

Linux系统基础

[Linux 介绍](#)

[文件操作](#)

[文件交互命令](#)

[列出目录内容](#)

[切换目录](#)

[创建文件和文件夹](#)

[移动文件和文件夹](#)

[删除文件和文件夹](#)

[识别文件类型](#)

[Shell基本操作](#)

[终端常见快捷键](#)

[Tab键](#)

[命令自动补全](#)

[文件和目录名自动补全](#)

[查看可用命令或文件](#)

[帮助信息查看](#)

[用户和权限管理](#)

[用户管理](#)

[查看用户信息](#)

[切换用户](#)

[文件权限](#)

Linux 介绍

Linux 由芬兰计算机科学家林纳斯·托瓦兹 (Linus Torvalds) 于 1991 年开发。他最初只是为了个人兴趣，开发一个新的自由操作系统内核（即 Linux 内核）。随后，托瓦兹将代码发布到互联网上，吸引了来自全球的开发者参与，从而快速发展成为一个成熟的操作系统。

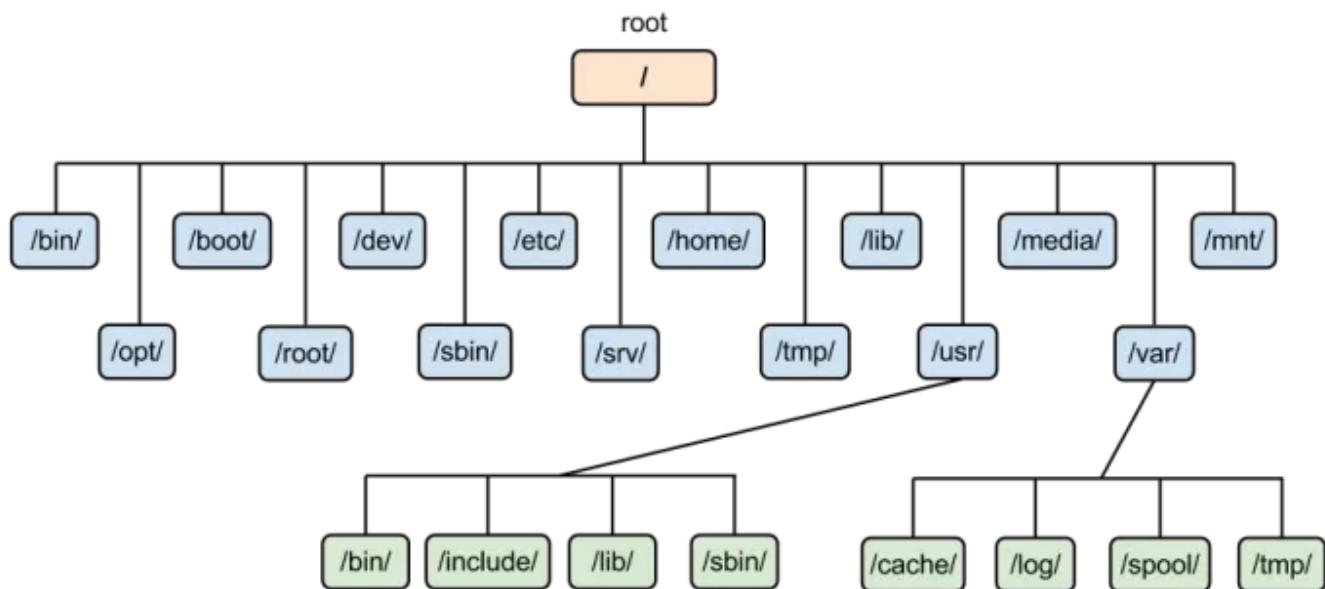
Linux 是完全开源的操作系统，任何人都可以查看、修改和分发其代码。这种开源模式推动了 Linux 的迅速发展和广泛应用。

