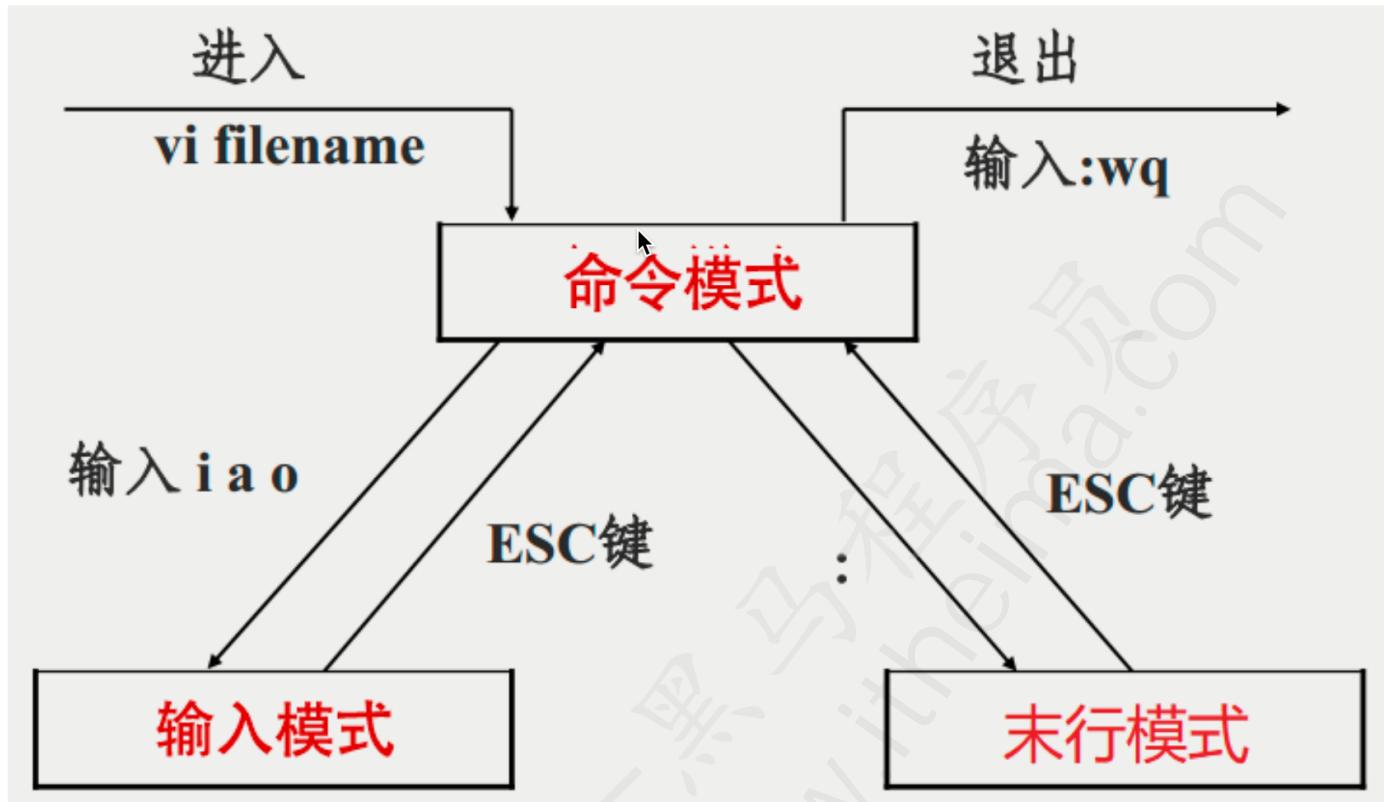
语法: unzip -d 解压路径 压缩包名  
注意: 解压需要先指定解压路径

