

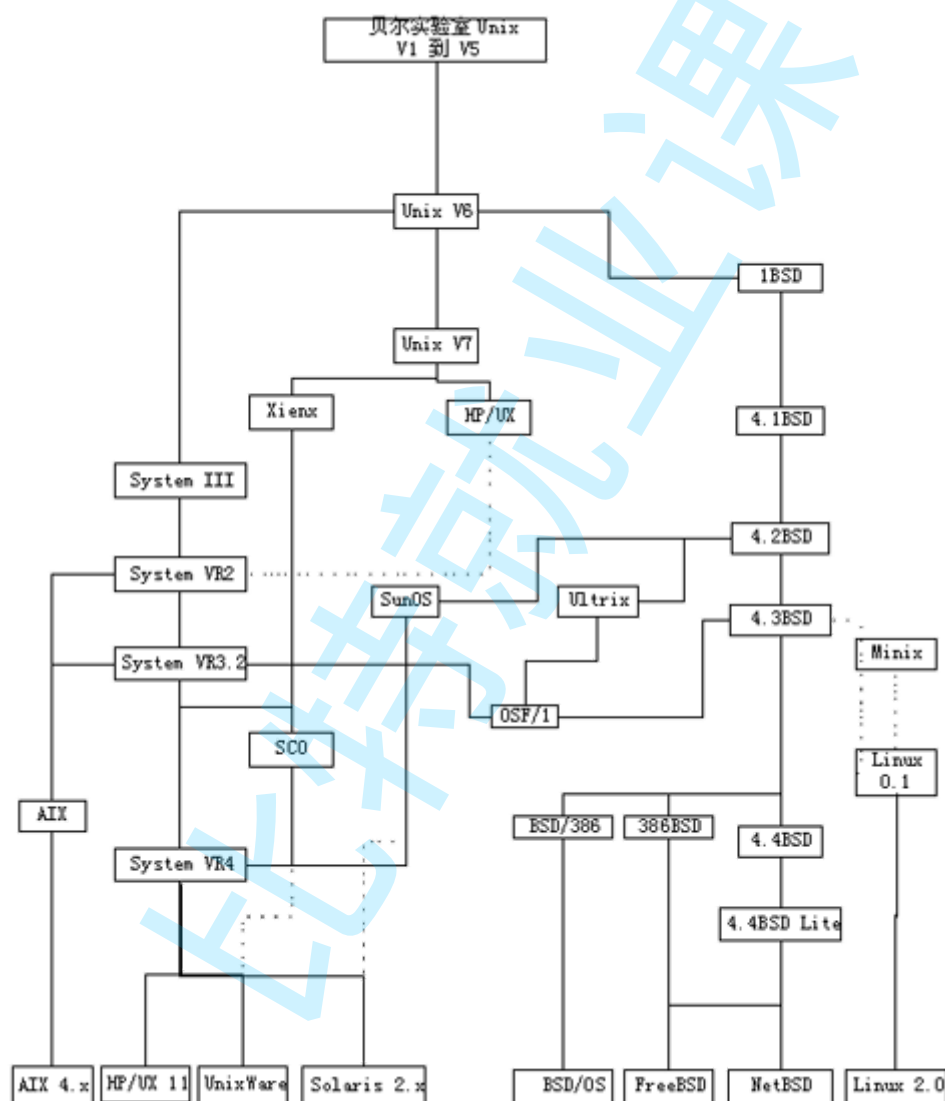
Linux 基本使用和 web 程序部署

Linux 背景知识

Linux 是什么

Linux 是一个操作系统. 和 Windows 是 "并列" 的关系.

Unix & Linux 发展流程图



1) 1969 - 1970 年, 贝尔实验室的 Dennis Ritchie (左图) 和 Ken Thompson (右图) 开发了 Unix 操作系统.



这两大佬本来是想开发一个操作系统, 但是开发着开发着, 发现现有的编程语言不太好使, 于是俩人一合计, 先整个编程语言出来吧. 于是 **C 语言** 就诞生了.