有多个 Linux 发行版（如 Ubuntu、CentOS、Debian 等），每个发行版都基于相同的 Linux 内核，但提供不同的工具和软件包管理系统，以满足不同用户的需求。



文件操作

在Linux中，文件系统采用层次化结构。所有的文件和目录都由根目录（/）开始，形成一个树状结构。每个目录都可以包含其他目录或者文件。了解和掌握Linux的文件系统结构，有助于更好地管理系统文件以及数据。



各目录作用如下：

目录	描述	示例
/	根目录，是文件系统的起点，所有目录和文件都位于此。	/bin, /home等

/bin	存放系统启动时必要的基本命令和程序，供所有用户使用。	ls, cp, mv, rm等
/boot	包含引导加载器和启动文件，如内核文件。	vmlinuz, grub/等
/dev	存放设备文件，用于与硬件设备交互。	/dev/sda, /dev/tty0等
/etc	存放系统配置文件。	/etc/passwd, /etc/fstab等
/home	用户的主目录，每个用户有一个子目录，用于存储个人文件和设置。	/home/alice, /home/bob
/lib	存放系统和应用程序所需的共享库文件。	/lib/libc.so, /lib/modules/
/media	用于挂载可移动媒体，如USB驱动器和光盘。	/media/usb, /media/cdrom
/mnt	用于挂载临时文件系统或设备。	/mnt/external_drive
/opt	存放第三方应用程序。	/opt/google/, /opt/mysql/
/proc	存放虚拟文件，提供系统和进程的实时信息。	/proc/cpuinfo, /proc/meminfo
/root	root用户的主目录。	/root/.bashrc, /root/data/
/sbin	存放系统管理命令，主要由管理员（root）使用。	ifconfig, reboot等
/tmp	存放临时文件，供用户和程序使用。	/tmp/file.txt, /tmp/cache/
/usr	存放共享的只读文件，包含应用程序、库和文档等。	/usr/bin, /usr/lib, /usr/share/
/var	存放可变文件，如日志、邮件和缓存文件。	/var/log/, /var/www

文件交互命令

在 Linux 系统中，常用的命令可以帮助用户进行文件管理、系统操作、进程监控等日常任务。对于文件管理，`ls` 命令可以列出目录内容，`cd` 命令用来更改当前工作目录，`pwd` 用来显示当前路径。创建文件和目录可以使用 `touch` 和 `mkdir` 命令，删除文件和目录则使用 `rm` 和 `rmdir`。在操作文件时，还可以使用 `cp` 命令复制文件，`mv` 命令移动或重命名文件。

命令	功能描述	示例用法
<code>ls</code>	列出目录内容	<code>ls</code> 、 <code>ls -l</code> 、 <code>ls -a</code>
<code>cd</code>	切换目录	<code>cd /path/to/dir</code> 、 <code>cd ..</code> 、 <code>cd ~</code>
<code>pwd</code>	显示当前工作目录	<code>pwd</code>
<code>mkdir</code>	创建新目录	<code>mkdir new_dir</code>
<code>rmdir</code>	删除空目录	<code>rmdir empty_dir</code>
<code>rm</code>	删除文件或目录	<code>rm file.txt</code> 、 <code>rm -r dir</code>
<code>cp</code>	复制文件或目录	<code>cp file1.txt file2.txt</code> 、 <code>cp -r dir1 dir2</code>
<code>mv</code>	移动或重命名文件/目录	<code>mv file1.txt file2.txt</code> 、 <code>mv oldname newname</code>
<code>cat</code>	显示文件内容	<code>cat file.txt</code>
<code>find</code>	查找指定文件或目录	<code>find /path/to/search -name "file.txt"</code>
<code>locate</code>	快速查找文件（依赖数据库索引）	<code>locate file.txt</code>
<code>which</code>	查找可执行文件的位置	<code>which python</code>
<code>file</code>	查看文件类型	<code>file file.txt</code>

