

Linux 操作系统

Tedu Python 教学部

Author: 吕泽

Linux 操作系统

[Linux操作系统简介](#)

[Linux 操作系统结构](#)

[文件系统结构](#)

[主要目录功能](#)

[绝对路径和相对路径](#)

[环境变量](#)

[编译器使用](#)

[vi使用](#)

[Vi常用命令](#)

[shell命令](#)

[基础命令](#)

[压缩管理](#)

[软件管理](#)

[用户管理](#)

[ssh服务](#)

[Linux下的SSH服务](#)

[ssh命令](#)

[ssh登录远程主机](#)

[scp命令](#)

[ssh密钥](#)

[生产ssh密钥步骤](#)

[shell脚本介绍](#)

Linux操作系统简介

1. 什么是操作系统

操作系统（OS）是管理计算机硬件与软件资源的计算机程序，同时也是计算机系统的内核与基石。操作系统需要处理如管理与配置内存、决定系统资源供需的优先次序、控制输入设备与输出设备、操作网络与管理文件系统等基本事务。操作系统也提供一个让用户与系统交互的操作界面。

2. Linux简介

Linux是一套免费使用和自由传播的类Unix操作系统，是一个多用户、多任务、支持多线程和多CPU的操作系统。伴随着互联网的发展，Linux得到了来自全世界软件爱好者、组织、公司的支持。它除了在服务器操作系统方面保持着强劲的发展势头以外，在个人电脑、嵌入式系统上都有着长足的进步。

Linux不仅系统性能稳定，而且是开源软件。其核心防火墙组件性能高效、配置简单，保证了系统的安全。在很多企业网络中，为了追求速度和安全，Linux操作系统不仅仅是被网络运维人员当作服务器使用。

Linux与其他操作系统相比，具有开放源码、没有版权、技术社区用户多等特点，开放源码使得用户可以自由裁剪，灵活性高，功能强大，成本低。



3. Linux特点

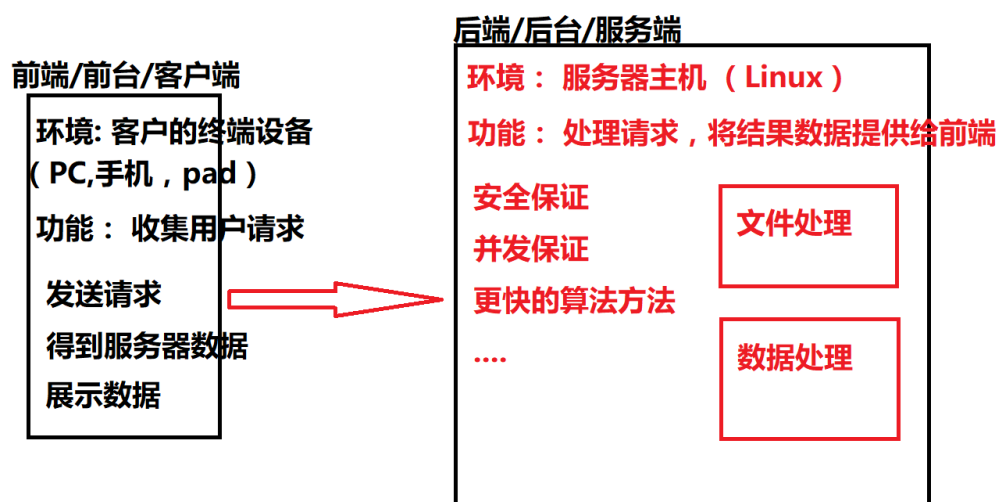
- 一切都是文件
- Linux是一款免费的操作系统
- Linux支持多用户，多任务
- 支持多种平台
- 具有非常强大的网络功能
- 简化开发过程，较强的移植性

4. 发行版本

严格的来讲，Linux 只是一个系统内核，即计算机软件与硬件通讯之间的平台。一些组织或厂家将Linux 内核与GNU软件（系统软件 and 工具）整合起来，并提供一些安装界面和系统设定与管理工具，这样就构成了一个发型套件，目前市面上较知名的发行版有：Ubuntu、RedHat、CentOS、Debian、Fedora、SuSE、OpenSUSE、Arch Linux、SolusOS 等。