Dennis Ritchie 大佬 2011 年就去天堂写代码了. Ken Thompson 大佬后来在 google, 又参与发明了 Go 语言.

2) Unix 火了之后, 衍生出很多的分支. 其中有一只为 "Minix". 由荷兰的 Andrew S. Tanenbaum 教授开发. MINIX 含义为 "小型的 UNIX", 源代码开放给大学教学和研究工作.



3) 1991 年, 还在读大学的 芬兰人 Linus Benedict Torvalds (1969年12月28日出生, 当时 22 岁) 基于 Minix 的启发, 开发出了 Linux 的第一个版本.



值得一提的是, Linus 在开发 Linux 的时候, 觉得现有的 版本管理工具 并不太方便. 于是开发了一个新的版本管理工具. 也就是现在最广泛使用的 git .

Linus 大佬现在 50 多岁了, 但是仍然在互联网上非常活跃. 经常会拍一些视频. 在 油管 或者 B站 上就可以找到.

<https://www.bilibili.com/video/BV1qy411b79d?from=search&seid=3828654573220073441>

经过这么多年的发展, Linux 已经成为 **世界第一大操作系统**.

有的同学可能表示, Linux 都是世界第一大操作系统了, 我咋还没听说过? 不, 其实你听说过. 安卓系统本质上就是 Linux.

- 服务器领域
- 嵌入式设备
- 移动端

Linux 发行版

Linux 严格意义来说只是一个 "操作系统内核".

一个完整的操作系统 = 操作系统内核 + 配套的应用程序.

由于 Linux 是一个完全开源免费的内核, 因此有些公司/开源组织又基于 Linux 内核, 提供了不同的配套程序. 这就构成了不同的 "发行版".



企业中最主要使用的发行版是 RedHat (红帽). 但是 RedHat 是一个收费的系统(报价参考 <https://www.redhat.com/en/store/linux-platforms?intcmp=701f20000012m33AAA>). 因此我们课堂上使用的是基于红帽系统的社区免费版本 CentOS.

CentOS 和 RedHat 的关系

RedHat一直都提供源代码的发行方式, CentOS就是将RedHat发行的源代码去掉 RedHat 的 logo 商标, 重新编译一次.

RedHat对这种发行版的态度是: "我们其实并不反对这种发行版, 真正向我们付费的用户, 他们重视的并不是系统本身, 而是我们所提供的商业服务。"

所以, CentOS可以得到RedHat的所有功能, 甚至是更好的软件. 但CentOS并不向用户提供商业支持, 当然也不负上任何商业责任.

关于 Linux 我们学习什么

1) 基础命令

Linux 虽然也有图形化界面, 但是在 服务器 / 嵌入式设备上往往都是通过命令行的方式操作的. 因此学习 Linux 命令就是使用 Linux 的重要基础.

使用命令相比于使用图形界面主要有以下好处:

- 节省系统资源: 运行图形界面需要让系统付出一些额外的资源开销. 尤其是对于配置比较低的嵌入式设备, 这一点至关重要.
- 节省网络带宽: 如果通过网络访问服务器, 使用图形界面需要传输一帧一帧的图像, 而使用命令只需要传输简单的字符串.
- 便于批量执行任务: 可以通过一些 "脚本" 代码 (比如 Linux Shell) 来批量执行一些任务, 完成一些简单的编程工作. (比如定时备份文件, 删除文件等).

Windows 也有命令(也就是 cmd), 只是对于普通用户来说很少使用.

2) 系统编程 & 网络编程

Linux 自身提供了一些 API, 供程序猿调用来完成一些更复杂的编程任务(比如文件操作, 多线程编程, socket 编程等).

但是由于 Java 跨平台的特性, 这部分功能已经被 Java 自身封装好了 (流对象, Thread 对象, Socket 对象等). 所以这部分内容我们不必再学习了.

3) 部署 JavaWeb 项目

我们自己写的 web 程序, 要想让其他的用户能够访问, 就需要发布到服务器上. 这是我们接下来重点学习的内容.

小结: 对于 Java 程序猿, 关于 Linux 重点学习 基础命令 和 项目部署 即可.

