

Table of Contents

Linux介绍、命令	1.1
操作系统	1.1.1
操作系统的发展史	1.1.2
Centos图形界面	1.1.3
文件和目录	1.1.4
常用Linux命令的基本使用	1.1.5
Linux终端命令格式	1.1.6
Linux基本命令(一)	1.1.7
Linux基本命令(二)	1.1.8
编辑器vim	1.1.9

Linux介绍、命令

黑马程序员软件测试

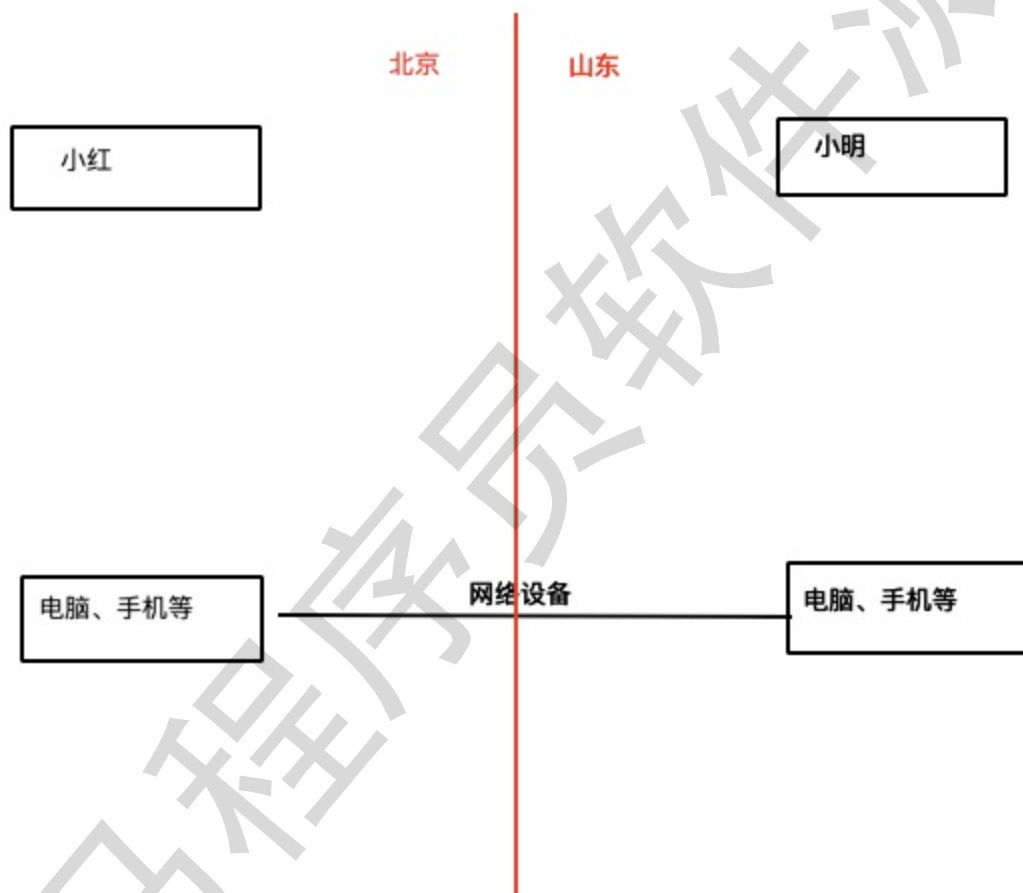
操作系统

目标

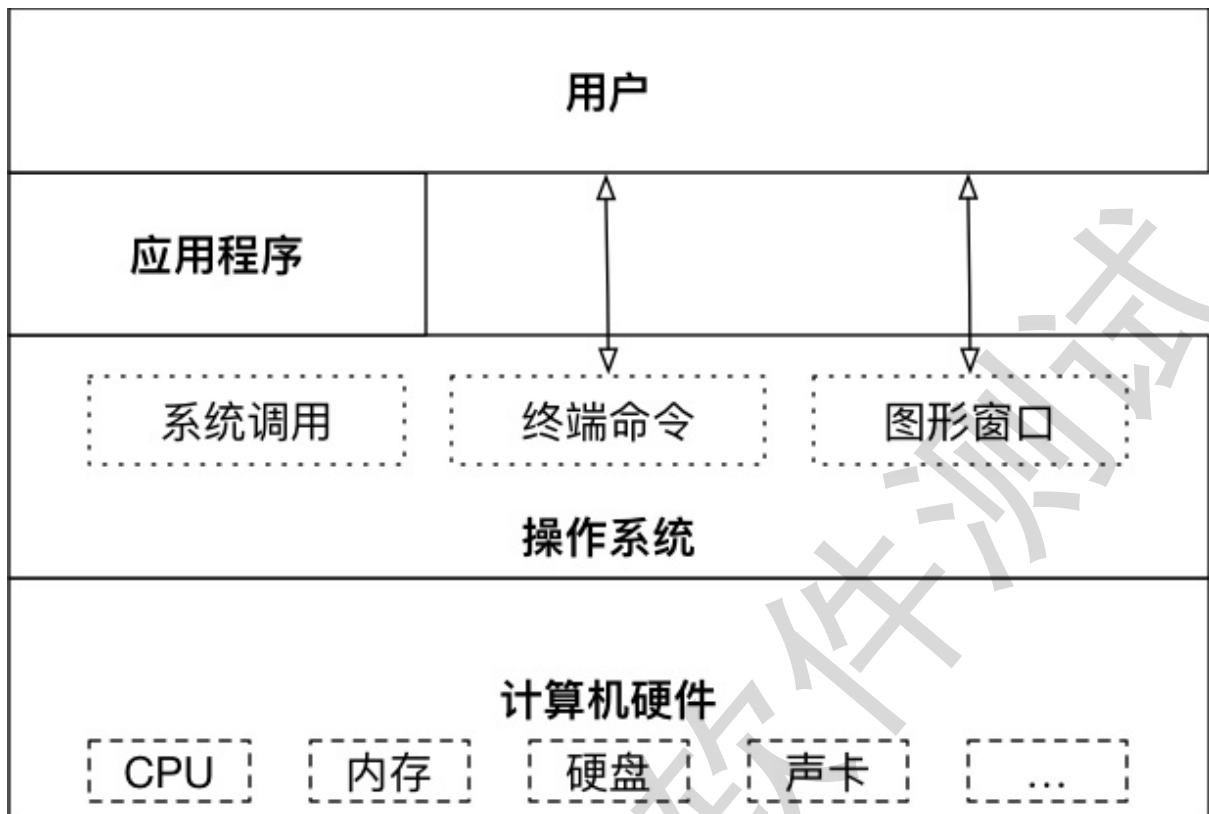
- 了解操作系统及作用

1. 操作系统（Operation System, OS）

一个例子说明操作系统



操作系统作为接口的示意图



没有安装操作系统的计算机，通常被称为裸机

- 如果想在裸机上运行自己所编写的程序，就必须用机器语言书写程序
- 如果计算机上安装了操作系统，就可以在操作系统上安装支持的高级语言环境，用高级语言开发程序

1.1 操作系统的作用

- 是现代计算机系统中最基本和最重要的系统软件
- 是配置在计算机硬件上的第一层软件，是对硬件系统的首次扩展
- 主要作用是管理好硬件设备，并为用户和应用程序提供一个简单的接口，以便于使用
- 而其他的诸如编译程序、数据库管理系统，以及大量的应用软件，都直接依赖于操作系统的支持

1.2 不同应用领域的主流操作系统

- 桌面操作系统
- 服务器操作系统
- 嵌入式操作系统
- 移动设备操作系统

1> 桌面操作系统

- Windows 系列
 - 用户群体大
- macOS
 - 适合于开发人员
- Linux
 - 应用软件少

2> 服务器操作系统

- Linux
 - 安全、稳定、免费
 - 占有率高
- Windows Server
 - 付费