列出目录内容

`ls` 命令用于列出指定目录下的文件和目录。

`ls -l`：显示文件的详细信息（包括权限、大小、修改时间等），`ls -a` 显示所有文件，包括隐藏文件（以 `.` 开头的文件）

切换目录

`cd` 命令用于改变当前的工作目录。

常见用法：

- `cd ..` 返回上一级目录；
- `cd ~` 切换到当前用户的主目录；
- `cd -` 返回到上次所在的目录。

创建文件和文件夹

使用 `touch` 命令可以创建空白文件。

```
└──(kali㉿kali)-[~/Gok/PT]
    └─$ touch note

└──(kali㉿kali)-[~/Gok/PT]
    └─$ ls
        note
```

如果要创建文件夹，需要使用 `mkdir` 命，`-p` 选项递归创建目录。

```
└──(kali㉿kali)-[~/Gok/PT]
    └─$ mkdir folder1

└──(kali㉿kali)-[~/Gok/PT]
    └─$ ls
        folder1  note

└──(kali㉿kali)-[~/Gok/PT]
    └─$
```

移动文件和文件夹

`cp` 命令用于复制文件或目录，如果想递归复制目录及其内容则用 `cp -r dir1 dir2` 命令。

```
└── (kali㉿kali)-[~/Gok/PT]
    └─$ cp note note2
```

```
└── (kali㉿kali)-[~/Gok/PT]
    └─$ ls
        folder1  note  note2
```

`mv` 命令用于移动文件或重命名文件。

```
└── (kali㉿kali)-[~/Gok/PT]
    └─$ mv note2 note3
```

```
└── (kali㉿kali)-[~/Gok/PT]
    └─$ ls
        folder1  note  note3
```

删除文件和文件夹

`rm` 命令用于删除文件或目录，如果想删除文件夹则添加 `-r` 选项。

```
└── (kali㉿kali)-[~/Gok/PT]
    └─$ rm note3
```

```
└── (kali㉿kali)-[~/Gok/PT]
    └─$ ls
        folder1  note
```

```
└── (kali㉿kali)-[~/Gok/PT]
    └─$ rm -r folder1
```

```
└── (kali㉿kali)-[~/Gok/PT]
    └─$ ls
        note
```

识别文件类型

正常情况下我们可以通过肉眼识别文件拓展名来确定文件类型，当我们不了解文件类型及其用途的时候，就可以用 `file` 命令进行识别。

```
└── (kali㉿kali)-[~/Gok/PT]
    └─$ file test
        test: ASCII text
```

Shell基本操作

Shell 是一种命令行界面（CLI），用于与操作系统进行交互。它是用户与操作系统之间的“中介”，接收用户输入的命令并将其传递给操作系统执行，再将执行结果返回给用户。Shell 本质上是一个命令解释器，负责解释并执行用户输入的命令，以下是一些常见的shell。

Bourne Shell (**sh**)：最早的 Unix Shell 在 1977 年开发，现在仍然被一些系统使用。

Bash Shell (**bash**)：Bash 是 Bourne-Again Shell 的缩写，在1987年开发，是 Linux系统中最常用的 Shell 之一，也是默认的 Shell。它继承了 sh 和 ksh 的大部分特性，并增加了很多新的特性，例如命令补全、历史命令、别名、条件语句等。

Z Shell (**zsh**)：在1990年开发，是 Bash 的替代品之一。Zsh 具有比 Bash 更强大的命令补全、更灵活的配置和更强大的脚本语言等特性。