Linux 环境搭建

要想学习 Linux, 需要先有一个 Linux 的环境.

环境搭建方式

主要有四种:

1. 直接安装在物理机上. 但是由于 Linux 桌面使用起来非常不友好. **[不推荐]**.
2. 使用虚拟机软件, 将 Linux 搭建在虚拟机上. 但是由于当前的虚拟机软件(如 VMWare 之类的)存在一些 bug, 会导致环境上出现各种莫名其妙的问题, 比较折腾. **[非常不推荐]**
3. 使用 WSL (Windows Subsystem for Linux). 这个是 Windows 近几年开发的新功能, 在 Windows 系统内集成了一个 Linux. 但是目前这个技术还不够成熟. **[暂时不推荐]**.
4. 使用云服务器, 可以直接在 腾讯云, 阿里云或华为云 等服务器厂商处直接购买一个云服务器. **[墙裂推荐]**

如腾讯云阿里云等为在校学生提供了优惠, 只要通过学生认证, 最低可以 10 块钱一个月. 还是非常划算的.

甚至同学们可以 4, 5 个人共用一台服务器, 平均下来一个人一个月 2 块钱.

使用云服务器不仅环境搭建简单, 避免折腾, 同时还有一个最大的好处, 部署在云服务器上的项目可以直接被外网访问到, 这个时候就和一个公司发布一个正式的网站没有任何区别. 也就能让我们自己写的程序真的去给别人去使用.

能够在外网被访问是非常有意义的. 这样我们以后面试的时候就可以提前部署好项目, 现场给面试官演示了.

使用云服务器

我们以腾讯云为例, 其他的服务器厂商也是类似.

注意: 由于腾讯云官网一直在改版, 同学们实际看到的页面可能和课件上略有差别. 如果遇到某个环节搞不定, 随时可以咨询腾讯云的客服小姐姐. (腾讯云的客服是非常热情的, 尤其是在你付钱之前~~).

1. 进入官方网站 <https://cloud.tencent.com/act/campus> (直接在百度上搜 "腾讯云校园计划")
2. 登陆网站(可以使用 qq 或者 微信 登陆)

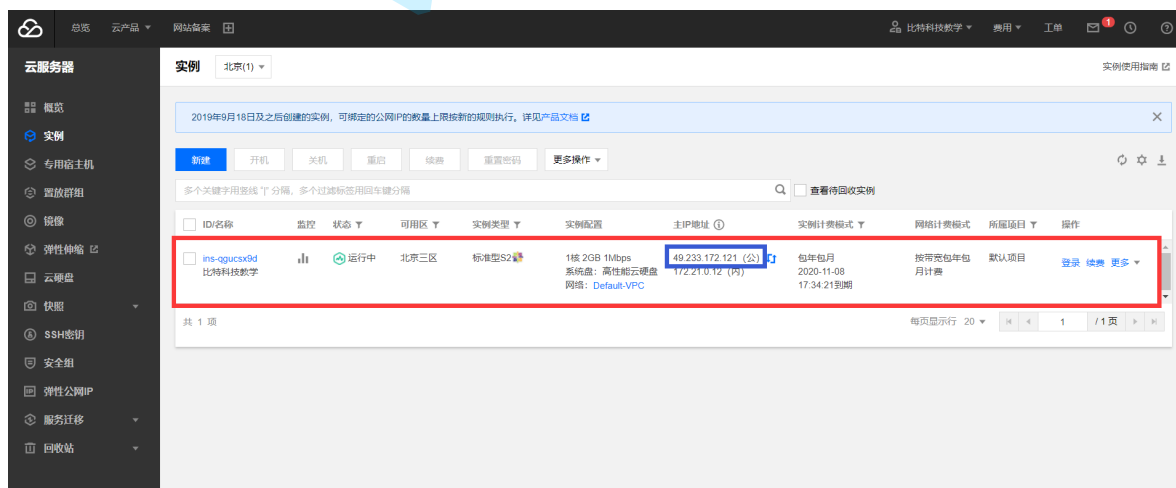
The screenshot shows the 'Campus Plan' (校园优惠套餐) page on the Tencent Cloud website. At the top, there are navigation links: 'Campus Plan' (校园优惠套餐), 'Cloud Practitioner Training and Certification' (云从业者培训及认证), 'Online Learning Center' (在线学习中心), 'Play with AI' (玩转人工智能), and 'Cloud Development Lab' (云开发实验室). Below these, the 'Campus Plan' (校园优惠套餐) is highlighted, with a sub-label '25 years old and below student authentication' (25岁以下免学生认证). The main content area is titled 'Cloud Server' (云服务器) and 'Limited' (限量). It lists specifications: 1 Core CPU (1核 CPU), 2GB Memory (2G 内存), 1Mbps Bandwidth (1Mbps 带宽), and 50GB High Performance Cloud Disk (50GB 高性能云盘). Below the specifications, there are three bullet points: 'The package includes special price cloud servers, domain names (optional), limited to 1 instance' (套餐包含特价云服务器、域名(可选), 限购1台), 'Complete student authentication to obtain 2 purchase continuation qualifications, need to be on this page to enjoy the discount' (完成学生认证获得2次购买价续费资格, 需在本页面优惠续费), and 'Suitable for building, learning, and other lightweight web application scenarios' (适用于建站、学习等轻量级web应用场景). To the right, there are options for 'Activity Location' (活动地域) with 'Guangzhou Area' (广州三区) and 'Beijing Area' (北京三区) buttons, 'Operating System' (操作系统) with a dropdown menu showing 'CentOS 7.6 64bit', and 'Purchase Duration' (购买时长) with buttons for '1 year' (1年), '6 months' (6月), '3 months' (3月), '2 months' (2月), and '1 month' (1月). The price is shown as '120元' (120元) and '4398元' (4398元). A blue button labeled '立即购买' (立即购买) is at the bottom right. A checkbox at the bottom left is labeled 'Add 16 yuan to purchase 1 year .cn domain name usage' (加16元可选购1年.cn域名使用权).