说明: 解压文件时出现重名文件提示操作选项

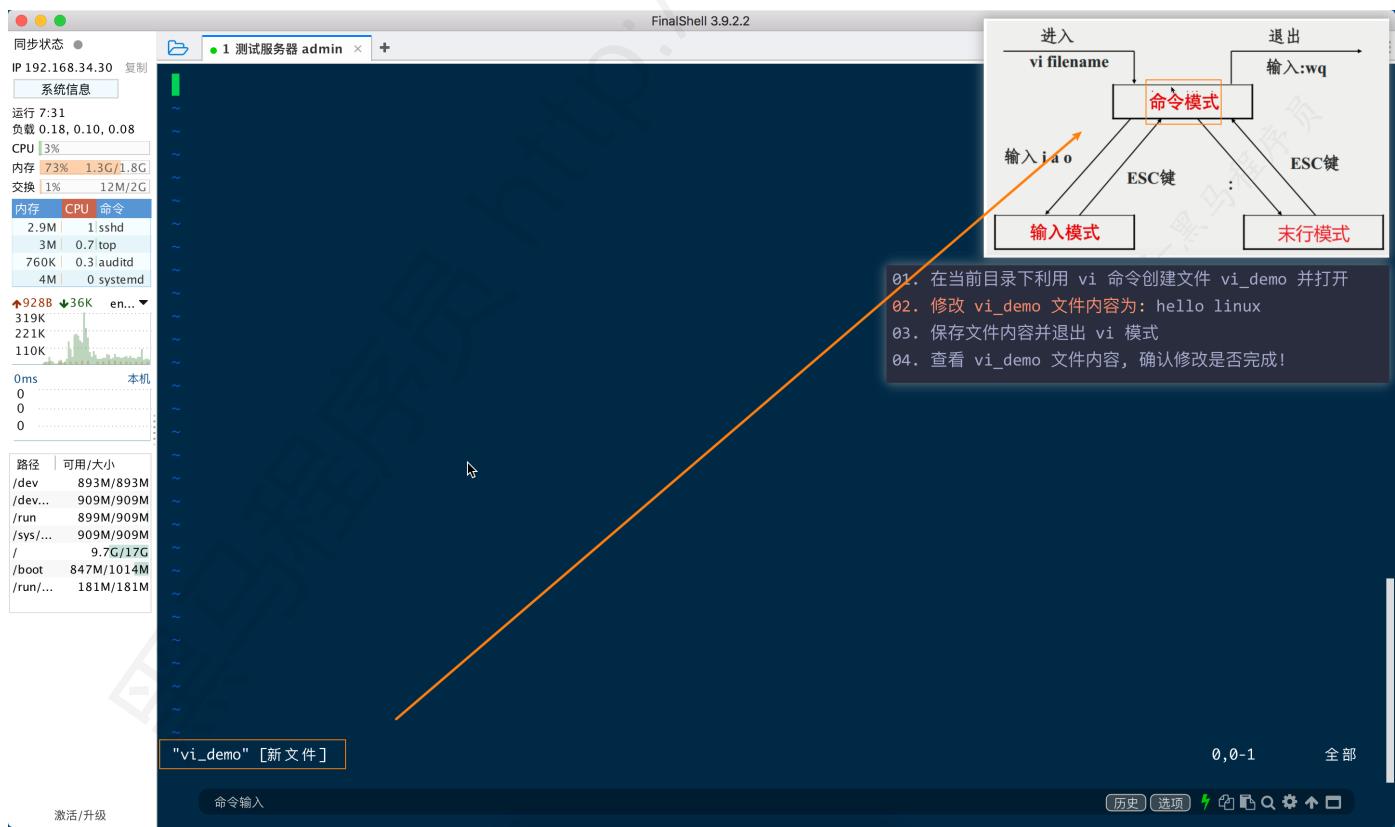
注意: 也可以直接在当前目录下解压缩文件

## 案例12: vi 命令行文本编辑器

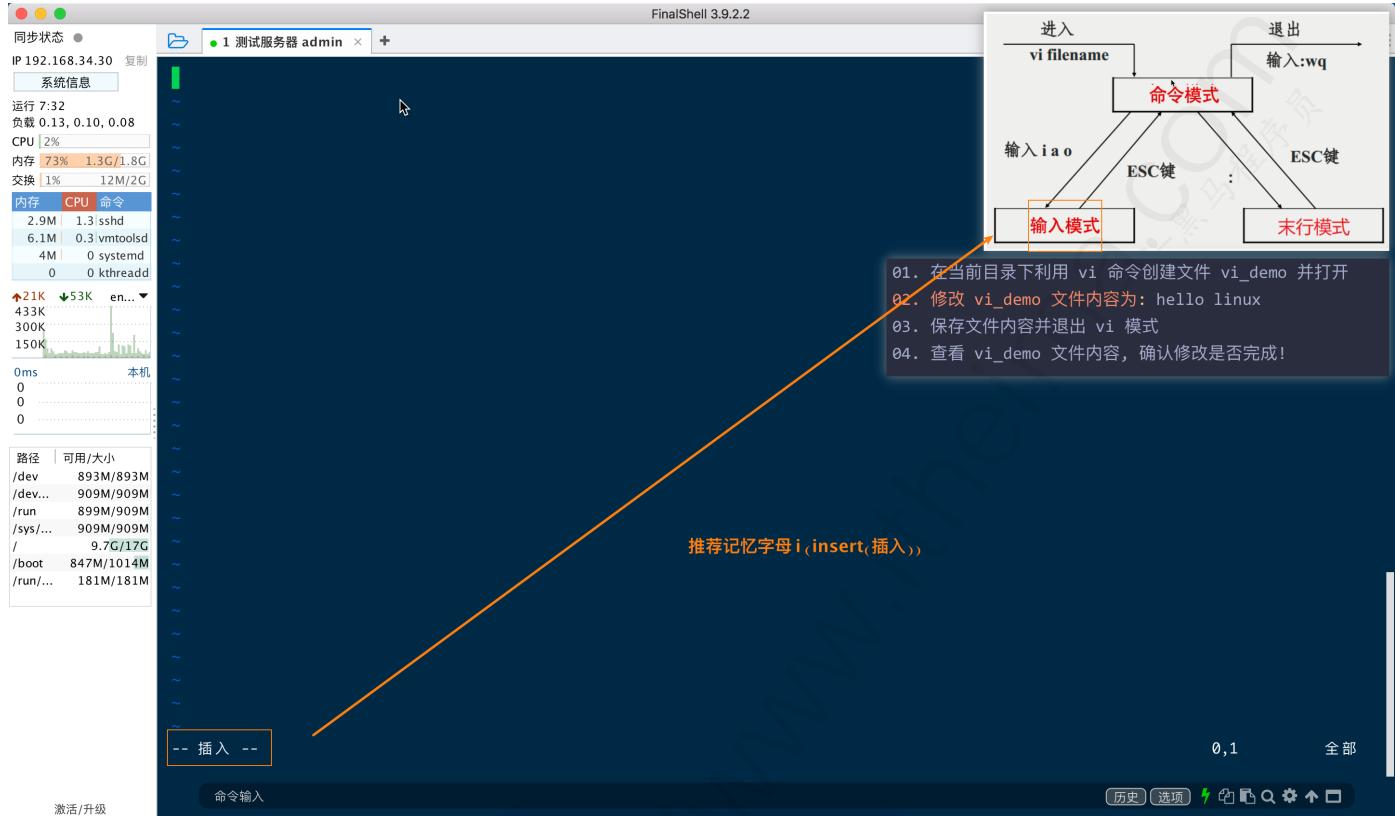
## 基本操作流程



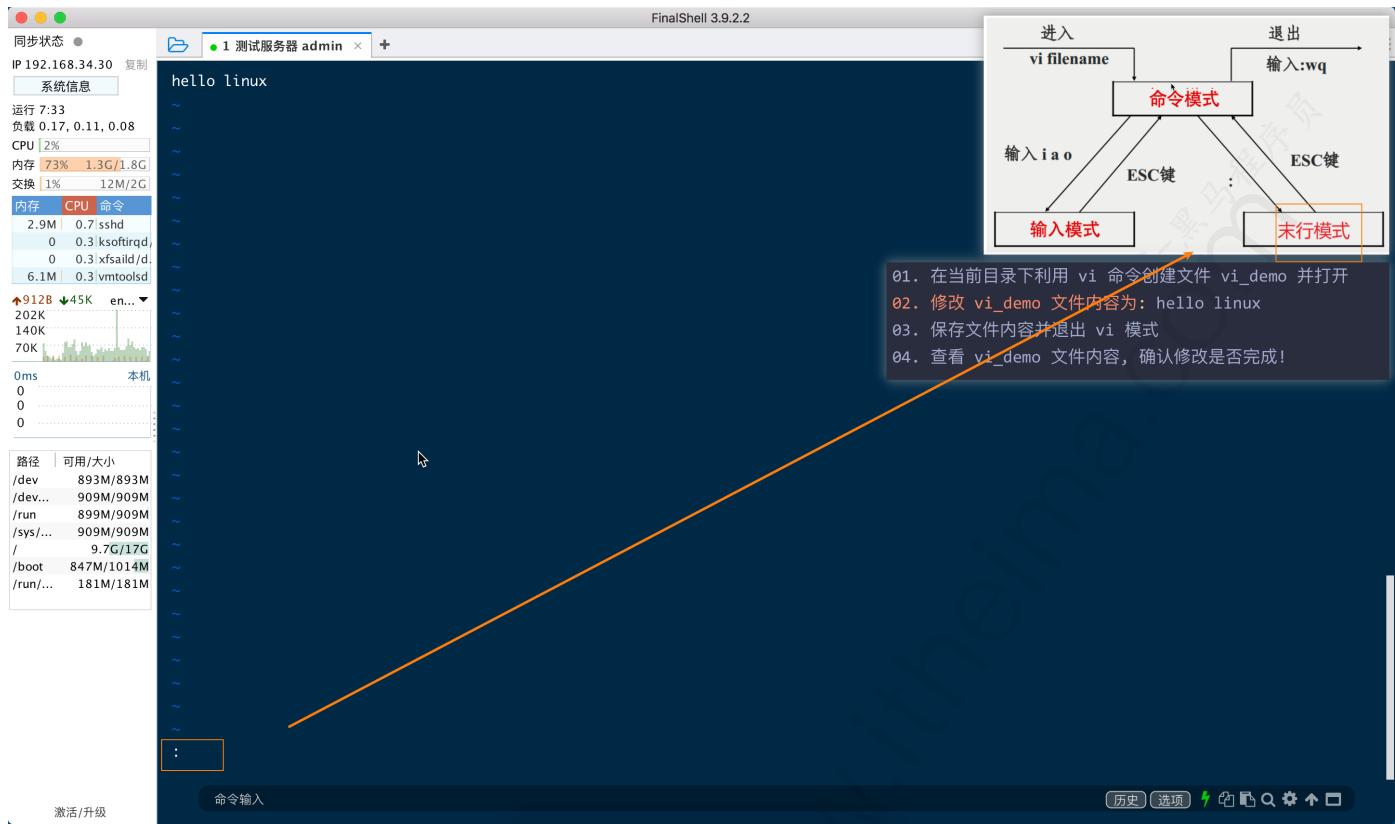
## 命令模式



## 输入模式



## 末行模式



## yum 软件包管理工具

### 说明:

yum (Yellow dog Updater, Modified) 是一个在 Linux 系统中常用的软件包管理器。

yum 提供了查找、安装、删除某一个、一组甚至全部软件包的命令，而且命令简洁而又好记。

### 常用命令:

`yum list`: 列出所有可安装的软件清单

`yum search`: 查找软件包

`yum install <package_name>`: 安装指定的软件

`yum update <package_name>`: 更新指定的软件

`yum update`: 更新所有软件

`yum remove <package_name>`: 删除软件包

# 数据库

**说明：**数据库是专门用来存储数据的软件

**注意：**对于测试工作而言，如果项目页面没有实现，但是我们又想要校验数据，则可以直接通过查询数据库实现

**关系：**具体存在的商品录入后 -> 产生对应的数据(存到数据库中) -> 最后会被加载到项目页面中

## 数据库的分类

**分类：**

1> **关系型数据库：**以数据表为核心