5. Linux应用领域

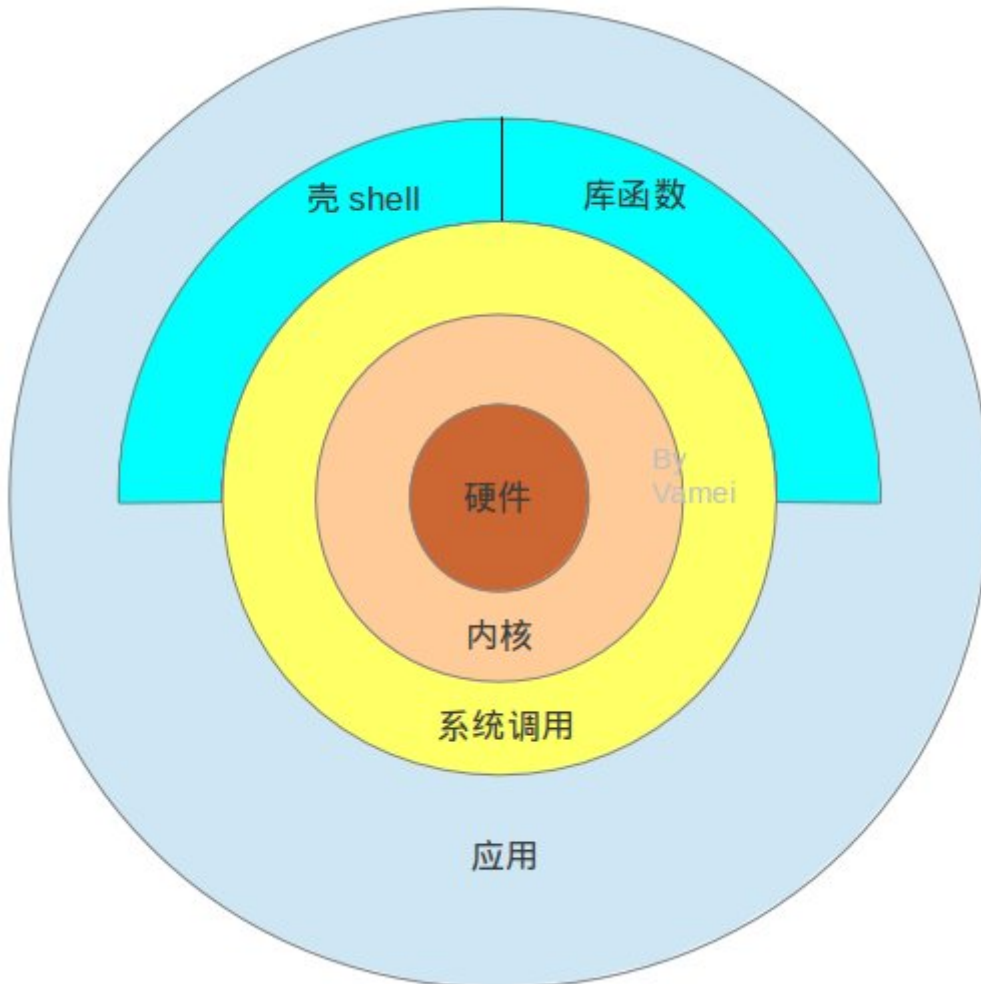
- Linux 服务器：目前Linux服务器是当前最广泛的应用。
- 桌面应用:新版本的Linux系统特别在桌面应用方面进行了改进，达到相当的水平
- 嵌入式系统：由于Linux系统开放源代码，功能强大、可靠、稳定性强、灵活而且具有极大的伸缩性，因此，在嵌入式应用的领域里LINUX操作系统都有很广阔的应用市场。



