**Linux教程**

Linux 是一套免费使用和自由传播的类 Unix 操作系统，是一个基于 POSIX 和 UNIX 的多用户、多任务、支持多线程和多 CPU 的操作系统。

Linux 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。

|  |  |  |
| --- | --- | --- |
| **比较** | **Windows** | **Linux** |
| 界面 | 界面统一，外壳程序固定所有 Windows 程序菜单几乎一致，快捷键也几乎相同 | 图形界面风格依发布版不同而不同，可能互不兼容。GNU/Linux 的终端机是从 UNIX 传承下来，基本命令和操作方法也几乎一致。 |
| 驱动程序 | 驱动程序丰富，版本更新频繁。默认安装程序里面一般包含有该版本发布时流行的硬件驱动程序，之后所出的新硬件驱动依赖于硬件厂商提供。对于一些老硬件，如果没有了原配的驱动有时很难支持。另外，有时硬件厂商未提供所需版本的 Windows 下的驱动，也会比较头痛。 | 由志愿者开发，由 Linux 核心开发小组发布，很多硬件厂商基于版权考虑并未提供驱动程序，尽管多数无需手动安装，但是涉及安装则相对复杂，使得新用户面对驱动程序问题（是否存在和安装方法）会一筹莫展。但是在开源开发模式下，许多老硬件尽管在Windows下很难支持的也容易找到驱动。HP、Intel、AMD 等硬件厂商逐步不同程度支持开源驱动，问题正在得到缓解。 |
| 使用 | 使用比较简单，容易入门。图形化界面对没有计算机背景知识的用户使用十分有利。 | 图形界面使用简单，容易入门。文字界面，需要学习才能掌握。 |
| 学习 | 系统构造复杂、变化频繁，且知识、技能淘汰快，深入学习困难。 | 系统构造简单、稳定，且知识、技能传承性好，深入学习相对容易。 |
| 软件 | 每一种特定功能可能都需要商业软件的支持，需要购买相应的授权。 | 大部分软件都可以自由获取，同样功能的软件选择较少。 |

**Linux 系统启动过程**

过程可以分为5个阶段：

1. **内核的引导。**

当计算机打开电源后，首先是BIOS开机自检，按照BIOS中设置的启动设备（通常是硬盘）来启动。操作系统接管硬件以后，首先读入 /boot 目录下的内核文件。

1. **运行 init。**

init 进程是系统所有进程的起点，你可以把它比拟成系统所有进程的老祖宗，没有这个进程，系统中任何进程都不会启动。init 程序首先是需要读取配置文件 /etc/inittab，然后启动运行各种程序。Linux允许为不同的场合，分配不同的开机启动程序，这就叫做"运行级别"（runlevel）。也就是说，启动时根据"运行级别"，确定要运行哪些程序。

Linux系统有7个运行级别(runlevel)：

* 运行级别0：系统停机状态，系统默认运行级别不能设为0，否则不能正常启动
* 运行级别1：单用户工作状态，root权限，用于系统维护，禁止远程登录
* 运行级别2：多用户状态(没有NFS)
* 运行级别3：完全的多用户状态(有NFS)，登录后进入控制台命令行模式
* 运行级别4：系统未使用，保留
* 运行级别5：X11控制台，登录后进入图形GUI模式
* 运行级别6：系统正常关闭并重启，默认运行级别不能设为6，否则不能正常启动

1. **系统初始化。**

在init的配置文件中有这么一行： si::sysinit:/etc/rc.d/rc.sysinit　它调用执行了/etc/rc.d/rc.sysinit，而rc.sysinit是一个bash shell的脚本，它主要是完成一些系统初始化的工作，rc.sysinit是每一个运行级别都要首先运行的重要脚本。