3. 右侧的操作系统选择 CentOS 7.6 64位. 购买时长根据需要选择(建议1年), 点击立即购买即可. 这个步骤需要实名认证, 否则会提示

The screenshot shows a warning message box with the title '温馨提示' (Warm Reminder). The text inside says '抱歉, 您未完成实名认证, 无法参与该活动。' (Sorry, you have not completed real-name authentication, and cannot participate in this activity.). Below the text is a blue button labeled '立即认证' (Authenticate Now).

点击立即认证, 按照系统提示, 完成实名认证即可(认证速度很快).

4. 购买完成后, 可以在控制台中找到自己买的服务器. 点进去能够看到服务器的 IP 地址.



蓝色方框为公网 ip 地址, 稍后我们就会使用这个 ip 登陆服务器.

5. 设置 root 密码: 勾选服务器, 点击重置密码(这个环节可能需要手机短信验证). root 密码建议设置的稍微复杂一些, 否则容易被黑客入侵.



小结:

在这个环节我们最重要的是得到三个信息:

1. 服务器的外网 IP
2. 服务器的管理员账户 (固定为 root)
3. 管理员账户密码(在腾讯云网站上设置的)

这三个信息是我们登陆到 Linux 上的必要条件.

使用终端软件连接到 Linux

什么是终端软件

终端软件是一类工具软件, 可以和远程的主机建立网络连接, 从而对主机进行一些操作.

常见的终端软件:

- XShell
- Putty
- MobaXTerm
- Iterm2
- SecureCRT
-

我们课堂上使用的是 XShell.

下载安装 XShell

下载地址: <https://www.netsarang.com/en/free-for-home-school/>



Free for Non-Commercial Use Only

Your Name (required)

Your Email (required)

☐ Both ☐ Xshell Only ☐ Xftp Only

DOWNLOAD

NOTE: A valid email address is required. A download link will be sent to your email.

XShell 提供了两种授权方式.

- 企业版: 收费.
- 家庭/教育版: 免费.

我们选择 家庭/教育 版即可. 下载的时候需要填一下名字和邮箱. 下载链接会被发送到填写的邮箱中.