Linux 操作系统结构

一个典型的Linux操作系统组成为：Linux内核，文件系统，命令行shell，图形界面和桌面环境，并包含各种工具和应用软件。

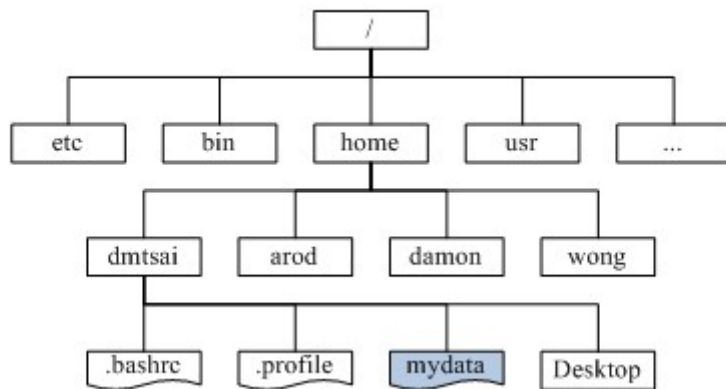
- Linux内核: Linux操作系统的核心代码，是Linux系统的最底层，提供了系统的核心功能并允许进程以一种有序的方式访问硬件。
- 文件系统：通常指称管理磁盘数据的系统，可将数据以目录或文件的型式存储。每个文件系统都有自己的特殊格式与功能
- shell命令：Shell是一个命令解释器，它使得用户能够与操作系统进行交互，负责接收用户命令，然后调用相应的应用程序，并根据用户输入的指令来反馈给用户指定的信息。
- 应用软件：包含桌面系统和基础的软件操作工具等。



文件系统结构

在任何一个操作系统中，文件系统都是其重要的组件，用于组织和管理计算机存储设备上的大量文件，并提供用户交互接口。Linux同样具备完善的文件系统。Linux的文件组织模式犹如一颗倒置的树，所有存储设备作为这颗树的一个子目录。将根（/）作为整个文件系统的唯一起点，其他所有目录都从该点出发。

- linux下的目录结构



主要目录功能

1. /bin目录

/bin目录包含了引导启动所需的命令或普通用户可能用的命令(可能在引导启动后)。这些命令都是二进制文件的可执行程序(bin是binary----二进制的简称)，多是系统中重要的系统文件。

2. /sbin目录

/sbin目录类似/bin，也用于存储二进制文件。因为其中的大部分文件多是系统管理员使用的基本的系统程序，所以虽然普通用户必要且允许时可以使用，但一般不给普通用户使用。

3. /etc目录

/etc目录存放着各种系统配置文件，其中包括了用户信息文件/etc/passwd，系统初始化文件/etc/rc等。linux正是因为这些文件才得以正常地运行。

4. /root目录

/root目录是超级用户的目录。

5. /lib目录

/lib目录是根文件系统上的程序所需的共享库，存放了根文件系统程序运行所需的共享文件。这些文件包含了可被许多程序共享的代码，以避免每个程序都包含有相同的子程序的副本，故可以使得可执行文件变得更小，节省空间。

6. /dev目录

/dev目录存放了设备文件，即设备驱动程序，用户通过这些文件访问外部设备。比如，用户可以通过访问/dev/mouse来访问鼠标的输入，就像访问其他文件一样。

7. /usr文件系统

/usr是个很重要的目录，通常这一文件系统很大，因为所有程序安装在这里。本地安装的程序和其他东西在/usr/local下，因为这样可以在升级新版系统或新发行版时无须重新安装全部程序。

8. /var文件系统

/var包含系统一般运行时要改变的数据。通常这些数据所在的目录的大小是要经常变化或扩充的。

9. /home

普通用户的默认目录，在该目录下，每个用户拥有一个以用户名命名的文件夹。

绝对路径和相对路径

绝对路径(肯定以/开头)和相对路径都是在Linux中用来描述一个文件位置的描述方法

绝对路径：指文件在文件系统中以根目录为起始点的准确位置描述。例如“/usr/bin/gnct”就是绝对路径。

相对路径：指相对于用户当前位置为起始点，对一个文件位置的逐层描述。例如，用户处在usr目录中时，只需要“games/gnct”就可确定这个文件。在相对路径描述时，. 表示当前目录，.. 表示上一级目录。

环境变量

环境变量一般是指在操作系统中用来指定操作系统运行环境的一些参数，比如系统目录位置等。如果将一个目录设置为环境变量目录，那么这个目录下的文件在任何目录下都可以查找到。

