

# Linux 基本指令

## 本节要点

1. 掌握 Linux 的常用命令
2. 在 Linux 上能安装 MySQL / JDK / Tomcat 等 Java 常用软件.
3. 理解 Linux 中的权限, 能够进行简单的用户管理和权限控制.

## 1. 文件/目录基本操作

### 1.1 ls 指令

**语法:** ls [选项] [目录或文件]

**功能:** 对于目录, 该命令列出该目录下的所有子目录与文件。对于文件, 将列出文件名以及其他信息。

**常用选项:**

- -a 列出目录下的所有文件, 包括以 . 开头的隐含文件。
- -d 将目录象文件一样显示, 而不是显示其下的文件。如: ls -d 指定目录
- -k 以 k 字节的形式表示文件的大小。ls -alk 指定文件
- -l 列出文件的详细信息。
- -r 对目录反向排序。
- -t 以时间排序。
- -R 列出所有子目录下的文件。(递归)

**举例:**

```
ls -l
```

### 1.2 pwd命令

**语法:** pwd **功能:** 显示用户当前所在的目录

**举例:**

```
pwd
```

### 1.3 cd 指令

Linux系统中, 磁盘上的文件和目录被组成一棵目录树, 每个节点都是目录或文件。

**语法:** cd 目录名

**功能:** 改变工作目录。将当前工作目录改变到指定的目录下。 **举例:**

```
cd .. : 返回上级目录
cd ~: 进入用户家目
cd -: 返回最近访问目录
```

## 认识 Linux 目录结构

Linux 是一个树形目录结构.



几个特殊的目录:

- / 称为根目录
- . 称为当前目录
- .. 称为当前目录的上级目录

## 绝对路径 vs 相对路径

形如: `/usr/share/tomcat/logs/` 以根目录开头的, 称为绝对路径.

形如: `./logs` 以 `.` 或者 `..` 开头的, 称为相对路径.

## 使用 tab 键补全

我们敲的所有的 Linux 命令, 都可以使用 tab 键来尝试补全, 加快效率.

## 使用 ctrl + c 重新输入

如果命令或者目录敲错了, 可以 ctrl + c 取消当前的命令.

## 1.4 touch指令

**语法:** touch [选项]... 文件... **功能:** touch命令参数可更改文档或目录的日期时间, 包括存取时间和更改时间, 或者新建一个不存在的文件。 **举例:**

```
touch test.txt
```

## 1.5 mkdir 指令

**语法:** mkdir [选项] dirname... **功能:** 在当前目录下创建一个名为“dirname”的目录 **常用选项:**

- -p, --parents 可以是一个路径名称。此时若路径中的某些目录尚不存在,加上此选项后,系统将自动建立好那些尚不存在的目录,即一次可以建立多级目录

**举例:**

```
mkdir -p test/test1 : 递归建立多个目录
```

## 1.6 rm 指令

**语法:** rm [-f-i-r-v] [dirName/dir] **功能:** 删除文件或目录 **常用选项:**

- -f 即使文件属性为只读(即写保护),亦直接删除
- -i 删除前逐一询问确认
- -r 删除目录及其下所有文件

**举例:**

```
rm test.txt
```

**重要注意事项:**

千万不要运行 `rm -rf /`, 尤其是在公司的生产服务器上.

**理解递归删除的过程:**

先手动创建如下目录结构:

```
test
├── a
│   ├── a1
│   │   ├── 1.txt
│   │   └── 2.txt
│   └── a2
├── b
│   ├── b1`
│   │   ├── 1.txt
│   │   └── 2.txt
│   └── b2
└── c
```

使用 `rm -ri` 命令删除 `test`，观察删除的顺序。

## 1.7 cp指令

**语法:** `cp [选项] 源文件或目录 目标文件或目录`

**功能:** 复制文件或目录

**说明:** `cp`指令用于复制文件或目录，如同时指定两个以上的文件或目录，且最后的目的地是一个已经存在的目录，则它会把前面指定的所有文件或目录复制到此目录中。若同时指定多个文件或目录，而最后的目的地并非一个已存在的目录，则会出现错误信息 **常用选项:**