Thank you for requesting a download of our software.

A download link will be emailed to the email address you have inputted.

If you are unable to locate it, please make sure our email was not filtered to your Spam folder.

邮箱中收到的内容形如:

Dear user,

Thank you for your interest in Xshell 7. If you did not request a download link for Xshell 7, please contact our support team at support@netsarang.com to have your email address removed from any future emails related to Xshell 7.

Please click the link below to start downloading your software:

<https://www.netsarang.com/en/downloading/?token=N19rYUhBSzFING5XUU9ZUGVSa2l1UUB2U2hWNjRhcTFkZGtnX3J3RJVOEpB>

This link will expire on August 08, 2021

点击邮箱中的下载链接即可完成下载.

使用 XShell 登陆主机

在 XShell 终端下输入

```
ssh 服务器外网ip地址
```

ip 为腾讯云后台页面中看到的外网 IP.

如果网络畅通, 将会提示输入用户名密码. 输入即可正确登陆.

备注: 这里的用户名为 root, 密码是在最初购买服务器的时候设置的密码.

个别同学可能会出现无法输入密码的情况, 解决方案参考 <https://www.cnblogs.com/lemon-le/p/11168609.html>

关于XShell 下的复制粘贴

复制: ctrl + insert (有些同学的 insert 需要配合 fn 来按)

粘贴: shift + insert

ctrl + c / ctrl + v 是不行的.

Linux 常用命令

ls

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
```

pwd

语法: pwd

功能: 显示用户当前所在的目录

举例:

```
pwd
```

cd

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

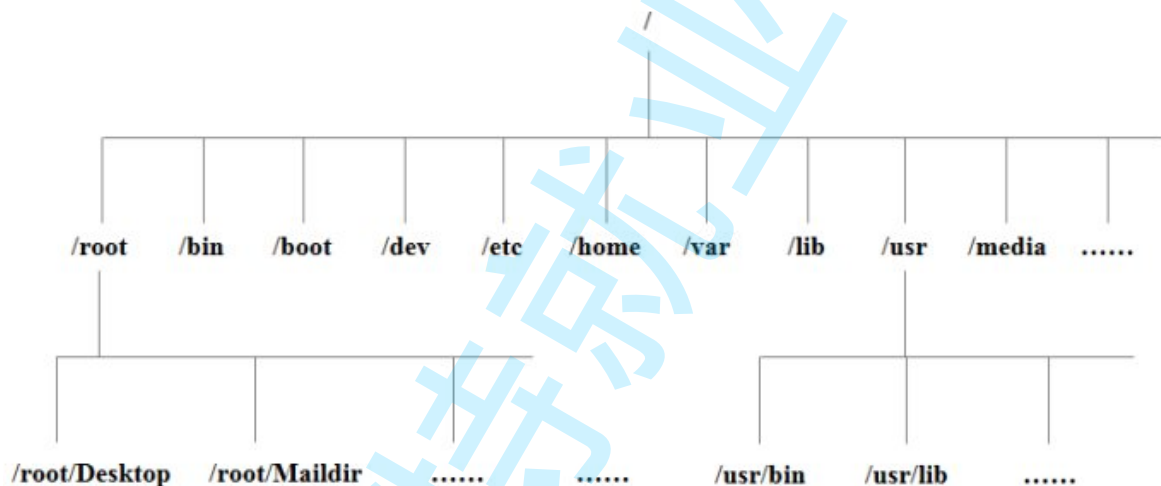
功能: 改变工作目录。将当前工作目录改变到指定的目录下。

举例:

```
cd .. : 返回上级目录
cd ~: 进入用户家目录
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.

touch

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
```

cat

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
```

mkdir

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
```

rm

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.

cp

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
```

mv

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
```

man

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
```

less

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

head

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

tail

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.

date

date 指定格式显示时间: `date +%Y-%m-%d %H:%M:%S"`

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。

vim

vim 是一个知名的文本编辑器. 前面学习的 cat, less, head, tail 等命令只能查看文本, 不能编辑文本. 使用 vim 就可以进行编辑了.

vim 就相当于 Windows 的记事本. 只不过功能比记事本强大一些.