- 临时设置环境变量： `export PATH="$PATH:/your_dir_path"`
- 查看环境变量目录：

```
tarena@tedu:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
```

- 永久设置环境变量：修改etc下的profile文件，输入vi /etc/profile，将上述语句添加进去

举例：

将hello.py文件变成环境变量：

- 将/home/tarena/下的hello.py文件copy到usr/下面：

```
tarena@tedu:~$ vi hello.py #通过vi命令添加内容： print("hello world")
tarena@tedu:~$ sudo cp hello.py /bin #成功复制到/bin目录下面
[sudo] tarena 的密码：
tarena@tedu:~$ hello.py
-bash: /bin/hello.py: 权限不够
tarena@tedu:/bin$ sudo chmod 777 hello.py #添加执行权限
tarena@tedu:/bin$ ls hello.py
hello.py
tarena@tedu:/bin$ vi hello.py #首行添加！ #/usr/bin/env/ python3
tarena@tedu:/bin$ hello.py
hello world #此时bin下面的hello.py文件可正常运行
#此时在其他任何目录下面皆可运行hello.py文件。如下：
tarena@tedu:~$ hello.py #前面都不用加python3
hello world
```

编译器使用

vi使用

Vi是一个著名的功能强大的文本编辑器，它是Linux中最基本的编辑器，虽然不像图形界面编辑器那样美观，没有使用鼠标那样方便。但vi编辑器在系统管理、服务器配置工作中永远都是无可替代的。

插入模式：用来输入文本内容，在这种模式下摁 `esc` 键进入命令模式

命令模式：命令操作，命令模式又包含底行命令，即在vi底部命令行输入命令。在命令模式下键入 ':' 即进入底行命令输入

Vi常用命令

操作命令

- i 在当前字符的左边插入
- o 在当前行下面插入一个新行
- h 向前移动一个字符
- j 向下移动一行
- k 向上移动一行
- l 向后移动一个字符
- b 移动到当前单词的开始
- e 移动到当前单词的结尾
- w 移动到下一个单词
- 0: 光标移动到行首
- \$: 光标移动到行末
- G: 光标移动到文件结尾
- gg: 光标移动到文件开头
- yy 复制当前一行
- dd 剪切当前一行
- p 粘贴内容到光标之后
- P 将剪贴板中的内容粘贴在光标前
- x 删除光标所在字符
- r 修改光标所在字符, r后接着要修正的字符
- R 进入取代状态, 新增资料会覆改原先资料, 直到按ESC回到指令模式下为止
- s 删除光标所在字符, 并进入输入模式
- S 删除光标所在的行, 并进入输入模式
- cw 删除光标所在的后一单词进入插入模式
- cb 删除光标所在的前一单词进入插入模式
- c\$ 删除自光标位置至行尾的所有字符, 进入插入模式
- c0 删除自光标位置至行首的所有字符, 进入插入模式
- cc 删除当前行进入插入模式
- u 撤销操作
- ctrl-r 反撤销

底行命令

- :n 跳转到底n行
- :w 保存
- :w filename 存入文件filename 中
- :w! filename 强制性存入文件filename 中
- :q 退出
- :q! 强行退出
- :set nu 显示行号
- /word 查找内容
- : %s/hello/nihao/g 将所有的hello 替换为nihao
- : 5,8s/hello/nihao/g 将5-8行的hello 替换为nihao

vi
Cheat Sheet

You're in command mode when you start **vi**.

Pressing **ESC** always takes you back to command mode.

Exiting **vi**...be in command mode! Press **Enter** to end these commands.

:q quit unless save needed

:q! quit, forget changes

wq write file then quit

Cursor Movement Commands

beginning of line

previous word

b ← **h** ← **j** ← **k** → **l** → next word

l lowercase ell

end of line

^ caret

\$

Most commands can be preceded with a repeat count.

5w move ahead 5 words

16k move up 16 lines

1G go to line 1

76G go to line 76

G go to end of file

ctrl G where am I?

vi only lets you move through text that exists. To extend a line, append spaces. To extend the file, append empty lines.