- `-f` 或 `--force` 强行复制文件或目录，不论目的文件或目录是否已经存在
- `-i` 或 `--interactive` 覆盖文件之前先询问用户
- `-r`递归处理，将指定目录下的文件与子目录一并处理。若源文件或目录的形态，不属于目录或符号链接，则一律视为普通文件处理
- `-R` 或 `--recursive`递归处理，将指定目录下的文件及子目录一并处理

**举例:**

```
cp test1.txt test2.txt
```

## 1.8 mv指令

**语法:** `mv [选项] 源文件或目录 目标文件或目录`

**功能:**

1. 视`mv`命令中第二个参数类型的不同（是目标文件还是目标目录），`mv`命令将文件重命名或将其移至一个新的目录中。
2. 当第二个参数类型是文件时，`mv`命令完成文件重命名，此时，源文件只能有一个（也可以是源目录名），它将所给的源文件或目录重命名为给定的目标文件名。
3. 当第二个参数是已存在的目录名称时，源文件或目录参数可以有多个，`mv`命令将各参数指定的源文件均移至目标目录中。

**常用选项**

- `-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
- `-i` : 若目标文件 (destination) 已经存在时，就会询问是否覆盖！

**举例**

```
mv test1.txt test2.txt
```

## 1.9 cat指令

**语法:** cat [选项] [文件]

**功能:** 查看目标文件的内容

**常用选项:**

- -n 对输出的所有行编号

```
cat test.txt
```

## 1.10 man指令

Linux的命令有很多参数，我们不可能全记住，我们可以通过查看联机手册获取帮助。**语法:** man [选项] 命令

**常用选项**

- -k 根据关键字搜索联机帮助
- num 只在第num章节找
- man man 能够看到 man 手册中的若干个章节及其含义.

**举例**

```
man ls
```

## 1.11 less指令

**语法:** less [参数] 文件 **功能:** 查看文件内容. 不会立刻把所有文件内容加载到内存中. 也能进行查找. **选项:**

- j k / 方向键: 向上向下滚动屏幕.
- -N 显示每行的行号
- /字符串: 向下搜索“字符串”的功能
- n: 重复前一个搜索 (与 / 或 ? 有关)
- q: 退出

## 1.12 head指令

**语法:** head [参数]... [文件]...

**功能:**

head 用来显示档案的开头至标准输出中，默认head命令打印其相应文件的开头10行。

**选项:**

- -n<行数> 显示的行数

## 1.13 tail指令

**语法:** tail [必要参数] [选择参数] [文件] **功能:** 用于显示指定文件末尾内容, 不指定文件时, 作为输入信息进行处理。常用查看日志文件。

**选项:**

- -f 循环读取
- -n<行数> 显示行数

**举例:** 有一个文件共有100行内容, 请取出第50行内容

```
# 方法1
head -n50 test > tmp      # 将前50行装入临时文件 tmp
tail -n1 tmp              # 得到中间行

# 方法2
head -n50 test | tail -n1
```

### 关于重定向

Linux 的很多指令, 如 cat head tail 等都是默认输出到 "标准输出" 中, 也就是显示器上的. 可以通过 > 这样的符号把本来要输出到标准输出上的内容写到特定文件中. 这样的操作称为 "重定向".

重定向有三种方式:

#### 1. 标准输入重定向(<)

```
# 先构造一个文件, 里面添加一些内容.
cat < test.txt
```

#### 2. 标准输出重定向(>)

```
cat test.txt > test2.txt
```

#### 3. 标准错误重定向(2>), 注意 2 和 > 之间不能有空格.

```
# 尝试删除一个不存在的文件
rm aaa > test.txt

# 输出结果, 这个提示并没有被重定向到 test.txt 文件中.
rm: cannot remove 'aaa': No such file or directory

# 正确做法
rm aaa 2> test.txt
```

标准输入, 标准输出, 标准错误是三个特殊的文件, 每个进程在启动的时候都会默认打开. 分别对应到键盘, 显示器, 显示器这样的设备.