1) 创建文件 / 打开文件

```
vim [文件名]
```

2) 进入插入模式

vim 打开文件后默认是**普通模式**. 普通模式下键盘的按键表示一些特殊功能的快捷键. (例如按下 j 并不是输入字母 "j", 而是表示光标往下移动一行). 需要进入插入模式才能进行文本编辑.

使用 i 键可以进入到**插入模式**. (左下角提示 --INSERT--) 然后就可以像记事本一样正常编辑了.

3) 保存

在插入模式下不能保存文件, 需要先回到 普通模式 . 按下 `ESC` 回到普通模式.

在普通模式下输入 `:w` , 再按下回车, 即可保存文件.

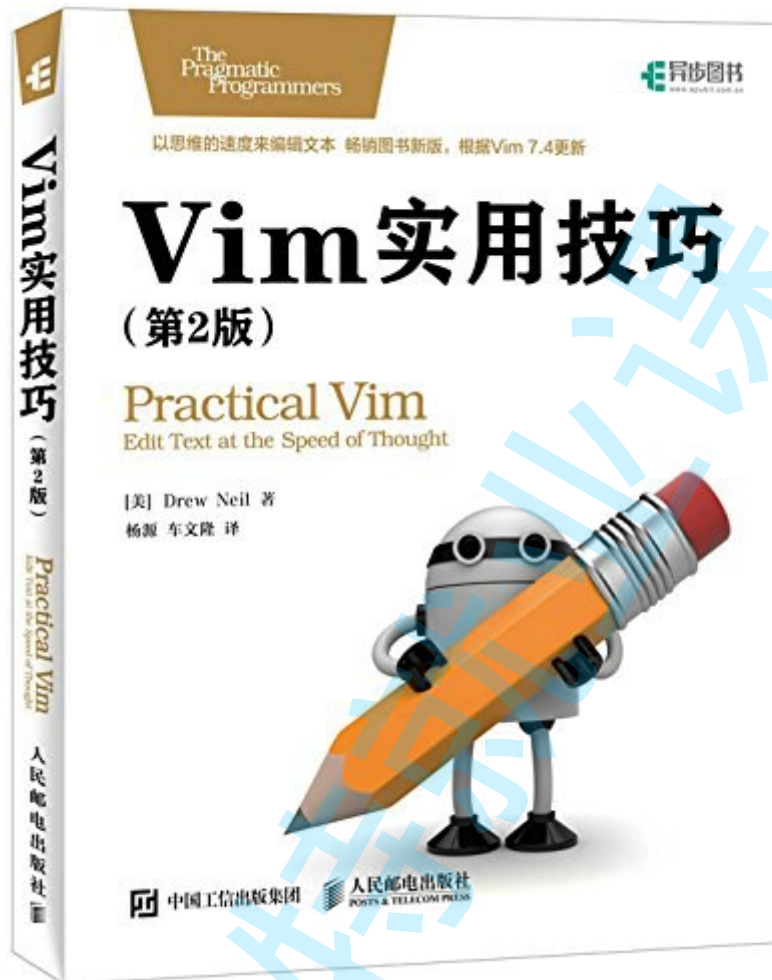
4) 退出

在插入模式下不能退出, 需要先回到 **普通模式**.

在普通模式下输入 `:q`, 再按下回车, 即可退出.

也可以直接使用 `:wq` 同时执行保存和退出.

关于 Vim 的用法还有很多, 此处不做过多介绍了. 有兴趣的同学可以参考 `vimtutor` (直接在终端输入 `vimtutor` 即可进入官方教程) 和 <<Vim 实用技巧>>



熟练使用 Vim 能够大大提高代码编辑效率. 主流的开发工具 (比如 IDEA, VSCode 等) 都支持 Vim 风格的快捷键.

遗憾的是熟练使用 Vim 并不是一件容易的事情, 可能需要长达几年的练习.

grep

语法: `grep [参数]... [文件]...`

功能: 用于查找文件中是否包含指定字符串, 并显示对应的行.