 - 占有率低



3> 嵌入式操作系统

- Linux

4> 移动设备操作系统

- iOS
- Android（基于 Linux）

1.3 虚拟机

虚拟机（Virtual Machine）指通过软件模拟的具有完整硬件系统功能的、运行在一个完全隔离环境中的完整计算机系统

- 虚拟系统通过生成现有操作系统的全新虚拟镜像，具有真实操作系统完全一样的功能
- 进入虚拟系统后，所有操作都是在这个全新的独立的虚拟系统里面进行，可以独立安装运行软件，保存数据，拥有自己的独立桌面，不会对真正的系统产生任何影响
- 而且能够在现有系统与虚拟镜像之间灵活切换的一类操作系统

黑马程序员软件测试

操作系统的发展史

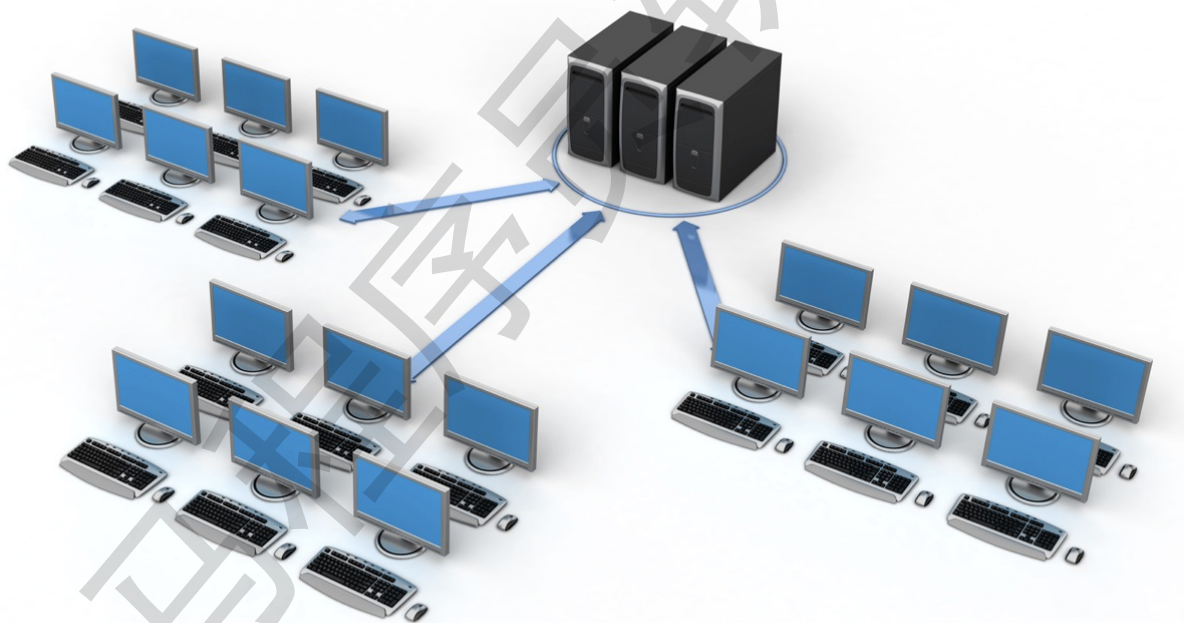
目标

- 了解操作系统的发展历史
- 知道 Linux 内核及发行版的区别
- 知道 Linux 的应用领域

01. 操作系统的发展历史

1.1 Unix

1965 年之前的时候，电脑并不像现在一样普遍，它可不是一般人能碰的起的，除非是军事或者学院的研究机构，而且当时大型主机至多能提供30台终端（30个键盘、显示器），连接一台电脑

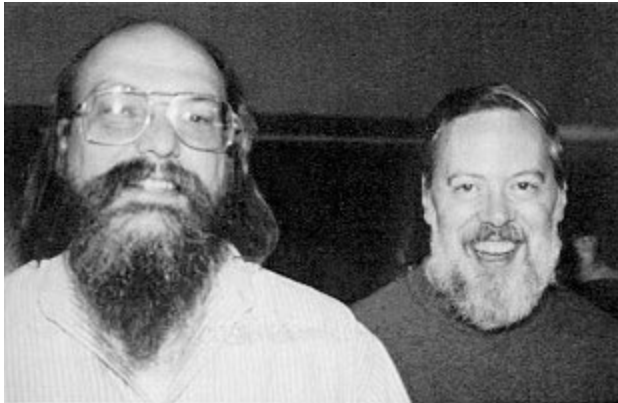


为了解决数量不够用的问题

- 1965 年左右由贝尔实验室加入了麻省理工学院以及通用电气合作的计划 —— 该计划要建立一套多使用者 (**multi-user**)、多任务(**multi-processor**)、多层次(**multi-level**)的**MULTICS**操作系统，想让大型主机支持 300 台终端
- 1969 年前后这个项目进度缓慢，资金短缺，贝尔实验室退出了研究
- 1969 年从这个项目中退出的**Ken Thompson**当时在实验室无聊时，为了让一台空闲的电脑上能够运行 "星际