在 Java 中, 分别对应到 `System.in`, `System.out`, `System.err`

### 关于管道

管道是一种古老的 "进程间通信" 方式. 在 Linux 指令中可以使用 | 作为管道标记.

意思是将前一个指令标准输出的内容, 作为第二个指令的标准输入内容.

## 1.14 date 命令

date 指定格式显示时间: `date +%Y:%m:%d`

date 用法: `date [OPTION]... [+FORMAT]`

1. 在显示方面, 使用者可以设定欲显示的格式, 格式设定为一个加号后接数个标记

- %H : 小时(00..23)
- %M : 分钟(00..59)
- %S : 秒(00..61)
- %X : 相当于 %H:%M:%S
- %d : 日 (01..31)
- %m : 月份 (01..12)
- %Y : 完整年份 (0000..9999)
- %F : 相当于 %Y-%m-%d

### 2. 时间戳

时间->时间戳: `date +%s`

时间戳->时间: `date -d@1508749502`

Unix时间戳 (英文为Unix epoch, Unix time, POSIX time 或 Unix timestamp) 是从1970年1月1日 (UTC/GMT的午夜) 开始所经过的秒数, 不考虑闰秒。

## 2. 搭建 Java 部署环境

### 2.1 使用 yum 命令

#### 认识 yum

- 在Linux下安装软件, 一个通常的办法是下载到程序的源代码, 并进行编译, 得到可执行程序.
- 但是这样太麻烦了, 于是有些人把一些常用的软件提前编译好, 做成软件包(可以理解成windows上的安装程序)放在一个服务器上, 通过包管理器可以很方便的获取到这个编译好的软件包, 直接进行安装.
- 软件包和软件包管理器, 就好比 "App" 和 "应用商店" 这样的关系.
- yum(Yellow dog Updater, Modified)是Linux下非常常用的一种包管理器. 主要应用在Fedora, RedHat, Centos等发行版上.

yum 起到的功能和 Maven 的依赖管理功能类似. 使用 Maven 能帮我们方便的安装一些第三方 jar 包, 而 yum 方便我们方便的安装第三方程序.

类似的, Github 也能起到 "软件仓库" 的效果, 而且确实有些编程语言的包管理工具就是基于 Github (例如 Go 语言). 只不过, Github 不光能用于分发程序, 也能管理源码并进行协同开发, 而 yum 和 maven 都是仅用于分发程序.

查看软件包列表

```
yum list | grep [软件包关键字]    # 注意，最好要加上 grep，否则罗列的内容会非常多，导致机器很卡。
```

### 安装软件包(需要管理员权限)

```
yum install [软件包名字]
```

### 卸载软件包(需要管理员权限)

```
yum remove [软件包名字]
```

### 注意事项

- yum 所有的命令必须保证网络是联通情况下, 才能使用.
- yum install / yum remove 必须具备管理员权限(root 用户).
- 可以使用 `ping www.baidu.com` 来检测网络的畅通情况.

## 2.2 安装 git

### 1. 查看 git 安装包

```
# 由于带 git 关键字的软件包很多，可以在 grep 的时候加上 -w，表示全字匹配。  
yum list | grep git -w
```

### 2. 安装 git

```
yum install git.x86_64
```

### 3. git 的基本使用(和 Windows 版本的 git 是一致的. 只是使用命令行操作)

```
git clone  
git add  
git commit  
git push
```

## 2.3 安装 Maven

### 1. 查看 Maven 安装包

```
yum list | grep maven -w
```

### 2. 安装 Maven

```
yum install maven.noarch
```



### 3. 验证 Maven

```
mvn -v
```

# 输出结果

Apache Maven 3.0.5 (Red Hat 3.0.5-17)

Maven home: /usr/share/maven

Java version: 1.8.0\_232, vendor: Oracle Corporation

Java home: /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.232.b09-0.e17\_7.x86\_64/jre

Default locale: en\_US, platform encoding: UTF-8

OS name: "linux", version: "3.10.0-957.5.1.e17.x86\_64", arch: "amd64", family: "unix"

此时说明安装成功.