选项:

- `-n<行数>` 显示的行数
- `-w` 全字匹配. 要求整个单词都完全相同的结果才能匹配出来, 而不仅仅是一个单词的一部分.
- `-r` 递归查找. 可以搜索多级目录下的所有文件.
- `--color` 高亮查找到的结果
- `--include` 指定查找某些文件
- `--exclude` 指定排除某些文件

举例:

```
grep "hello" Hello.java
```

ps

语法: ps [参数]...

功能: 用于查看当前系统上运行的进程

选项:

- a 显示一个终端的所有进程
- u 以用户为主的格式来显示程序状况
- x 显示所有程序, 不以终端机来区分

举例:

```
ps aux    # 显示系统上所有的进程
ps aux | grep "进程名"
ps aux | grep "进程id"
```

netstat

语法: netstat [参数]...

功能: 查看系统上的网络状态.

选项:

- -a 显示所有正在或不在侦听的套接字
- -n 显示数字形式地址而不是去解析主机、端口或用户名
- -p 显示套接字所属进程的PID和名称

举例:

```
netstat -anp
netstat -anp | grep "进程名"
netstat -anp | grep "端口号"
```

Linux 权限

用户操作

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 用户权力比较大, 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。

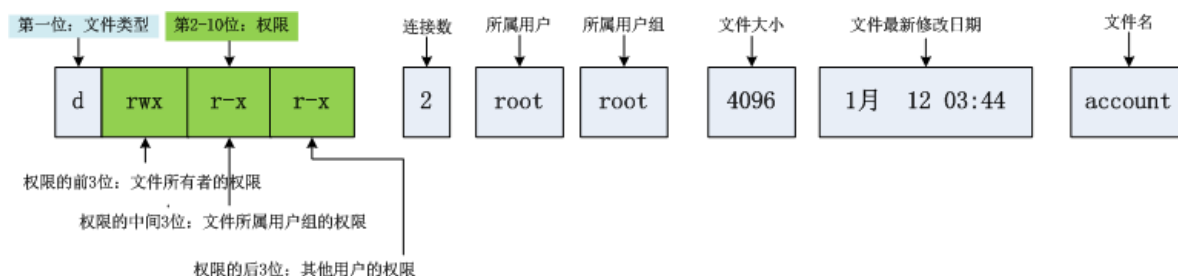
三种角色

访问一个文件的人可能分成三种类别。

- 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用户组: g---Group
- 其它用户: o---Others

文件类型和访问权限

命令 `ll` 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

- 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
- 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
- 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
- "-" 表示不具有该项权限

修改文件权限

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.

搭建 Java 部署环境

yum

认识 yum

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.

包管理器就好比 "应用商店", 我们可以在应用商店上下载一些 app.

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(云服务器一般网络都是良好的).
- 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.

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
```

JDK

```
yum install java-1.8.0-openjdk.x86_64
```

注意: yum上的JDK 是 OpenJDK, 是一个开源版本的JDK, 和 Oracle 官方的JDK 略有差别. 此处我们就使用 OpenJDK 即可. 安装 Oracle JDK 比较麻烦.

使用 `java -version` 验证是否安装成功.

如果提示 "java 命令找不到" 则说明安装失败.

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.

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
```

如果上面的地址过期了, 不能下载, 则可以在官网上, 通过右键 zip -> 复制链接地址 获得.

- Core:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
 - [32-bit Windows zip \(pgp, sha512\)](#)
 - [64-bit Windows zip \(pgp, sha512\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)

2. 解压缩 Tomcat

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

如果 unzip 命令找不到, 则先 `yum install unzip` 即可.