旅行（SpaceTravel）" 游行，在 8 月份左右趁着其妻子探亲的时间，用了 1 个月的时间，使用汇编写出了 Unix 操作系统的原型

- 1970 年，美国贝尔实验室的**Ken Thompson**，以**BCPL**语言为基础，设计出很简单且很接近硬件的**B** 语言（取BCPL的首字母），并且他用**B** 语言写了第一个 UNIX 操作系统
- 1971 年，同样酷爱 "星际旅行（Space Travel）" 的**Dennis M.Ritchie**为了能早点儿玩上游戏，加入了 **Thompson**的开发项目，合作开发 UNIX，他的主要工作是改造**B** 语言，因为**B** 语言的跨平台性较差
- 1972 年，**Dennis M.Ritchie**在**B** 语言的基础上最终设计出了一种新的语言，他取了**BCPL**的第二个字母作为这种语言的名字，这就是**C** 语言
- 1973 年初，**C** 语言的主体完成，**Thompson**和**Ritchie**迫不及待地开始用它完全重写了现在大名鼎鼎的**Unix** 操作系统



肯·汤普逊（左）和丹尼斯·里奇（右）

1.2 Minix

- 因为**AT&T**（通用电气）的政策改变，在 Version 7 Unix 推出之后，发布新的使用条款，将 UNIX 源代码私有化，在大学中不能再使用 UNIX 源代码
- **Andrew S. Tanenbaum**（塔能鲍姆）教授为了能在课堂上教授学生操作系统运作的细节，决定在不使用任何 AT&T 的源代码前提下，自行开发与 UNIX 兼容的操作系统，以避免版权上的争议
- 以小型 UNIX（mini-UNIX）之意，将它称为**MINIX**

1.3 Linux

- 1991 年林纳斯（**Linus**）就读于赫尔辛基大学期间，对 Unix 产生浓厚兴趣，尝试着在 Minix 上做一些开发工作
- 因为**Minix**只是教学使用，因此功能并不强，林纳斯经常要用他的终端仿真器（**Terminal Emulator**）去访问大学主机上的新闻组和邮件，为了方便读写和下载文件，他自己编写了磁盘驱动程序和文件系统，这些在后来成为了 Linux 第一个内核的雏形，当时，他年仅 21 岁！
- 林纳斯利用 GNU 的 **bash** 当做开发环境，**gcc** 当做编译工具，编写了 Linux 内核，一开始 Linux 并不能兼容 Unix
 - 即 Unix 上跑的应用程序不能在 Linux 上跑，即应用程序与内核之间的接口不一致
 - 一开始 Linux 只适用于 386，后来经过全世界的网友的帮助，最终能够兼容多种硬件



C 语言

- 在把**UNIX**移植到其他类型的计算机上使用时，**C** 语言强大的移植性（Portability）在此显现
 - 机器语言和汇编语言都不具有移植性，为 x86 开发的程序，不可能在 Alpha，SPARC 和 ARM 等机器上运行
- 而**C** 语言程序则可以使用在任意架构的处理器上，只要那种架构的处理器具有对应的 **C** 语言编译器和库，然后将 **C** 源代码编译、连接成目标二进制文件之后即可运行

Unix 家谱

- Mageia
- CentOS
- Arch

03. Linux 的应用领域

3.1 服务器领域

- Linux 在服务器领域的应用是其重要分支
- Linux 免费、稳定、高效等特点在这里得到了很好的体现
 - 早期因为维护、运行等原因同样受到了很大的限制
 - 近些年来 Linux 服务器市场得到了飞速的提升，尤其在高端领域尤为广泛

3.2 嵌入式领域

- 近些年来 Linux 在嵌入式领域的应用得到了飞速的提高
- Linux 运行稳定、对网络的良好支持性、低成本，且可以根据需要进行软件裁剪，内核最小可以达到几百 KB 等特点，使其近些年来在嵌入式领域的应用得到非常大的提高

主要应用：机顶盒、数字电视、网络电话、程控交换机、手机、PDA、等都是其应用领域，得到了 Google、三星、摩托罗拉、NEC 等公司的大力推广



3.3 个人桌面领域

- 此领域是传统 Linux 应用最薄弱的环节
- 传统 Linux 由于界面简单、操作复杂、应用软件少的缺点，一直被 Windows 所压制
- 近些年来随着 Ubuntu、Fedora 等优秀桌面环境的兴起，同时各大硬件厂商对其支持的加大，Linux 在个人桌面领域的占有率在逐渐的提高