## 2.4 安装 MySQL

MySQL 同样也可以使用 yum 安装, 但是要修改一些配置, 稍微复杂一些.

可以参考陈沛鑫老师写的一个知乎上的文章.

<https://zhuanlan.zhihu.com/p/49046496>

在数据库运行过程中出现问题, 可以查看 MySQL 的错误日志.

在 MySQL 中通过这个命令, 获取到日志的路径

```
mysql> show variables like 'log_error';
```

# 输出结果

variable_name	value
log_error	/var/log/mariadb/mariadb.log

使用 vim 或 less 查看该文件内容即可.

```
less /var/log/mariadb/mariadb.log
```

MySQL 中除了错误日志, 还有很多其他类型的日志. 详细可以参考

<https://www.cnblogs.com/f-ck-need-u/p/9001061.html>

## 2.5 安装 JDK

由于 yum 源上的 JDK 是 openjdk, 和官方提供的 jdk 存在一定差异. 我们只能手动安装官方 JDK.

### 1. 下载官方 JDK rpm 包

在 Linux 上直接使用 `wget` 来进行下载.

安装地址参考 Java 官网 [https://www.java.com/zh\\_CN/download/linux\\_manual.jsp](https://www.java.com/zh_CN/download/linux_manual.jsp)

```
wget https://javadl.oracle.com/webapps/download/AutoDL?
BundleId=240717_5b13a193868b4bf28bcb45c792fce896 -O jdk.rpm
```

### 关于 rpm

rpm 就是刚才在介绍 yum 中提到的 "软件包". 类似于一个 Windows 的安装包文件.

yum 的功能就是帮我们自动管理 rpm 包. 也可以使用 rpm 命令手动管理.

## 2. 安装 JDK

```
# 安装软件包操作需要 root 权限(后面会介绍)
rpm -ivh jdk.rpm
```

## 3. 验证安装成功

```
java -version

# 输出结果
java version "1.8.0_231"
Java(TM) SE Runtime Environment (build 1.8.0_231-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.231-b11, mixed mode)
```

如果提示 "java 命令找不到" 则说明安装失败.

## 4. 写一个简单的 hello world 程序

- 使用 vim 创建 Test.java 文件
- 按下 i 进入插入模式, 输入 hello world 代码
- esc 返回到普通模式
- 使用 `javac / java` 命令编译执行

### 关于 vim

vim 是一个非常古老的命令行下的开发工具. 可以理解成一个更高端的记事本.

很多 Linux 都会内置 vim, 作为默认编辑器. 我们后续在服务器上直接修改一些配置文件的时候经常会用到.

vim 的操作风格非常有特点, 分成多种 "模式"

- 普通模式: 启动 vim 默认是普通模式. 此时键盘按键不是直接输入文本, 而是各种功能的快捷键.
- 插入模式: 在普通模式中按下 i 进入插入模式, 可以进行编辑.

3. 命令模式: 编辑完毕后, 按 `esc` 返回到普通模式, 然后输入 `:` 进入命令模式, 此时光标在屏幕最下面一行. 然后输入 `wq`, 表示 "保存退出" 的含义.

PS: vim 对于新手非常不友好, 因为快捷键实在太多了, 难以记忆(例如使用 `h j k l` 作为方向键), 又没有直观的像 "菜单" 这样的东西. 但是一旦使用熟练, 会容易让人沉迷难以自拔. 但是这个熟练的过程可能需要几年的时间.

PSS: vim 玩的溜的人类似于星际争霸玩家, 有一种 "手速如飞, 微操如神" 的感觉.

PSSS: IDEA 上也支持 vim 快捷键风格的插件. 有兴趣的童鞋可以试试.

## 2.6 安装 Tomcat

由于 yum 源上默认的 Tomcat 7 版本, 比较旧了. 我们课堂上使用 Tomcat 8, 需要手动安装, 不能使用 yum.