Changing Commands...can include any cursor movement command

c\$ change to end of line

d5w delete 5 words

dG delete to end of file

dd delete current line

X delete this character

U undo last change

U restore line

Add-text Commands...press **ESC** when you're done

a append here

A append at end of line

i insert here

I insert at start of line

R overstrike

capital i

shell命令

我们在Linux终端命令行下输入的命令称为shell命令，是通过shell解释器解释后由系统执行。shell解释器也是一代一代进化的，我们现在所使用的shell是 bash shell。在服务器系统操作维护中，shell命令起到了很重要的作用，因为服务器操作往往没有图形界面，或者通过远程命令控制完成，这时候shell命令可以帮助我们实现很多功能。

shell命令基本格式有三部分,每部分之间有空格，选项与选项，参数与参数之间也有空格。（多个选项可以写在一起）：

命令 [选项] [参数]---命令的操作对象

举例：

基础命令

作用	命令及常用选项
关闭重启系统	shutdown
临时管理员权限	sudo
帮助说明	man
退出终端	exit
清空屏幕	clear
查看文件绝对路径	pwd
切换工作目录	cd
查看文件	ls , ls -l , ls -a
打印内容	echo
创建目录	mkdir -p
删除目录	rmdir
创建文件	touch
复制文件	cp -r
移动文件	mv
删除文件	rm -rf
查找文件	find
查看文件信息	file
查看文件内容	cat
查看文件前 n 行	head
查看文件后n行	tail
对比文件差异	diff
查询文件内容信息	wc
查找文件里符合条件的字符串	grep
修改文件权限	chmod
创建链接文件	ln , ln -s
显示当前时间	date
查看磁盘使用情况	df
显示当前用户	whoami
查找服务程序位置	which
查找命令文件及关联目录	whereis

- shutdown：默认情况为1分钟后关机，+n 表示n分钟后关机，在这个过程中shutdown -c 可以取消。shutdown -h now表示立即关机

```
#进入root权限
tarena@tedu:~$ sudo su
[sudo] tarena 的密码:
root@tedu:/home/tarena#
#退出
root@tedu:/home/tarena# exit
exit
tarena@tedu:~$
```

- sudo：在命令前书写，表示已以管理员权限运行这个命令。
- man：使用man命令可以找到特定的联机帮助页，并提供简短的命令说明。
- exit：关闭一个终端，通过ctrl+alt+t ,shift+ctrl+t ,shift+ctrl+n 都可以快速打开一个终端
- clear：等同于ctrl-l，清空屏幕
- pwd：pwd命令用于显示用户在文件系统中的当前位置。
- cd：用于改变工作目录，参数为绝对路径或者相对路径。
- ls：查看目录下的文件列表 -l 展示详细信息，-a展示隐藏文件（Linux下 . 开头的为隐藏文件）

```
d  rwxr-xr-x  2  linux  linux   4096   Sep 22  2012   Desktop
```

各项含义分别是：文件类型，文件的权限，文件的链接数，文件的所有者，文件所在组，文件的大小，文件的更新时间，文件名

- echo：标准输出一段文字在显示器上，-n表示输出后不换行

输入/输出重定向是改变Shell命令或程序默认的标准输入/输出目标，重新定向到新的目标。

重定向符	含义	实例
> file	将file文件重定向为输出源，新建模式	echo "hello world" > out.txt, 将执行结果，写到out.txt文件中，若有同名文件将被删除
>> file	将file文件重定向为输出源，追加模式	ls /usr >> Lsoutput.txt, 将ls /usr的执行结果，追加到Lsoutput.txt文件已有内容后
< file	将file文件重定向为输入源	wc < file1, 将file1中的内容作为输入传给wc命令

```
#输出重定向-->将ls -l的内容输出到file文件里面，并覆盖原来的文件内容
tarena@tedu:~$ ls -l > file
tarena@tedu:~$ vi file
```

```
#追加重定向，不会覆盖原来文件原有的内容
tarena@tedu:~$ ls -a >> file
tarena@tedu:~$ vi file
```

- `mkdir`：创建一个目录，`-p`用户创建多层目录
- `rmdir`：删除一个空目录
- `touch`：创建一个文件
- `cp`：复制文件/文件另存，`-r`可以复制文件目录
- `mv`：移动文件/文件重命名
- `rm`：删除文件，`-rf`用于删除目录