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

Linux 的 Tomcat 目录结构和 Windows 完全一致. (就是同一个 zip 包解压缩出来的)

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

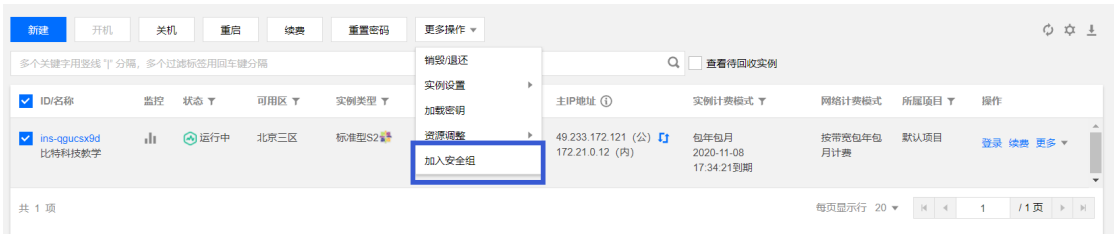
在外网访问这个页面之前需要先开启服务器的 "安全组" 功能.

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



(2) 选择新建安全组



(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

名称

放通全部端口

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.

<input type="checkbox"/>	0.0.0.0/0	TCP:8080	允许	tomcat端口	编辑 插入 删除
--------------------------	-----------	----------	----	----------	--

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>

yum 上的是 MariaDB, 和纯正的 MySQL 还不太一样.

MariaDB 和 MySQL 的关系, 类似于 CentOS 和 RedHat 的关系. 从使用角度上, MariaDB 和 MySQL 之间都是兼容的.

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>

部署 Web 项目到 Linux

0. 什么是部署

工作中涉及到的 "环境"

- **开发环境**: 程序猿写代码用的机器.
- **测试环境**: 测试人员测试程序使用的机器. 一般是退役的部署机器.
- **生产环境**(线上环境): 最终项目发布时所使用的机器. 对稳定性要求很高.

把程序拷贝安装到生产环境上, 这个过程称为 "部署". 也叫 "上线".

这个过程需要拷贝对应的 .class 文件和依赖的资源文件(各种配置文件, html等)

一旦程序部署成功, 那么这个程序就能被外网中千千万万的普通用户访问到.

换句话说, 如果程序有 BUG, 这个 BUG 也就被千千万万的用户看到了.

部署过程至关重要, 属于程序开发中最重要的一环. 一旦部署出现问题, 极有可能导致严重的事故(服务器不可用之类的).

为了防止部署出错, 一般公司内部都有一些自动化部署工具(如 Jenkins 等). 当前我们先使用手工部署的方式来完成部署.

1. 使用 git 传输代码

将本地代码上传到 github / 码云上

过程略, 参考之前的 git 使用过程.

通过 git 下载代码

```
# 进入用户家目录
cd ~
# 创建目录保存项目
mkdir project
cd project
# 使用 git 下载代码
git clone [项目链接]
```

备注:

除了使用 git 来完成代码同步之外, 也可以借助其他工具传输.

同学们可以自行了解 scp 命令和 rz/sz 命令.

2. 数据库建表

按照之前的数据库建表脚本, 在服务器上运行, 建立相同的表结构.

```
mysql -u[数据库用户名] -p[数据库密码] < [数据库脚本].sql
```

3. 构建项目并打包

切换到项目的主目录(包含 pom.xml 的目录), 使用 mvn 打包项目

```
mvn package
```

此时在 target 目录下就会生成对应的 war 包.

曾经我们主要是在 IDEA 中通过图形化工具来使用 mvn. 事实上 mvn 本质是一个基于命令行的工具.

曾经在 IDEA 上进行的 mvn 操作在 Linux 中同样也可以通过命令行来完成.

4. 拷贝到 Tomcat 中

将 war 包拷贝到 Tomcat 的 webapps 目录中.

```
cp [war包名字] [tomcat webapps目录]
```

稍等片刻, Tomcat 会自动解压缩 war 包, 完成部署.

注意: 如果该项目之前部署过, 需要先删除之前的包.

5. 验证

在浏览器上访问项目的链接. (按照项目的功能进行验证).

1. 验证账户注册登录
2. 验证展示博客列表
3. 验证新增博客
4. 验证展示博客内容
5.

注意事项:

- 要开启 Tomcat 对应端口的安全组(8080)

- 代码中的数据库的地址和账号密码可能需要调整
- 还可以把项目链接生成一个二维码(百度搜 "二维码生成"), 就可以用手机扫码访问.

比特就业课