### 1. 下载 Tomcat 压缩包

下载路径可以参考官网 <https://tomcat.apache.org/download-80.cgi>

```
wget http://mirrors.tuna.tsinghua.edu.cn/apache/tomcat/tomcat-8/v8.5.47/bin/apache-tomcat-8.5.47.zip -O apache-tomcat-8.5.47.zip
```

### 2. 解压缩 Tomcat

```
# 使用 unzip 命令解压缩
unzip apache-tomcat-8.5.47.zip
```

### 3. 修改可执行权限

```
cd apache-tomcat-8.5.47/bin

# 将所有 .sh 后缀的文件加上可执行权限(后面会介绍)
chmod +x *.sh
```

### 4. 启动 Tomcat

```
sh bin/startup.sh
```

验证启动成功

```
# 方法1 查看 tomcat 进程是否存在
ps aux | grep tomcat

# 方法2 查看端口 8080 是否被绑定
netstat -anp | grep 8080

# 方法3 使用 curl 命令访问默认 demo
curl 127.0.0.1:8080
```

如果进程存在或者端口状态正确(LISTEN状态)或者能够访问到默认主页, 说明启动成功.

## 5. Tomcat 的目录结构

其中几个比较重要的:

- bin目录: tomcat 启动/停止脚本.
- conf: tomcat 各种配置文件

```
# 进入到这个目录中
cd conf
# 搜索 8080 关键字在哪些文件中存在.
grep 8080 * -r

# 输出结果
server.xml:      Define a non-SSL HTTP/1.1 Connector on port 8080
server.xml:      <Connector port="8080" protocol="HTTP/1.1"
server.xml:      port="8080" protocol="HTTP/1.1"

# server.xml 就是我们配置端口号的文件.
```

- logs: 日志目录. 如果 tomcat 运行过程中出现问题, 可以来查看这个日志. (使用 cat / less / vim 命令打开日志文件即可查看)

## 6. 外网访问 tomcat 默认 demo

在浏览器地址栏中

```
http://[服务器外网ip]:8080/
```

即可看到默认的面

## Apache Tomcat/8.5.47



If you're seeing this, you've successfully installed Tomcat. Congratulations!



Recommended Reading:

[Security Considerations How-To](#)

[Manager Application How-To](#)

[Clustering/Session Replication How-To](#)

Server Status

Manager App

Host Manager

## Developer Quick Start

[Tomcat Setup](#)

[First Web Application](#)

[Realms & AAA](#)

[JDBC Data Sources](#)

[Examples](#)

[Servlet Specifications](#)

[Tomcat Versions](#)

## Managing Tomcat

For security, access to the `manager webapp` is restricted. Users are defined in:

`$CATALINA_HOME/conf/tomcat-users.xml`

In Tomcat 8.5 access to the manager application is split between different users. [Read more...](#)

[Release Notes](#)

[Changelog](#)

[Migration Guide](#)

[Security Notices](#)

## Documentation

[Tomcat 8.5 Documentation](#)

[Tomcat 8.5 Configuration](#)

[Tomcat Wiki](#)

Find additional important configuration information in:

`$CATALINA_HOME/RUNNING.txt`

Developers may be interested in:

[Tomcat 8.5 Bug Database](#)

[Tomcat 8.5 JavaDocs](#)

[Tomcat 8.5 Git Repository at GitHub](#)

## Getting Help

[FAQ and Mailing Lists](#)

The following mailing lists are available:

[tomcat-announce](#)

Important announcements, releases, security vulnerability notifications. (Low volume).

[tomcat-users](#)

User support and discussion

[taglibs-user](#)

User support and discussion for [Apache Taglibs](#)

[tomcat-dev](#)

Development mailing list, including commit messages

在外网访问这个页面之前需要先开启服务器的 "安全组" 功能.

登陆自己的云服务账户, 在主页中找到 控制台 -> 自己的服务器 -> 安全组

需要配置安全组, 允许外部主机访问服务器的 8080 端口.