黑马程序员软件测试

Centos 图形界面

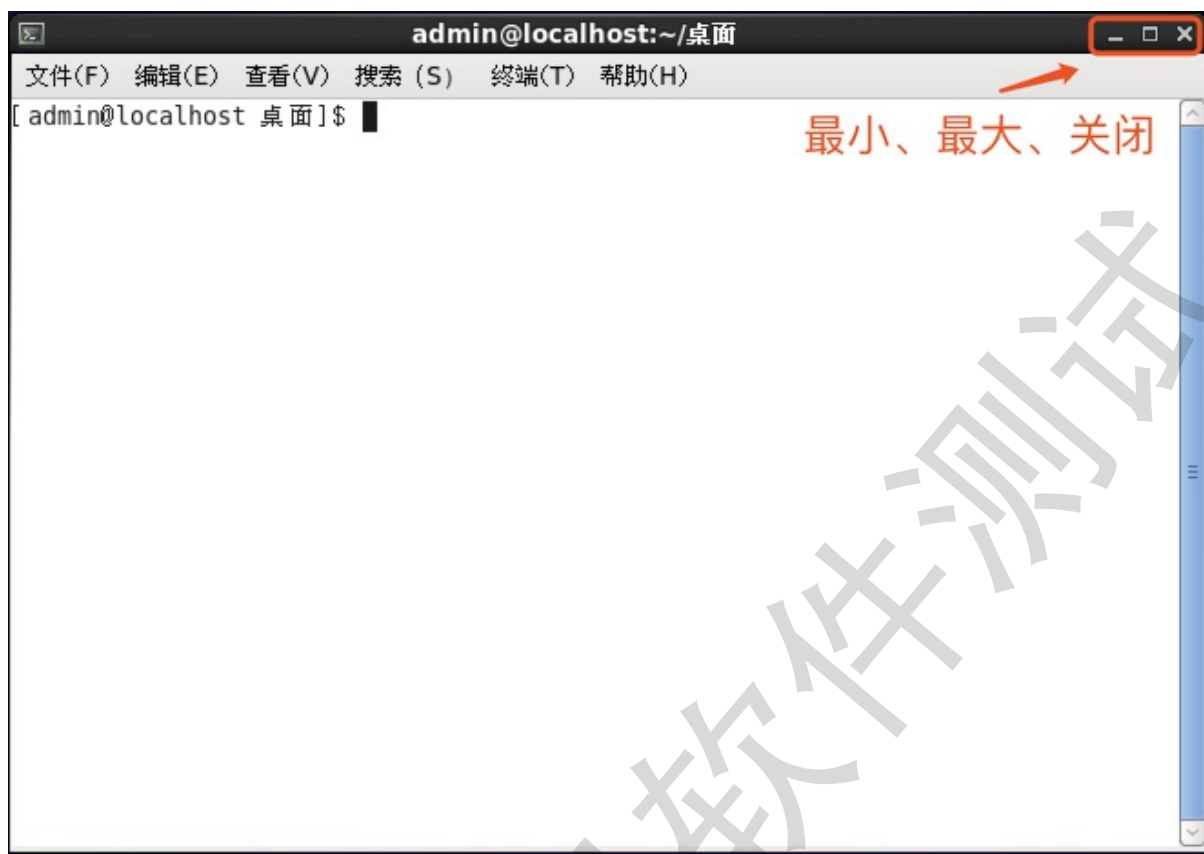
目标

- 熟悉 Centos 图形界面的基本使用

01. Centos 的桌面



02. 窗口操作按钮



文件和目录

目标

- 了解 Linux 文件目录的结构

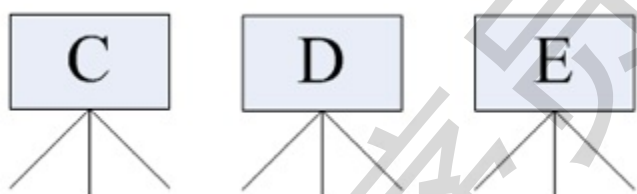
01. Windows 和 Linux 文件系统区别

1.1 Windows 下的文件系统

- 在 Windows 下，打开“计算机”，我们看到的是一个个的驱动器盘符：



- 每个驱动器都有自己的根目录结构，这样形成了多个树并列的情形，如图所示：

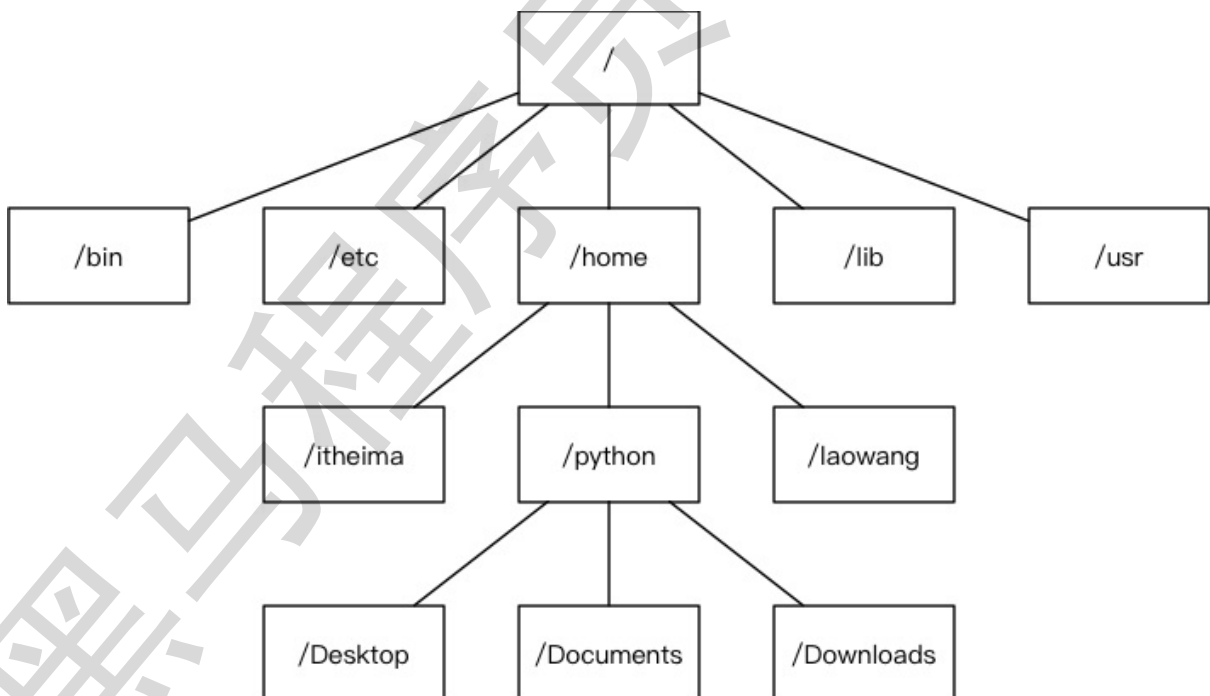


1.2 Linux 下的文件系统

- 在 Linux 下，我们是看不到这些驱动器盘符，我们看到的是文件夹（目录）：



- Linux没有盘符这个概念，只有一个根目录 / ，所有文件都在它下面



1.3 用户目录

位于 `/home/user` ，称之为用户工作目录或家目录，表示方式：

```

/home/user
~

```


1.4 Linux 主要目录速查表

- **/:** 根目录，一般根目录下只存放目录，在 **linux** 下有且只有一个根目录，所有的东西都是从这里开始
 - 当在终端里输入 **/home**，其实是在告诉电脑，先从 **/**（根目录）开始，再进入到 **home** 目录
- **/bin**、**/usr/bin**：可执行二进制文件的目录，如常用的命令 **ls**、**tar**、**mv**、**cat** 等
- **/boot**：放置 **linux** 系统启动时用到的一些文件，如 **linux** 的内核文件：**/boot/vmlinuz**，系统引导管理器：**/boot/grub**
- **/dev**：存放 **linux** 系统下的设备文件，访问该目录下某个文件，相当于访问某个设备，常用的是挂载光驱 **mount /dev/cdrom /mnt**
- **/etc**：系统配置文件存放的目录，不建议在此目录下存放可执行文件，重要的配置文件有
 - **/etc/inittab**
 - **/etc/fstab**
 - **/etc/init.d**
 - **/etc/X11**
 - **/etc/sysconfig**
 - **/etc/xinetd.d**
- **/home**：系统默认的用户家目录，新增用户账号时，用户的家目录都存放在此目录下
 - **~** 表示当前用户的家目录
 - **~edu** 表示用户 **edu** 的家目录
- **/lib**、**/usr/lib**、**/usr/local/lib**：系统使用的函数库的目录，程序在执行过程中，需要调用一些额外的参数时需要函数库的协助
- **/lost+found**：系统异常产生错误时，会将一些遗失的片段放置于此目录下
- **/mnt**、**/media**：光盘默认挂载点，通常光盘挂载于 **/mnt/cdrom** 下，也不一定，可以选择任意位置进行挂载
- **/opt**：给主机额外安装软件所摆放的目录
- **/proc**：此目录的数据都在内存中，如系统核心，外部设备，网络状态，由于数据都存放于内存中，所以不占用磁盘空间，比较重要的文件有：**/proc/cpuinfo**、**/proc/interrupts**、**/proc/dma**、**/proc/ioports**、**/proc/net/*** 等
- **/root**：系统管理员 **root** 的家目录
- **/sbin**、**/usr/sbin**、**/usr/local/sbin**：放置系统管理员使用的可执行命令，如 **fdisk**、**shutdown**、**mount** 等。与 **/bin** 不同的是，这几个目录是给系统管理员 **root** 使用的命令，一般用户只能“查看”而不能设置和使用
- **/tmp**：一般用户或正在执行的程序临时存放文件的目录，任何人都可以访问，重要数据不可放置在此目录下
- **/srv**：服务启动之后需要访问的数据目录，如 **www** 服务需要访问的网页数据存放在 **/srv/www** 内
- **/usr**：应用程序存放目录
 - **/usr/bin**：存放应用程序
 - **/usr/share**：存放共享数据
 - **/usr/lib**：存放不能直接运行的，却是许多程序运行所必需的一些函数库文件
 - **/usr/local**：存放软件升级包