当需要用命令处理一组文件，例如file1.txt、file2.txt、file3.txt……，用户不必一一输入文件名，可以使用Shell通配符。

通配符	含义	实例
星号 (*)	匹配任意长度的字符串	用file_*.txt，匹配file_wang.txt、file_Lee.txt、file3_Liu.txt
问号 (?)	匹配一个长度的字符	用file_?.txt，匹配file_1.txt、file1_2.txt、file_3.txt
方括号 ([...])	匹配其中指定的一个字符	用file_[otr].txt，匹配file_o.txt、file_r.txt和file_t.txt
方括号 ([-])	匹配指定的一个字符范围	用file_[a-z].txt，匹配file_a.txt、file_b.txt，直到file_z.txt
方括号 ([^...])	除了其中指定的字符，均可匹配	用file_[^otr].txt，除了file_o.txt、file_r.txt和file_t.txt的其他文件

- `find`：在一个目录下查找文件 `find path -name` 按照名字查找
- `file`：查看一个文件的属性类型
- `cat`：查看一个文件内容
- `head`：`head -n file` 用来查看一个文件前n行
- `tail`：`tail -n file` 用来查看一个文件的后n行
- `wc`：查看文件内容信息，`-c`表示查看多少字符，`-l`查看多少行，`-w`查看多少单词
- `grep`：用于查找内容包含指定的范本样式的文件，`grep`指令会把含有范本样式的那一行显示出来。若不指定任何文件名称，则`grep`指令会从标准输入中读取数据。`-n`用于显示行号，`-i`忽略大小写

管道可以把一系列命令连接起来，意味着第一个命令的输出将作为第二个命令的输入，通过管道传递给第二个命令，第二个命令的输出又将作为第三个命令的输入，以此类推。

```
ls | grep 'test'
```

- `chmod`：`chmod [权限声明] [file]` 修改一个文件权限

`u`表示该文件的拥有者，`g`表示与该文件的拥有者属于同一个群体(group)者，`o` (others)表示其他以外的人，`a(all)`表示这三者皆是。

`+`表示增加权限、`-`表示取消权限、`=`表示唯一设定权限。

`r`表示可读取，`w`表示可写入，`x`表示可执行，`X`表示只有当该文件是个子目录或者该文件已经被设定过为可执行。

也可以用三位8进制表达权限

- ln：ln [源文件] [目标文件] 创建链接，-s表示创建软链接，不加表示硬链接。硬链接想相当于副本，源文件名改变也不会影响链接，源文件删除则链接断开。软链接相当于快捷方式，源文件改变则链接断开，源文件删除重新创建后链接可以恢复。
- date：显示时间
- df：显示目前在Linux系统上的文件系统的磁盘使用情况统计,其中-T显示文件系统类型 ext4的为磁盘，其他的为虚拟文件。-h以M为单位显示
- whoami：显示当前用户
- which：查找一个程序位置
- whereis：查找一个命令文件及其关联目录

压缩管理

归档和压缩可以帮助我们很好的整理文件，在Linux系统下支持多种压缩格式。其中gzip是Linux中最流行的压缩工具，具有很好的移植性，可在很多不同架构的系统中使用。bzip2在性能上优于gzip，提供了最大限度的压缩比率。如果用户需要经常在Linux和微软Windows间交换文件，建议使用zip。

作用	命令及常用选项
压缩解压zip格式	zip -r, unzip
压缩解压gz格式	gzip, gunzip
压缩解压bz2格式	bzip2,bunzip2
归档压缩解压文件	tar

- zip：将文件压缩为zip格式
 - `zip test.zip filelist`
- unzip：将zip格式文件解压
 - `unzip test.zip`
- gzip：将文件压缩为gz格式
 - `gzip file`
- gunzip：将gz文件解压
 - `gunzip file.gz`
- bzip2：将文件压缩为bz2格式
 - `bzip2 file`
- bunzip2：将bz2 文件解压
 - `bunzip2 file.bz2`
- tar：主要用于将若干文件或目录合并为一个文件，以便备份和压缩。-cjf 用于压缩bz2格式文件，-czf用于压缩gz格式文件，-xvf用于解压文件