备注: 安全组配置界面如果找不到, 可以随时咨询云服务器客服.

(1) 选择加入安全组

The screenshot shows the Alibaba Cloud console interface. At the top, there are tabs for '新建' (New), '开机' (Start), '关机' (Stop), '重启' (Restart), '续费' (Renew), '重置密码' (Reset Password), and '更多操作' (More Actions). Below these tabs is a table of ECS instances. The first instance is 'ins-ggucsx9d' with the name '比特科技教学'. It is in the '运行中' (Running) state, located in the '北京三区' (Beijing Zone 3) region, and is a '标准型S2' (Standard S2) instance. The '更多操作' (More Actions) dropdown menu is open, showing options like '销毁/退还' (Destroy/Refund), '实例设置' (Instance Settings), '加载密钥' (Load Key Pair), '资源调整' (Resource Adjustment), and '加入安全组' (Add to Security Group). The '加入安全组' option is highlighted with a blue box. The table also shows the instance's IP address (49.233.172.121), billing mode (包年包月 - Pay by Month), and network type (公网 - Public Network).

(2) 选择新建安全组

加入安全组

×

您已选1个实例 [查看详情](#)

ID/实例名	IP地址	网络计费模式/带宽
<a href="#">ins-qgucsx9d</a> 比特科技教学	49.233.172.121 (公) 172.21.0.12 (内)	包月带宽 1Mbps

每个实例至少需要加入一个安全组， [新建或查看我的安全组详情](#)。

选择处于当前地区和项目的安全组

搜索名称、ID

安全组ID安全组名称

暂无数据

确认取消

(3) 点击新建按钮

广州上海北京成都重庆中国香港新加坡曼谷孟买首尔东京硅谷弗吉尼亚多伦多法兰克福

+新建

多个关键字用竖线 | 分隔，多个过滤标签

ID/名称	关联实例数	备注	类型	创建时间	项目
没有记录					

(4) 选择安全组的模板, 推荐使用第二个模板

新建安全组

模板

放通22, 80, 443, 3389端口和ICMP协议

放通全部端口

名称

放通22, 80, 443, 3389端口和ICMP协议

自定义

所属项目

默认项目

备注

公网放通云主机常用登录及web服务端口, 内网全放通。

显示模板规则

确定

取消

(5) 在任意一行规则的地方, 点击插入, 新增一行安全组规则即可.

☐

0.0.0.0/0

TCP:8080

允许

tomcat端口

编辑

插入

删除

## 2.7 小结

通过上面的操作, 我们又学习到了一些命令

- yum 是 Linux 中的一种包管理器. 帮助我们方便的安装管理程序.
- find 查找文件所在目录. 示例 `find [路径] -name [关键字]`
- grep 查看某个字符串在哪些文件中包含 `grep [关键字] [文件]`
- ps 查看进程. 示例 `ps aux | grep xxx`
- netstat 查看网络状态. 示例 `netstat -anp | grep xxx`
- curl 是一个命令行的 http 客户端程序
- wget 也是一个命令行的 http 客户端程序
- \* 是一个特殊的符号, 称为 **通配符**, 可以用来替代任何其他字符.

## 3. 认识 Linux 权限

### 3.1 认识 Linux 用户

**Linux下有两种用户：超级用户（root）、普通用户。**

- 超级用户：可以再linux系统下做任何事情，不受限制
- 普通用户：在linux下做有限的事情。
- 超级用户的命令提示符是“#”，普通用户的命令提示符是“\$”。

我们的服务器买好了, 默认是 root 用户. 但是 root 用户权利比较大, 一旦使用不当可能会造成严重后果(例如 `rm -rf /`). 因此我们真正在公司中不会直接使用 root 用户来操作服务器.

## 1. 创建用户

**命令** `useradd [用户名]`

**功能** 创建新的用户

**示例**

```
useradd test
```

## 2. 配置密码

**命令** `passwd [用户名]`

**功能** 设置或修改用户密码

**示例**

```
passwd test
```

注意, 输入密码的时候, 在控制台中不会有提示. 但实际上已经输入进去了.