它主要完成的工作有：激活交换分区，检查磁盘，加载硬件模块以及其它一些需要优先执行任务。

l5:5:wait:/etc/rc.d/rc 5

1. **建立终端 。**

rc执行完毕后，返回init。这时基本系统环境已经设置好了，各种守护进程也已经启动了。

init接下来会打开6个终端，以便用户登录系统。在inittab中的以下6行就是定义了6个终端：

1:2345:respawn:/sbin/mingetty tty1

2:2345:respawn:/sbin/mingetty tty2

3:2345:respawn:/sbin/mingetty tty3

4:2345:respawn:/sbin/mingetty tty4

5:2345:respawn:/sbin/mingetty tty5

6:2345:respawn:/sbin/mingetty tty6

从上面可以看出在2、3、4、5的运行级别中都将以respawn方式运行mingetty程序，mingetty程序能打开终端、设置模式。

同时它会显示一个文本登录界面，这个界面就是我们经常看到的登录界面，在这个登录界面中会提示用户输入用户名，而用户输入的用户将作为参数传给login程序来验证用户的身份。

1. **用户登录系统。**

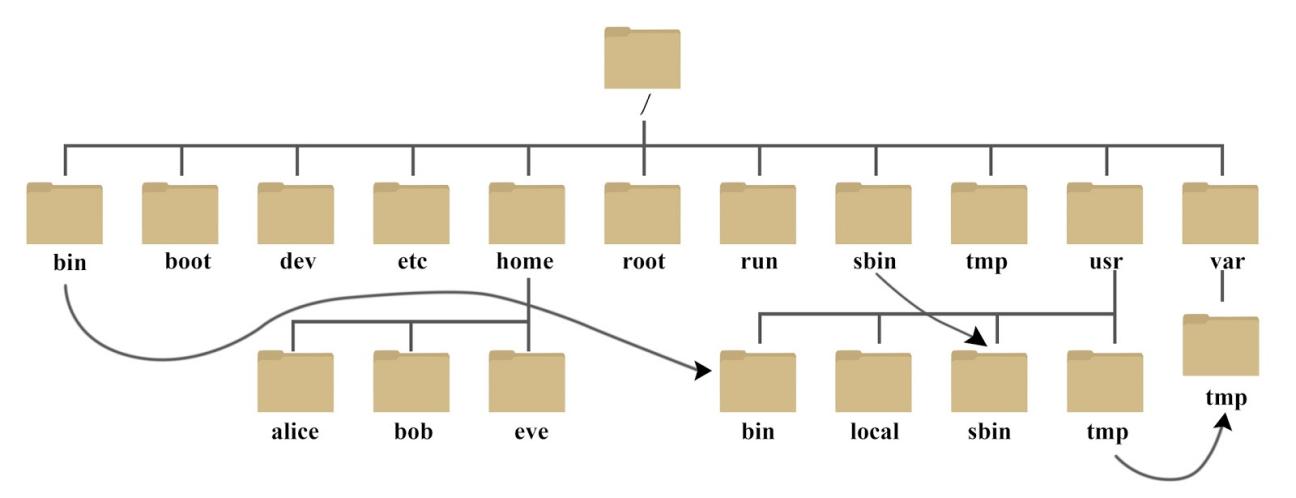
一般来说，用户的登录方式有三种：

（1）命令行登录

（2）ssh登录

（3）图形界面登录

**Linux 系统目录结构**



登录系统后，在当前命令窗口下输入命令：

ls /

你会看到如下图所示:

4_20

以下是对这些目录的解释：

* /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。
* /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，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，如上图中的 alice、bob 和 eve。
* /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：
* 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 system resources(unix 系统资源) 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。

在 Linux 系统中，有几个目录是比较重要的，平时需要注意不要误删除或者随意更改内部文件。

/etc： 上边也提到了，这个是系统中的配置文件，如果你更改了该目录下的某个文件可能会导致系统不能启动。

/bin, /sbin, /usr/bin, /usr/sbin: 这是系统预设的执行文件的放置目录，比如 ls 就是在 /bin/ls 目录下的。

/var： 这是一个非常重要的目录，系统上跑了很多程序，那么每个程序都会有相应的日志产生，而这些日志就被记录到这个目录下，具体在 /var/log 目录下，另外 mail 的预设放置也是在这里。

**Linux 文件基本属性**

Linux 系统是一种典型的多用户系统，不同的用户处于不同的地位，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。