选项	描述
-t	显示归档文件中的内容
-x	释放归档文件
-u	更新归档文件
-c	创建一个新的归档文件
-v	显示归档和释放的过程信息
-f	用户指定归档文件的文件名，否则使用默认名称
-j	由tar生成归档，然后由bzip2压缩
-z	由tar生成归档，然后由gzip压缩

```
tar -czf file.tar.gz file1 file2
tar -xvf file.tar.gz
```

软件管理

Linux下安装的软件包是 deb格式软件包。Deb软件包将二进制应用程序、配置文档、帮助页面都整合在一个文件中，便于传递、安装、升级和备份。dpkg是最早的Deb包管理工具，使用dpkg可以实现软件包的安装、编译、卸载、查询，以及应用程序打包等功能。但是由于当时Linux系统规模和Internet网络条件的限制，没有考虑到操作系统中软件包存在如此复杂的依赖关系。因而，为了解决软件包依赖性问题 and 获取问题，就出现了APT工具。

作用	命令
本地包安装	dpkg -i
本地安装包卸载	dpkg -r
本地包状态查看	dpkg -s
本地包卸载并清除记录	dpkg -P
升级软件包	apt-get update
安装软件	apt-get install
卸载软件	apt-get remove --purge
下载源码包	apt-get source
删除缓存的软件安装包	apt-get clean

用户管理

Linux系统是一个多用户多任务的操作系统，任何一个要使用系统资源的用户，都必须首先向系统管理员申请一个账号，然后以这个账号的身份进入系统。用户的账号可以帮助用户组织文件，并为用户提供安全性保护。每个用户账号都拥有一个惟一的用户名和各自的口令。用户在登录时键入正确的用户名和口令后，就能够进入系统和自己的主目录。

实现用户账号的管理，要完成的工作主要有如下几个方面：

- 用户账号的添加、删除与修改。
- 用户口令的管理。
- 用户组的管理。

作用	命令及常用选项
增加用户	adduser
设置用户密码	passwd
设置用户权限	usermod
删除用户	deluser
删除用户组	delgroup
修改文件归属	chown

- adduser：增加一个用户，--home 生产主目录 --ingroup选择用户组名称。默认组名和用户名相同

```
tarena@tedu:~$ sudo adduser --home /home/wb wb
[sudo] tarena 的密码:
adduser: "wb"组已经存在。
```

- passwd：修改一个用户的密码
- usermod: 修改用户权限，如密码或则所属的组等
- deluser: 删除一个用户 --remove-home 表示同时删除用户主目录
- delgroup：删除一个用户组
- chown：修改文件的所属用户

添加用户后为保证用户有sudo 权限，需修改 /etc/sudoers 文件，找到下面一行，在root下面添加一行，假设aid为用户名，如下所示：

```
## Allow root to run any commands anywhere
root ALL=(ALL) ALL
aid ALL=(ALL) ALL
```

通过sudo vipw命令可以查看用户信息，在信息列表中绝大多数为虚拟用户。当添加一个用户时该文件会自动修改，如果删除一个用户则需要自己修改这个文件用户才彻底删除

ssh服务

ssh是一种安全协议，主要用于给远程登录会话数据进行加密，保证数据传输的安全。在数据传输方面有很多应用。

Linux下的SSH服务

在Linux下SSH服务端是一个在后台运行的程序，响应来自客户端的连接请求。SSH服务端的进程名为sshd，负责实时监听远程SSH客户端的远程连接请求，并进行处理。