2> **非关系型数据库：**不存在数据表的概念

**关系型数据库：**RDMS(Relational Database Management System)关系型数据库系统

**常见的关系型数据库：**

**Oracle：**在大型项目中使用，例如：银行、电信等项目

**MySQL：**Web 项目中使用最广泛的关系型数据库

**Microsoft SQL Server：**在微软的项目中使用

**SQLite：**轻量级数据库，主要应用在移动平台

**关系型数据库的核心要素：**

数据行 (一条记录)

数据列 (字段)

数据表 (数据行的集合)

数据库 (数据表的集合，一个数据库中能够有 n 多个数据表)

关系型数据库核心要素示例

工作簿1

J3

开始 插入 绘图 页面布局 公式 数据 审阅 视图

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

A B C D E F G H I J K L M

数据行(一条数据)

数据列(字段)

数据表(大量数据行和数据列构成)

Sheet1 +

就绪

## SQL 语言

说明: SQL: Structured Query Language (结构化查询语言), 通过 SQL 语  
言可以对数据库进行操作

注意:

1. SQL 语言默认支持操作所有常见的关系型数据库
2. 作为测试人员, 必须要掌握 SQL 的查询语句  
(DQL: 数据查询语言, 用于对数据进行查询, 例如: `select`)
3. 对于 MySQL 而言, 编写 SQL 语句时, 不区分字母的大小写

# MySQL 数据库介绍

**说明：**MySQL 是一个关系型数据库管理系统，目前属于 Oracle 旗下产品，目前为止 MySQL 社区版是可以免费使用的

**特点：**开源/免费/跨平台(Windows/macOS/Linux)/跨语言(Java/Python...)

## 数据库连接工具 Navicat

**说明：**由于数据库软件处于服务器中，想要操作数据库，就必须使用工具远程连接数据库后，进行操作

## 连接数据库操作步骤

**说明：**将来在工作中，想要远程连接数据库，需要具备以下条件：

- 1> 数据库所在服务器的 IP 地址及数据库的端口号
- 2> 向相关人员获取数据库的账号和密码
- 3> 使用数据库连接工具，远程连接数据库即可

**注意：**远程连接需要注意网络连通性

## 今日任务

- 总结所有 Linux 命令(以将来自己可以快速查询并使用为准)
- 熟练操作 Navicat 工具(连接数据/操作数据库/操作数据表/操作数据等实现)**[最高优先级]**
- 本阶段需要完成未来简历内容中的第一部分内容(自我介绍)，晚自习会进行活动动员

黑马程序员 <http://www.itheima.com>