在 Linux 中我们通常使用以下两个命令来修改文件或目录的所属用户与权限：

* chown (change owner) ： 修改所属用户与组。
* chmod (change mode) ： 修改用户的权限。

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，如：

[root@www /]# ls -l

total 64

dr-xr-xr-x 2 root root 4096 Dec 14 2012 bin

dr-xr-xr-x 4 root root 4096 Apr 19 2012 boot

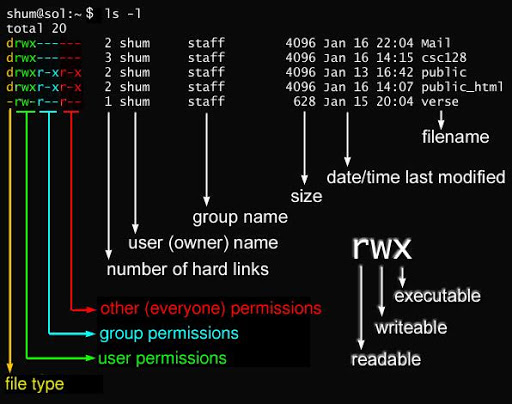
……

实例中，bin 文件的第一个属性用 d 表示。d 在 Linux 中代表该文件是一个目录文件。

在 Linux 中第一个字符代表这个文件是目录、文件或链接文件等等。

* 当为 d 则是目录
* 当为 - 则是文件；
* 若是 l 则表示为链接文档(link file)；
* 若是 b 则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 则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)。

接下来的字符中，以三个为一组，且均为 rwx 的三个参数的组合。其中， r 代表可读(read)、 w 代表可写(write)、 x 代表可执行(execute)。 要注意的是，这三个权限的位置不会改变，如果没有权限，就会出现减号 - 而已。



更改文件属性

**1、chgrp：更改文件属组**

语法：

chgrp [-R] 属组名 文件名

参数选项

* -R：递归更改文件属组，就是在更改某个目录文件的属组时，如果加上 -R 的参数，那么该目录下的所有文件的属组都会更改。

**2、chown：更改文件所有者（owner），也可以同时更改文件所属组。**

语法：

chown [–R] 所有者 文件名

chown [-R] 所有者:属组名 文件名

进入 /root 目录（~）将install.log的拥有者改为bin这个账号：

[root@www ~] cd ~

[root@www ~]# chown bin install.log

[root@www ~]# ls -l

-rw-r--r-- 1 bin users 68495 Jun 25 08:53 install.log

将install.log的拥有者与群组改回为root：

[root@www ~]# chown root:root install.log

[root@www ~]# ls -l

-rw-r--r-- 1 root root 68495 Jun 25 08:53 install.log

**3、chmod：更改文件9个属性**

Linux文件属性有两种设置方法，一种是数字，一种是符号。

**用数字修改：**Linux 文件的基本权限就有九个，分别是 owner/group/others(拥有者/组/其他) 三种身份各有自己的 read/write/execute 权限。先复习一下刚刚上面提到的数据：文件的权限字符为： -rwxrwxrwx ， 这九个权限是三个三个一组的！其中，我们可以使用数字来代表各个权限，各权限的分数对照表如下：

r:4

w:2

x:1

每种身份(owner/group/others)各自的三个权限(r/w/x)分数是需要累加的，例如当权限为： -rwxrwx--- 分数则是：

owner = rwx = 4+2+1 = 7

group = rwx = 4+2+1 = 7

others= --- = 0+0+0 = 0

所以等一下我们设定权限的变更时，该文件的权限数字就是 770。变更权限的指令 chmod 的语法是这样的：

chmod [-R] xyz 文件或目录

选项与参数：

* xyz : 就是刚刚提到的数字类型的权限属性，为 rwx 属性数值的相加。
* -R : 进行递归(recursive)的持续变更，以及连同次目录下的所有文件都会变更