- 安装：sudo apt-get install openssh-server

- 查看ssh服务状态： `ps -e|grep ssh`

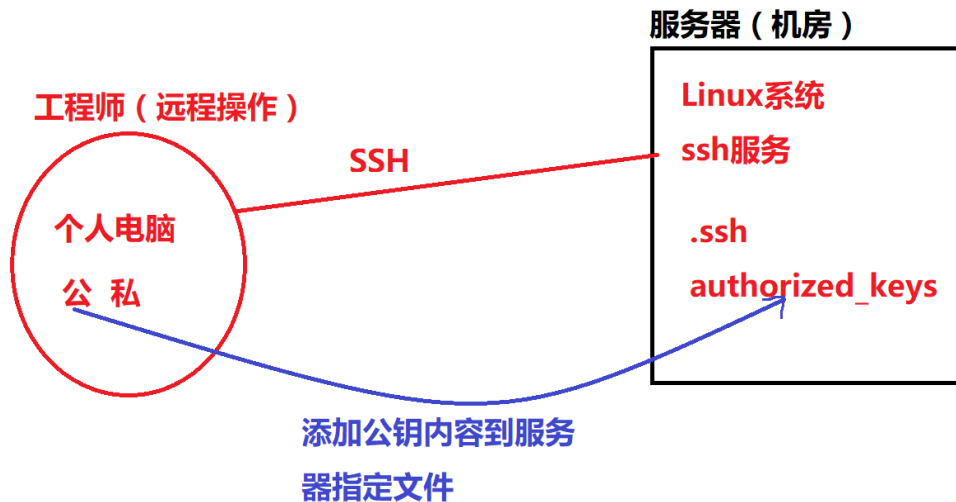
- 启动和关闭：

```
sudo service ssh start/restart/stop  
/etc/init.d/ssh start/restart/stop
```

ssh命令

ssh登录远程主机

```
ssh [-p port] username@ip  
退出： exit 或 ctrl-D
```



scp命令

scp命令可以用来通过安全、加密的连接在机器间传输文件。

把本地文件传输给远程系统：

```
scp localfile username@tohostname:/newfilename
```

把远程文件传输给本地系统：

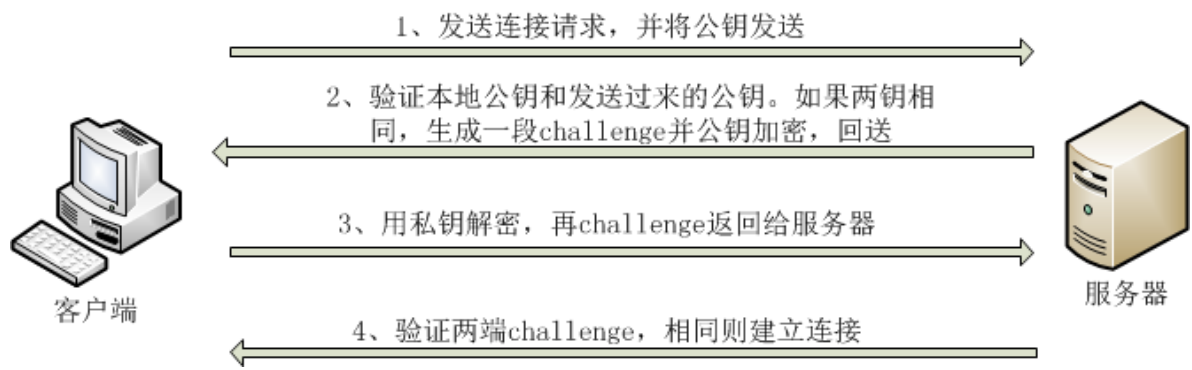
```
scp username@tohostname:/remotefile /localfile
```

ssh秘钥

每次登录远程主机都需要输入密码是很不便捷的，如果要加速这一步骤，可以利用密钥对进行连接，主要思路是：生成一对公钥私钥，私钥在local主机上，公钥在远程服务器上，每次建立ssh连接自动检查密钥对是否匹配。

生产ssh秘钥步骤

1. 生产密钥对： `ssh-keygen` 执行以后会在主目录下生成一个.ssh文件夹,其中包含私钥文件id_rsa和公钥文件id_rsa.pub。
2. 在服务器主机上将id_rsa.pub文件的内容附加~/ssh/authorized_keys文件中，并修改器权限。



shell脚本介绍

shell在Linux系统下也是脚本语言，和python一样属于解释型语言，在Linux运维和自动化处理方面有很多应用。下面简单介绍下shell脚本。

其实shell脚本的本质是一系列命令的有序集合，创建shell脚本的过程主要分三步：

- 创建 .sh 文件，写入若干行shell命令或者操作系统命令
- 使用 chmod 修改可执行权限
- 直接在命令行调用执行

当然shell脚本语法中也有变量，语句等内容，这里不一一介绍。