## 3. 切换用户

**命令:** `su [用户名]` **功能:** 切换用户。例如, 要从root用户切换到普通用户user, 则使用 `su user`。要从普通用户user切换到root用户则使用 `su root` (root可以省略), 此时系统会提示输入root用户的口令。

## 3.2 三种角色

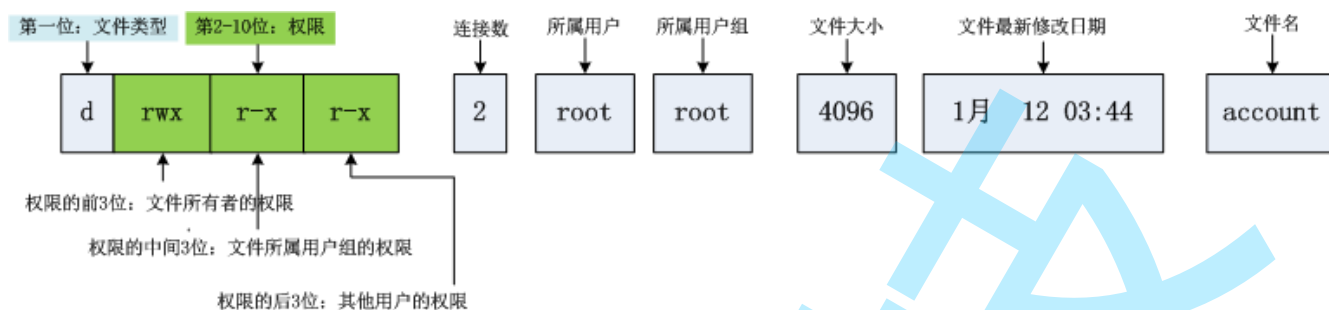
访问一个文件的人可能分成三类别.

- 文件和文件目录的所有者: u---User
- 文件和文件目录的所有者所在的组的用户: g---Group
- 其它用户: o---Others

## 3.3 文件类型和访问权限



命令 11 显示了一个文件的详细信息. 解读如下图:



#### a) 文件类型

d: 文件夹  
-: 普通文件  
l: 软链接 (类似Windows的快捷方式)  
b: 块设备文件 (例如硬盘、光驱等)  
p: 管道文件  
c: 字符设备文件 (例如屏幕等串口设备)  
s: 套接口文件

#### b) 基本权限

- i. 读 (r/4) : Read对文件而言, 具有读取文件内容的权限; 对目录来说, 具有浏览该目录信息的权限
- ii. 写 (w/2) : Write对文件而言, 具有修改文件内容的权限; 对目录来说具有删除移动目录内文件的权限
- iii. 执行 (x/1) : execute对文件而言, 具有执行文件的权限; 对目录来说, 具有进入目录的权限
- iv. "-" 表示不具有该项权限

## 3.4 chmod 命令

**功能:** 设置文件的访问权限

**格式:** chmod [参数] 权限 文件名

**常用选项:**

- R -> 递归修改目录文件的权限
- 说明: 只有文件的拥有者和root才可以改变文件的权限

chmod命令权限值的格式

#### ① 用户表示符+/-/=权限字符

- +: 向权限范围增加权限代号所表示的权限
- -: 向权限范围取消权限代号所表示的权限
- =: 向权限范围赋予权限代号所表示的权限
- 用户符号:
- u: 拥有者

- g: 拥有者同组用
- o: 其它用户
- a: 所有用户

**实例:**

```
# chmod u+w /home/abc.txt  
# chmod o-x /home/abc.txt  
# chmod a=x /home/abc.txt
```

②三位8进制数字

**实例:**

```
# chmod 664 /home/abc.txt  
# chmod 640 /home/abc.txt
```

类似的还有 chgrp, chown 命令. 同学们自行了解.

## 总结

本章内容是 Linux 的最基本的操作, 需要大家在实践中掌握.