举例来说，如果要将 .bashrc 这个文件所有的权限都设定启用，那么命令如下：

[root@www ~]# ls -al .bashrc

-rw-r--r-- 1 root root 395 Jul 4 11:45 .bashrc

[root@www ~]# chmod 777 .bashrc

[root@www ~]# ls -al .bashrc

-rwxrwxrwx 1 root root 395 Jul 4 11:45 .bashrc

那如果要将权限变成 -rwxr-xr-- 呢？那么权限的分数就成为 [4+2+1][4+0+1][4+0+0]=754。

**用符号修改：**还有一个改变权限的方法，从之前的介绍中我们可以发现，基本上就九个权限分别是：

* user：用户
* group：组
* others：其他

那么我们就可以使用 u, g, o 来代表三种身份的权限。

此外， a 则代表 all，即全部的身份。读写的权限可以写成 r, w, x，也就是可以使用下表的方式来看：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| chmod | u g o a | +(加入) -(除去) =(设定) | r w x | 文件或目录 |

比如，如果我们需要将文件权限设置为 -rwxr-xr-- ，可以使用下面的语法来设定：

chmod u=rwx,g=rx,o=r 文件名

**Linux 文件与目录管理**

我们知道 Linux 的目录结构为树状结构，最顶级的目录为根目录 / 。

其他目录通过挂载可以将它们添加到树中，通过解除挂载可以移除它们。

* 绝对路径：路径的写法，由根目录 / 写起，例如： /usr/share/doc 这个目录。
* 相对路径：路径的写法，不是由 / 写起，例如由 /usr/share/doc 要到 /usr/share/man 底下时，可以写成： cd ../man 这就是相对路径的写法。

处理目录的常用命令：

* ls（英文全拼：list files）: 列出目录及文件名

选项与参数：

-a ：列出全部文件，包括隐藏文件(开头为 . 的文件)(常用)

-d ：仅列出目录名本身，而不进入目录显示其内容(常用)

-l ：列出文件详细信息，包含文件的属性与权限等等数据；(常用)

* cd（英文全拼：change directory）：切换目录
* pwd（英文全拼：print work directory）：显示目前的目录

选项与参数：

-P ：显示出真实路径，而非使用链接 (link) 路径，也就是链接真正指向的文件

* mkdir（英文全拼：make directory）：创建一个新的目录

选项与参数：

-m ：手动指定这个目录的权限，如果没有会以默认权限 (umask) 自动配置

-p ：逐层创建目录(包含上一级目录)，比如a/b/c...

* rmdir（英文全拼：remove directory）：删除一个空的目录

选项与参数：

-p ：从该目录起，一次性删除多级空目录

注意！！如果任一目录有其他文件（夹），则会报错

* cp（英文全拼：copy file）: 复制文件或目录

选项与参数：

-a ：相当于 -pdr 的组合使用， pdr 请参考下列说明(常用)

-d ：若来源档为链接档的属性(link file)，则复制链接档属性而非文件本身

-f ：为强制(force)的意思，会强制覆盖已有文件（不询问），即使为只读文件

-i ：与-f相反，若目标文件已经存在，在覆盖时会先询问(常用)

-l ：创建一个硬链接（hard link），而非复制文件本身

注意！！两个文件实际上是“同一文件”的两个名字，共享同一个 inode 和数据块

1. 修改 A，B 会跟着变（因为是同一数据），相当于java的引用传递
2. 删除其中一个，另一个仍然可用
3. 节省空间
4. 只能用于同一文件系统
5. 不能用于目录（除非超级用户）

-p ：连同文件的属性（修改时间，权限等）一起复制，而非使用默认属性(备份常用)

-r ：将文件夹下的所有文件递归复制，用于目录（文件夹）的复制行为(常用)

-s ：复制成为符号链接档 (symbolic link)，即『捷径』文件；

-u ：只复制源文件比目标文件新，或者不存在的文件，也就是“补全（update）”

* rm（英文全拼：remove）: 删除文件或目录

选项与参数：

-f ：就是 force 的意思，忽略不存在的文件，不会出现警告信息

-i ：互动模式，在删除前会询问使用者

-r ：递归删除，将一个目录下的文件（夹）逐级删除

* mv（英文全拼：move file）: 移动文件与目录，或修改文件与目录的名称

选项与参数：

-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 ：若目标文件已经存在，会询问是否覆盖

-u ：当目标文件不存在，或者源文件比较新，才会覆盖 (update)