 - **/usr/share/doc**：系统说明文件存放目录
 - **/usr/share/man**：程序说明文件存放目录
- **/var**：放置系统执行过程中经常变化的文件
 - **/var/log**：随时更改的日志文件
 - **/var/spool/mail**：邮件存放的目录
 - **/var/run**：程序或服务启动后，其 **PID** 存放在该目录下

常用 **Linux** 命令的基本使用

目标

- 了解学习 Linux 终端命令的原因
- 常用 Linux 命令体验

01. 学习 **Linux** 终端命令的原因



- 很多人可能在电视或电影中看到过类似的场景，黑客面对一个黑色的屏幕，上面飘着密密麻麻的字符，梆梆一顿敲，就完成了窃取资料的任务。
- Linux 刚出世时没有什么图形界面，所有的操作全靠命令完成，就如同电视里的黑客那样，充满了神秘与晦涩。近几年来，尽管 Linux 发展得非常迅速，图形界面越来越友好，但是在真正的开发过程中，Linux 命令行的应用还是占有非常重要的席位，而且许多Linux功能在命令行界面要比图形化界面下运行的快。可以说不会命令，就不算会 Linux。
- 在职场中，大量的服务器维护工作都是在远程通过SSH 客户端来完成的，并没有图形界面，所有的维护工作都需要通过命令来完成。如磁盘操作、文件存取、目录操作、进程管理、文件权限设定等

- Linux 发行版本的命令大概有 200 多个，但是常用的命令只有 10 多个而已

学习终端命令的技巧：

- 不需要死记硬背，对于常用命令，用的多了，自然就记住了
- 不要尝试一次学会所有的命令，有些命令是非常不常用的，临时遇到，临时百度就可以

02. 常用 Linux 命令的基本使用

序号	命令	对应英文	作用
01	ls	list	查看当前文件夹下的内容
02	pwd	print work directory	查看当前所在文件夹
03	cd [目录名]	change directory	切换文件夹
04	touch [文件名]	touch	如果文件不存在，新建文件
05	mkdir [目录名]	make directory	创建目录
06	rm [文件名]	remove	删除指定的文件名
07	clear	clear	清屏

小技巧

- `ctrl + shift + =` 放大终端窗口的字体显示
- `ctrl + -` 缩小终端窗口的字体显示

03. 自动补全

- 在敲出 文件 / 目录 / 命令 的前几个字母之后，按下 `tab` 键
 - 如果输入的没有歧义，系统会自动补全
 - 如果还存在其他 文件 / 目录 / 命令，再按一下 `tab` 键，系统会提示可能存在的命令

小技巧

- 按 `上 / 下` 光标键可以在曾经使用过的命令之间来回切换
- 如果想要退出选择，并且不想执行当前选中的命令，可以按 `ctrl + c`

Linux 终端命令格式

目标

- 了解终端命令格式
- 知道如何查阅终端命令帮助信息

01. 终端命令格式

```
command [-options] [parameter]
```

说明：

- `command`：命令名，相应功能的英文单词或单词的缩写
- `[-options]`：选项，可用来对命令进行控制，也可以省略
- `parameter`：传给命令的参数，可以是零个、一个或者多个

[] 代表可选

02. 查阅命令帮助信息

提示

- 现阶段只需要知道通过以下两种方式可以查询命令的帮助信息
- 先学习常用命令及常用参数的使用即可，工作中如果遇到问题可以借助网络搜索命令的中文资料

2.1 --help

```
command --help
```

说明：

- 显示 `command` 命令的帮助信息

2.2 man

```
man command
```

说明：

- 查阅 `command` 命令的使用手册

`man` 是 **manual** 的缩写，是 Linux 提供的一个手册，包含了绝大部分的命令、函数的详细使用说明

使用 `man` 时的操作键：

操作键	功能
-----	----

空格键	显示手册页的下一屏
Enter 键	一次滚动手册页的一行
b	回滚一屏
f	前滚一屏
q	退出
/word	搜索 word 字符串

Linux基本命令(一)

目标

- 熟练使用 Linux常用的命令
- ls
- clear
- cd
- pwd
- mkdir
- touch
- rm
- cp
- mv
- chmod
- find
- grep
- 重定向
- 软连接、硬链接
- tar
- shutdown
- reboot
- who
- exit
- passwd