终端常见快捷键

基本操作

Ctrl + C：终止当前正在运行的命令或进程。

Ctrl + Shift + C：复制终端中的文本。

Ctrl + Shift + V：粘贴剪贴板中的内容到终端中。

光标移动

Ctrl + A：将光标移动到行首（类似 `Home` 键）。

Ctrl + E：将光标移动到行尾（类似 `End` 键）。

Ctrl + Left / Ctrl + Right：按单词移动光标（在某些终端中）。

删除操作

Ctrl + K : 删除从光标位置到行尾的内容。

Ctrl + U : 删除从光标位置到行首的内容。

Ctrl + W : 删除光标前的一个单词。

Alt + D : 删除光标后的一个单词。

Ctrl + D : 删除光标位置的字符。

Backspace : 删除光标前的一个字符。

Ctrl + H : 删除光标前的一个字符（与 **Backspace** 相同效果）。

命令历史

Ctrl + P 或 **↑** : 显示上一条命令历史。

Ctrl + N 或 **↓** : 显示下一条命令历史。

Ctrl + R : 向后搜索命令历史（递归搜索）。

Ctrl + S : 向前搜索命令历史（在部分终端中可能被暂停功能占用）。

进程控制

Ctrl + Z : 将当前进程挂起到后台。

jobs : 查看当前后台任务列表。

fg : 将后台进程恢复到前台运行。

bg : 将挂起的任务继续在后台运行。

终端控制

Ctrl + S : 暂停终端输出（屏幕冻结）。

Ctrl + Q : 恢复终端输出（解除冻结状态）。

Ctrl + D : 退出当前 shell（在空行输入 **exit** 的作用相同）。

Ctrl + Shift + N 或 **Ctrl + Shift + T** : 在终端中打开一个新的窗口或标签页（在某些终端模拟器中）。

Tab键

在终端中，**Tab** 键是一个非常有用的快捷键，主要用于自动补全和查看建议，作用非常强大。

命令自动补全

当你输入命令的前几个字母时，按下 `Tab` 键，终端会尝试自动补全该命令。

例如，输入 `pyth` 然后按 `Tab` 键，终端可能会补全为 `python`。

如果有多个匹配项，比如同时存在 `python2` 和 `python3`，按一次 `Tab` 键后可能不会自动补全，再按一次 `Tab` 键，则会列出所有匹配项。

文件和目录名自动补全

在输入文件名或目录名时，按下 `Tab` 键也可以自动补全路径。例如，输入 `cd Doc` 然后按 `Tab` 键，可能会补全为 `cd Documents`。当文件名或目录名有多个匹配时（如 `Documents` 和 `Downloads`），按 `Tab` 两次会显示所有匹配项。

查看可用命令或文件

当你输入一个命令的部分内容或路径时，按 `Tab` 两次可以显示所有匹配的命令或文件名。例如，在终端中输入 `git c` 后按两次 `Tab`，会列出所有以 `c` 开头的 `git` 子命令（如 `checkout`、`clone`、`commit` 等）。

帮助信息查看

使用 `man` 命令可以打开命令的手册页，详细描述该命令的功能、选项、用法等信息，使用 `空格` 和 `b` 可以进行翻页操作，按 `q` 退出。

大多数 Linux 命令都支持 `--help` 选项，用于显示命令的简短帮助信息，包括命令的基本用法和常见选项。这种方式比 `man` 命令更简洁，适合快速查看命令用法。

用户和权限管理

用户管理

Linux 用户管理是系统管理员在Linux系统上创建、管理和删除用户账户的过程。这对于确保系统安全性和管理权限至关重要。Linux 用户管理包括用户创建、修改、删除、密码管理以及组管理等操作。在Linux系统中，用户分为 普通用户 和 超级用户（root）。每个用户都有唯一的用户名、用户ID（UID）和组ID（GID）。

普通用户: 用于日常任务，如编辑文件、运行应用程序等。每个普通用户通常属于一个或多个组。

超级用户: 具有系统所有权限的用户，通常用来进行系统管理任务。

`useradd` 命令用于创建新用户。它可以指定用户名、用户的默认目录、用户所属的组等。`usermod` 命令可以修改用户的属性，`userdel` 可以用来删除用户。