Linux 文件内容查看

* cat 由第一行开始显示文件内容

-A ：相当於 -vET 的整合选项，可列出一些特殊字符而不是空白

-b ：列出行号，仅针对非空白行做行号显示，空白行不标行号

-E ：将结尾的断行字节 $ 显示出来；

-n ：列印出行号，连同空白行也会有行号，与 -b 的选项不同；

-T ：将 [tab] 按键以 ^I 显示出来；

-v ：列出一些看不出来的特殊字符

* tac 从最后一行开始显示，可以看出 tac 是 cat 的倒着写
* nl 显示的时候，输出行号

-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)；

-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 ；

* more 一页一页的显示文件内容
* less 与 more 类似，但是比 more 更好的是，他可以往前翻页！
* head 只看头几行

-n ：后面接数字，代表显示几行的意思

* tail 只看尾巴几行

-n ：后面接数字，代表显示几行的意思

-f ：表示持续侦测后面所接的档名，要等到按下[ctrl]-c才会结束tail的侦测

**Linux 用户和用户组管理**

1. 添加新的用户账号

useradd 选项 用户名

-c comment 指定一段注释性描述。

-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。

-g 用户组 指定用户所属的用户组。

-G 用户组，用户组 指定用户所属的附加组。

-s Shell文件 指定用户的登录Shell。

-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。

1. 删除帐号

userdel 选项 用户名

1. 修改帐号

userdel 选项 用户名

1. 用户口令的管理

passwd 选项 用户名

-l 锁定口令，即禁用账号。

-u 口令解锁。

-d 使账号无口令。

-f 强迫用户下次登录时修改口令。

**Linux 磁盘管理**

**df命令**

检查文件系统的磁盘空间占用情况。可以利用该命令来获取硬盘被占用了多少空间，目前还剩下多少空间等信息，df 是在问：“这个分区还剩多少空间？”

选项与参数：

* -h：以人类可读的方式显示输出结果（例如，使用 KB、MB、GB 等单位）。
* -T：显示文件系统的类型。
* -t <文件系统类型>：只显示指定类型的文件系统。
* -i：显示 inode 使用情况。
* -H：该参数是 -h 的变体，但是使用 1000 字节作为基本单位而不是 1024 字节。这意味着它会以 SI（国际单位制）单位（例如 MB、GB）而不是二进制单位（例如 MiB、GiB）来显示磁盘使用情况。
* -k：这个选项会以 KB 作为单位显示磁盘空间使用情况。
* -a：该参数将显示所有的文件系统，包括虚拟文件系统，例如 proc、sysfs 等。如果没有使用该选项，默认情况下，df 命令不会显示虚拟文件系统。

**du命令**

也是查看使用空间的，但是与 df 命令不同的是 Linux du 命令是对文件和目录磁盘使用的空间的查看，du 是在问：“这个目录或文件用了多少空间？”

选项与参数：

* -a ：列出所有的文件与目录容量，因为默认仅统计目录底下的文件量而已。
* -h ：以人们较易读的容量格式 (G/M) 显示；
* -s ：仅显示指定目录或文件的总大小，而不显示其子目录的大小。
* -S ：包括子目录下的总计，与 -s 有点差别。
* -k ：以 KBytes 列出容量显示；
* -m ：以 MBytes 列出容量显示

| **命令** | **全称** | **作用对象** | **显示内容** | **常见用途** |
| --- | --- | --- | --- | --- |
| df | disk free | **整个文件系统** | 查看磁盘**总体使用情况** | 查看磁盘剩余空间 |
| du | disk usage | **文件或目录** | 查看文件/目录**实际占用空间** | 查看哪个文件/目录占空间最多 |

**Linux vi/vim命令**

Vim 是从 vi 发展出来的一个文本编辑器。代码补全、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。

简单的来说， vi 是老式的字处理器，不过功能已经很齐全了，但是还是有可以进步的地方。 vim 则可以说是程序开发者的一项很好用的工具。

**vi/vim 的使用**

基本上 vi/vim 共分为三种模式，命令模式（Command Mode）、输入模式（Insert Mode）和命令行模式（Command-Line Mode）。