1. 查看文件信息：ls

ls是英文单词list的简写，其功能为列出目录的内容，是用户最常用的命令之一，它类似于DOS下的dir命令。

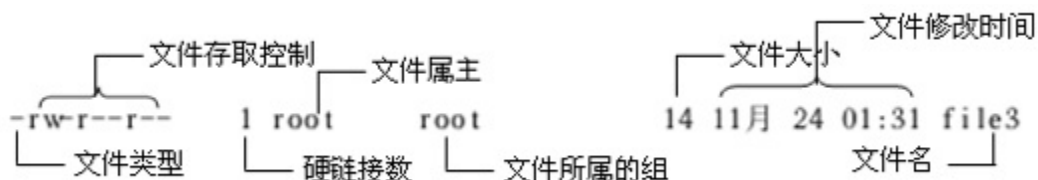
Linux文件或者目录名称最长可以有265个字符，“.”代表当前目录，“..”代表上一级目录，以“.”开头的文件为隐藏文件，需要用 -a 参数才能显示。

```
ls -lah
```

ls常用参数：

参数	含义
-a	显示指定目录下所有子目录与文件，包括隐藏文件
-l	以列表方式显示文件的详细信息
-h	配合 -l 以人性化的方式显示文件大小

详细信息含义如下图所示：



与DOS下的文件操作类似，在Unix/Linux系统中，也同样允许使用特殊字符来同时引用多个文件名，这些特殊字符被称为通配符。

通配符	含义
*	文件代表文件名中所有字符
ls te*	查找以te开头的文件
ls *html	查找结尾为html的文件
?	代表文件名中任意一个字符
ls ?.c	只找第一个字符任意，后缀为.c的文件
ls a.?	只找只有3个字符，前2字符为a.，最后一个字符任意的文件
[]	["和"]将字符组括起来，表示可以匹配字符组中的任意一个。“-”用于表示字符范围。
[abc]	匹配a、b、c中的任意一个
[a-f]	匹配从a到f范围内的的任意一个字符
ls [a-f]*	找到从a到f范围内的的任意一个字符开头的文件
ls a-f	查找文件名为a-f的文件,当“-”处于方括号之外失去通配符的作用
\	如果要使通配符作为普通字符使用，可以在其前面加上转义字符。“?”和“*”处于方括号内时不使用转义字符就失去通配符的作用。
ls *a	查找文件名为*a的文件

2. 清屏：clear

clear作用为清除终端上的显示(类似于DOS的cls清屏功能)，也可使用快捷键：Ctrl + l (“l” 为小写字母“L”)。

3. 切换工作目录：cd

在使用Unix/Linux的时候，经常需要更换工作目录。cd命令可以帮助用户切换工作目录。Linux所有的目录和文件名大小写敏感

cd后面可跟绝对路径，也可以跟相对路径。如果省略目录，则默认切换到当前用户的主目录。

```
cd ~
```

命令	含义
cd	切换到当前用户的主目录(/home/用户目录)，用户登陆的时候，默认的目录就是用户的主目录。
cd ~	切换到当前用户的主目录(/home/用户目录)

<code>cd .</code>	切换到当前目录
<code>cd ..</code>	切换到上级目录
<code>cd -</code>	可进入上次所在的目录

注意：

- 如果路径是从根路径开始的，则路径的前面需要加上“/”，如“/mnt”，通常进入某个目录里的文件夹，前面不用加“/”。

4. 显示当前路径：pwd

使用pwd命令可以显示当前的工作目录，该命令很简单，直接输入pwd即可，后面不带参数。

5. 创建目录：mkdir

通过mkdir命令可以创建一个新的目录。参数-p可递归创建目录。

需要注意的是新建目录的名称不能与当前目录中已有的目录或文件同名，并且目录创建者必须对当前目录具有写权限。

```
mkdir a/b/c -p
```

6. 删除文件：rm

可通过rm删除文件或目录。使用rm命令要小心，因为文件删除后不能恢复。为了防止文件误删，可以在rm后使用-i参数以逐个确认要删除的文件。

```
rm -r abc
```

常用参数及含义如下表所示：

参数	含义
-i	以进行交互式方式执行
-f	强制删除，忽略不存在的文件，无需提示
-r	递归地删除目录下的内容，删除文件夹时必须加此参数

7. 拷贝：cp

cp命令的功能是将给出的文件或目录复制到另一个文件或目录中，相当于DOS下的copy命令。

```
cp 1.txt 2.txt
```

常用选项说明：

选项	含义
-a	该选项通常在复制目录时使用，它保留链接、文件属性，并递归地复制目录，简单而言，保持文件原有属性。

-f	覆盖已经存在的目标文件而不提示
-i	交互式复制，在覆盖目标文件之前将给出提示要求用户确认
-r	若给出的源文件是目录文件，则cp将递归复制该目录下的所有子目录和文件，目标文件必须为一个目录名。
-v	显示拷贝进度

8. mv: 移动、重命名

用户可以使用mv命令来移动文件或目录，也可以给文件或目录重命名。

```
mv 1.txt 2.txt
```

常用选项说明：

选项	含义
-f	禁止交互式操作，如有覆盖也不会给出提示
-i	确认交互方式操作，如果mv操作将导致对已存在的目标文件的覆盖，系统会询问是否重写，要求用户回答以避免误覆盖文件
-v	显示移动进度

9. 创建文件: touch

用户可以通过touch来创建一个空的文件，如下：

```
touch hello.txt
```

说明：

- 则会在当前路径下创建名字为hello.txt的空文件
- Linux系统中没有严格的后缀（格式），所以创建文件时可以命名为任意的文件名

Linux基本命令(二)

1. 输出重定向命令：>

Linux允许将命令执行结果重定向到一个文件，本应显示在终端上的内容保存到指定文件中。

```
ls > test.txt ( test.txt 如果不存在，则创建，存在则覆盖其内容 )
```

注意：>输出重定向会覆盖原来的内容，>>输出重定向则会追加到文件的尾部。

2. 分屏显示：more

查看内容时，在信息过长无法在一屏上显示时，会出现快速滚屏，使得用户无法看清文件的内容，此时可以使用more命令，每次只显示一页，按下空格键可以显示下一页，按下q键退出显示，按下h键可以获取帮助。

```
more test.txt
```

3. 管道：|

管道：一个命令的输出可以通过管道做为另一个命令的输入。

管道我们可以理解现实生活中的管子，管子的一头塞东西进去，另一头取出来，这里“|”的左右分为两端，左端塞东西(写)，右端取东西(读)。

```
ls / | more
```