添加john用户，为其创建用户目录并指定默认shell:

```
sudo useradd -m -s /bin/bash john
```

`passwd` 命令用于更改用户密码。

```
sudo passwd john
```

查看用户信息

用户的基本信息（如用户名、UID、GID、主目录、默认shell等）存储在 `/etc/passwd` 文件中。

用户的密码信息（加密后的密码）存储在 `/etc/shadow` 文件中。这个文件只有超级用户可读。

`id` 命令用于显示当前用户或指定用户的 UID、GID 和所属的组。

切换用户

`su` 命令允许你以另一个用户的身份运行命令。常用于切换到超级用户（root）。

```
su root
```

`sudo` 命令允许普通用户以超级用户（root）权限执行命令。使用 `sudo` 时需要输入当前用户的密码，而不是 root 密码。

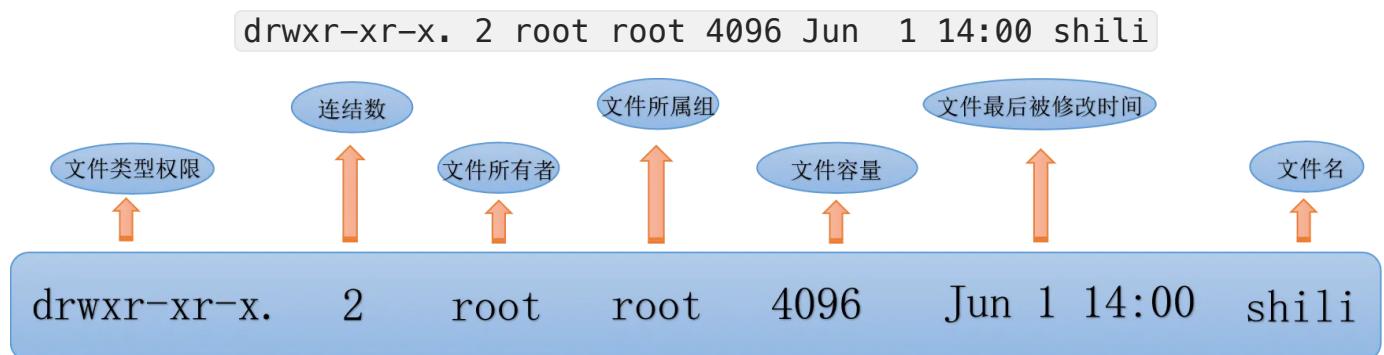
```
sudo apt update
```

文件权限

权限是操作系统用来限制对资源访问的一种机制，权限一般分为读、写、执行。在 Linux 系统中，不同的用户所处的地位也不尽相同，不同地位的用户拥有不同的权限等级，为了保证系统的安

全性，Linux 系统针对不同用户的权限制定了不同的规则。

以 root 用户文件信息为例：



Linux 中的权限使用三种不同的角色来表示：

用户 (User, U) : 文件的所有者。

组 (Group, G) : 文件所属的用户组。

其他 (Other, O) : 其他所有未包含在前两者中的用户。

权限通常以 `rwx` 的形式显示：

`r` (read) : 可读取文件内容。

`w` (write) : 可修改文件内容。

`x` (execute) : 可执行文件。

三个权限位的位置在文件或目录中是不变的。若是没有权限，则用“-”表示。

文件类型	属主权限	属组权限	其他用户权限
0	1 2 3	4 5 6	7 8 9
d	r w x	r - x	r - x
目录文件	读 写 执行	读 写 执行	读 写 执行

使用 `chmod` 命令可以修改文件或目录的权限。例如：

`chmod u+x filename` : 为文件所有者增加执行权限。

`chmod g-w filename` : 为用户组移除写权限。

`chmod o+r filename` : 为其他用户增加读取权限。

也可以使用数字的形式，读 (r) = 4；写 (w) = 2；执行 (x) = 1

例如我们要将文件权限设置为 `rw-r--r--`，则执行命令 `cmod 644 file.txt`。