**命令模式**

启动 vi/vim，便进入了命令模式。

此状态下敲击键盘动作会被 Vim 识别为命令，而非输入字符，比如我们此时按下 i，并不会输入一个字符，i 被当作了一个命令。以下是普通模式常用的几个命令：

* i -- 切换到输入模式，在光标当前位置开始输入文本。
* x -- 删除当前光标所在处的字符。
* : -- 切换到底线命令模式，以在最底一行输入命令。
* a -- 进入插入模式，在光标下一个位置开始输入文本。
* o：在当前行的下方插入一个新行，并进入插入模式。
* O -- 在当前行的上方插入一个新行，并进入插入模式。
* dd -- 剪切当前行。
* yy -- 复制当前行。
* p（小写） -- 粘贴剪贴板内容到光标下方。
* P（大写）-- 粘贴剪贴板内容到光标上方。
* u -- 撤销上一次操作。
* Ctrl + r -- 重做上一次撤销的操作。
* :w -- 保存文件。（:是进入底线命令行模式）
* :q -- 退出 Vim 编辑器。
* :q! -- 强制退出Vim 编辑器，不保存修改。

命令模式只有一些最基本的命令，因此仍要依靠底线命令行模式输入更多命令。

**输入模式**

在命令模式下按下 i 就进入了输入模式，使用 Esc 键可以返回到普通模式。

在输入模式中，可以使用以下按键：

* 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。底线命令模式可以输入单个或多个字符的命令，可用的命令非常多。在底线命令模式中，基本的命令有（已经省略了冒号）：

* :w：保存文件。
* :q：退出 Vim 编辑器。
* :wq：保存文件并退出 Vim 编辑器。
* :q!：强制退出Vim编辑器，不保存修改。

**Linux yum 命令**

yum（ Yellow dog Updater, Modified）是一个在 Fedora 和 RedHat 以及 SUSE 中的 Shell 前端软件包管理器。基于 RPM 包管理，能够从指定的服务器自动下载 RPM 包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软件包，无须繁琐地一次次下载、安装。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。

yum常用命令：

* 列出所有可更新的软件清单命令：yum check-update
* 更新所有软件命令：yum update
* 仅安装指定的软件命令：yum install <package\_name>
* 仅更新指定的软件命令：yum update <package\_name>
* 列出所有可安裝的软件清单命令：yum list
* 删除软件包命令：yum remove <package\_name>
* 查找软件包命令：yum search <keyword>
* 清除缓存命令:

yum clean packages: 清除缓存目录下的软件包

yum clean headers: 清除缓存目录下的 headers

yum clean oldheaders: 清除缓存目录下旧的 headers

yum clean, yum clean all (= yum clean packages; yum clean oldheaders) :清除缓存目录下的软件包及旧的 headers

**Linux apt 命令**

apt（Advanced Packaging Tool）是一个在 Debian 和 Ubuntu 中的 Shell 前端软件包管理器。apt 命令提供了查找、安装、升级、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。

apt 命令执行需要超级管理员权限(root)。

apt 常用命令：

* 列出所有可更新的软件清单命令：sudo apt update
* 升级软件包：sudo apt upgrade
* 列出可更新的软件包及版本信息：apt list --upgradable
* 升级软件包，升级前先删除需要更新软件包：sudo apt full-upgrade
* 安装指定的软件命令：sudo apt install <package\_name>
* 安装多个软件包：sudo apt install <package\_1> <package\_2> <package\_3>
* 更新指定的软件命令：sudo apt update <package\_name>
* 显示软件包具体信息,例如：版本号，安装大小，依赖关系等等：sudo apt show <package\_name>
* 删除软件包命令：sudo apt remove <package\_name>
* 清理不再使用的依赖和库文件: sudo apt autoremove
* 移除软件包及配置文件: sudo apt purge <package\_name>
* 查找软件包命令： sudo apt search <keyword>
* 列出所有已安装的包：apt list --installed
* 列出所有已安装的包的版本信息：apt list --all-versions