4. 建立链接文件：ln

Linux链接文件类似于Windows下的快捷方式。

链接文件分为软链接和硬链接。

软链接：软链接不占用磁盘空间，源文件删除则软链接失效。

硬链接：硬链接只能链接普通文件，不能链接目录。

使用格式：

```
硬链接
ln 源文件 链接文件

软链接
ln -s 源文件 链接文件
```

如果没有-s选项代表建立一个硬链接文件，两个文件占用相同大小的硬盘空间，即使删除了源文件，链接文件还是存在，所以-s选项是更常见的形式。

注意：如果软链接文件和源文件不在同一个目录，源文件要使用绝对路径，不能使用相对路径。

5. 查看或者合并文件内容：cat

```
cat test1.txt test2.txt
```

6. 文本搜索：grep

Linux系统中grep命令是一种强大的文本搜索工具，grep允许对文本文件进行模式查找。如果找到匹配模式，grep打印包含模式的所有行。

grep一般格式为：

```
grep [-选项] '搜索内容串' 文件名
```

在grep命令中输入字符串参数时，最好引号或双引号括起来。

例如：

```
grep 'a' 1.txt
```

常用选项说明：

选项	含义
-v	显示不包含匹配文本的所有行（相当于求反）
-n	显示匹配行及行号
-i	忽略大小写

grep搜索内容串可以是正则表达式。

正则表达式是对字符串操作的一种逻辑公式，就是用事先定义好的一些特定字符、及这些特定字符的组合，组成一个“规则字符串”，这个“规则字符串”用来表达对字符串的一种过滤逻辑。

grep常用正则表达式：

参数	含义
^a	行首,搜寻以 a 开头的行；grep -n '^a' 1.txt
ke\$	行尾,搜寻以 ke 结束的行；grep -n 'ke\$' 1.txt
[Ss]igna[Ll]	匹配 [] 里中一系列字符中的一个；搜寻匹配单词signal、signal、Signal、Signal的行；grep -n '[Ss]igna[Ll]' 1.txt
.	(点)匹配一个非换行符的字符；匹配 e 和 e 之间有任何一个字符，可以匹配 eee, eae, eve, 但是不匹配 ee, eaae；grep -n 'e.e' 1.txt

7. 查找文件：find

find命令功能非常强大，通常用来在特定的目录下搜索符合条件的文件，也可以用来搜索特定用户属主的文件。

常用用法：

命令	含义
find . -name test.sh	查找当前目录下所有名为test.sh的文件
find . -name '*.sh'	查找当前目录下所有后缀为.sh的文件

8. 归档管理：tar

计算机中的数据经常需要备份，tar是Unix/Linux中最常用的备份工具，此命令可以把一系列文件归档到一个大文件中，也可以把档案文件解开以恢复数据。

tar使用格式 **tar [参数] 打包文件名 文件**

```
打包
tar cvf a.tar *.txt

解包
tar xvf a.tar -C atar
```

tar命令很特殊，其参数前面可以使用“-”，也可以不使用。

常用参数：

参数	含义
-c	生成档案文件，创建打包文件
-v	列出归档解档的详细过程，显示进度
-f	指定档案文件名称
-t	列出档案中包含的文件
-x	解开档案文件

9. 文件压缩解压：gzip

tar与gzip命令结合使用实现文件打包、压缩。tar只负责打包文件，但不压缩，用gzip压缩tar打包后的文件，其扩展名一般用xxxx.tar.gz。

gzip使用格式如下：

```
gzip [选项] 被压缩文件
```

常用选项：

选项	含义
-d	解压
-r	压缩所有子目录

tar这个命令并没有压缩的功能，它只是一个打包的命令，但是在tar命令中增加一个选项(-z)可以调用gzip实现了一个压缩的功能，实行一个先打包后压缩的过程。

压缩用法：tar cvzf 压缩包包名 文件1 文件2 ...

```
-z : 指定压缩包的格式为: file.tar.gz
```

解压用法：tar zxvf 压缩包包名

```
-z : 指定压缩包的格式为: file.tar.gz
```

解压到指定目录: -C (大写字母“C”)

10. 文件压缩解压: bzip2

tar与bzip2命令结合使用实现文件打包、压缩(用法和gzip一样)。

tar只负责打包文件,但不压缩,用bzip2压缩tar打包后的文件,其扩展名一般用xxxx.tar.gz2。

在tar命令中增加一个选项(-j)可以调用bzip2实现了一个压缩的功能,实行一个先打包后压缩的过程。

压缩用法: tar -jcvf 压缩包包名 文件...(tar jcvf bk.tar.bz2 *.c)

解压用法: tar -jxvf 压缩包包名 (tar jxvf bk.tar.bz2)

11. 文件压缩解压: zip、unzip

通过zip压缩文件的目标文件不需要指定扩展名,默认扩展名为zip。

压缩文件: zip [-r] 目标文件(没有扩展名) 源文件

解压文件: unzip -d 解压后目录文件 压缩文件

12. 查看命令位置: which

13. 修改文件权限: chmod

chmod 修改文件权限有两种使用格式: 字母法与数字法。

字母法: chmod u/g/o/a +/-/= rwx 文件

[u/g/o/a]	含义
u	user 表示该文件的所有者
g	group 表示与该文件的所有者属于同一组(group)者,即用户组
o	other 表示其他以外的人
a	all 表示这三者皆是

[+-=]	含义
+	增加权限
-	撤销权限
=	设定权限

rwx	含义
r	read 表示可读取,对于一个目录,如果没有r权限,那么就意味着不能通过ls查看这个目录的内容。
w	write 表示可写入,对于一个目录,如果没有w权限,那么就意味着不能在目录下创建新的文件。
x	excute 表示可执行,对于一个目录,如果没有x权限,那么就意味着不能通过cd进入这个目录。

如果需要同时进行设定拥有者、同组者以及其他人的权限,参考如下:

数字法: “rwx” 这些权限也可以用数字来代替

字母	说明
r	读取权限，数字代号为 "4"
w	写入权限，数字代号为 "2"
x	执行权限，数字代号为 "1"
-	不具任何权限，数字代号为 "0"

如执行：chmod u=rwx,g=rx,o=r filename 就等同于：chmod u=7,g=5,o=4 filename

chmod 751 file:

- 文件所有者：读、写、执行权限
- 同组用户：读、执行的权限
- 其它用户：执行的权限

注意：如果想递归所有目录加上相同权限，需要加上参数“-R”。如：chmod 777 test/ -R 递归 test 目录下所有文件加 777 权限

14. 切换到管理员账号

Centos下切换到root的简单命令:su

15. 设置用户密码：passwd

在Unix/Linux中，超级用户可以使用passwd命令为普通用户设置或修改用户密码。用户也可以直接使用该命令来修改自己的密码，而无需在命令后面使用用户名。

16. 退出登录账户：exit

- 如果是图形界面，退出当前终端；
- 如果是使用ssh远程登录，退出登陆账户；
- 如果是切换后的登陆用户，退出则返回上一个登陆账号。

17. 查看登录用户：who

who命令用于查看当前所有登录系统的用户信息。

常用选项：

选项	含义
-q或--count	只显示用户的登录账号和登录用户的数量
-u或--heading	显示列标题

18. 关机重启：reboot、shutdown

命令	含义
reboot	重新启动操作系统
shutdown -r now	重新启动操作系统，shutdown会给别的用户提示
shutdown -h now	立刻关机，其中now相当于时间为0的状态

<code>shutdown -h 20:25</code>	系统在今天的20:25 会关机
<code>shutdown -h +10</code>	系统再过十分钟后自动关机

黑马程序员软件测试

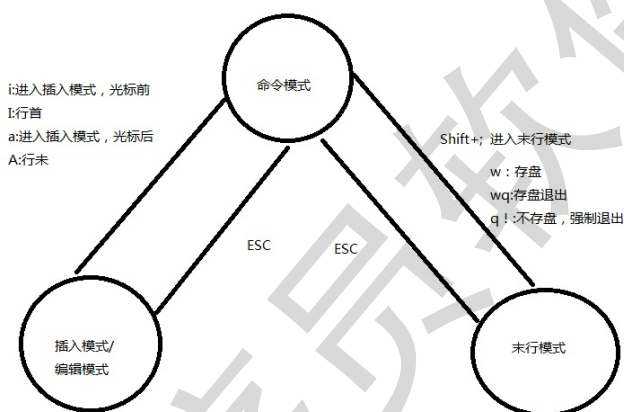
编辑器vim

vi简介

vi是“Visual interface”的简称，它在Linux上的地位就仿佛Edit程序在DOS上一样。它可以执行输出、删除、查找、替换、块操作等众多文本操作，而且用户可以根据自己的需要对其进行定制。Vi不是一个排版程序，它不象Word或WPS那样可以对字体、格式、段落等其他属性进行编排，它只是一个文本编辑程序。vi没有菜单，只有命令，且命令繁多。

Vi有三种基本工作模式：

- + 命令模式
- + 文本输入模式
- + 末行模式



命令行模式

任何时候，不管用户处于何种模式，只要按一下ESC键，即可使Vi进入命令模式；我们在shell环境(提示符为\$)下输入启动Vi命令，进入编辑器时，也是处于该模式下。在该模式下，用户可以输入各种合法的Vi命令，用于管理自己的文档。此时从键盘上输入的任何字符都被当做编辑命令来解释，若输入的字符是合法的Vi命令，则Vi在接受用户命令之后完成相应的动作。但需注意的是，所输入的命令并不在屏幕上显示出来。若输入的字符不是Vi的合法命令，Vi会响铃报警。

文本输入模式

在命令模式下输入插入命令`i`、附加命令`a`、打开命令`o`、修改命令`c`、取代命令`r`或替换命令`s`都可以进入文本输入模式。在该模式下，用户输入的任何字符都被Vi当做文件内容保存起来，并将其显示在屏幕上。在文本输入过程中，若想回到命令模式下，按键ESC即可。

末行模式

末行模式也称**ex**转义模式。在命令模式下，用户按“:”键即可进入末行模式下，此时Vi会在显示窗口的最后一行(通常也是屏幕的最后一行)显示一个“:”作为末行模式的提示符，等待用户输入命令。多数文件管理命令都是在此模式下执行的(如把编辑缓冲区的内容写到文件中)。末行命令执行完后，Vi自动回到命令模式。例如：

```
:sp newfile
```

则分出一个窗口编辑newfile文件。如果要从命令模式转换到编辑模式，可以键入命令**a**或者**i**；如果需要从文本模式返回，则按**Esc**键即可。在命令模式下输入“:”即可切换到末行模式，然后输入命令。

vim基础操作

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

进入插入模式:

```
i: 插入光标前一个字符
I: 插入行首
a: 插入光标后一个字符
A: 插入行末
o: 向下新开一行,插入行首
O: 向上新开一行,插入行首
```

进入命令模式:

ESC:从插入模式或末行模式进入命令模式

移动光标:

```
h: 左移
j: 下移
k: 上移
l: 右移
M: 光标移动到中间行
L: 光标移动到屏幕最后一行行首
G: 移动到指定行,行号 -G
w: 向后一次移动一个字
b: 向前一次移动一个字
{: 按段移动,上移
}: 按段移动,下移
Ctrl-d: 向下翻半屏
Ctrl-u: 向上翻半屏
```

Ctrl-f: 向下翻一屏

Ctrl-b: 向上翻一屏

gg: 光标移动文件开头

G: 光标移动到文件末尾

删除命令:

x: 删除光标后一个字符, 相当于 **Del**

X: 删除光标前一个字符, 相当于 **Backspace**

dd: 删除光标所在行, **n dd** 删除指定的行数 **D**: 删除光标后本行所有内容, 包含光标所在字符

d0: 删除光标前本行所有内容, 不包含光标所在字符

dw: 删除光标开始位置的字符, 包含光标所在字符

撤销命令:

u: 一步一步撤销

Ctrl-r: 反撤销

重复命令:

.: 重复上一次操作的命令

文本行移动:

>>: 文本行右移

<<: 文本行左移

复制粘贴:

yy: 复制当前行, **n yy** 复制 **n** 行

p: 在光标所在位置向下新开辟一行, 粘贴

可视模式:

v: 按字符移动, 选中文本

V: 按行移动, 选中文本可视模式可以配合 **d**, **y**, **>>**, **<<**实现对文本块的删除, 复制, 左右移动

替换操作:

r: 替换当前字符

R: 替换当前行光标后的字符, 直到按**esc**

查找命令:

```
/: str查找  
n: 下一个  
N: 上一个
```