

# 后台运行

---

- &: 命令行尾输入, 命令执行后台运行
- fg % 作业号: 作业前台执行
- bg % 作业号: 作业后台执行

# 用户权限

---

- 主user (u) -同一组的 (g) -其他 (o)
- rwx: 000-000-000
- chmod 777
- chmod a+rw file
- chmod u+x file: 将文件的执行权限添加给文件所有者。
- chmod g-rw file: 将文件的读取和写入权限从文件所属组中删除。
- chmod o-rwx file: 将文件的所有权限从其他用户中删除, 即禁止其他用户读取、写入和执行该文件。

# 硬链接软连接(符号链接)

---

在Linux中, 硬链接和软链接都可以用于实现文件共享, 但它们实现的方式有所不同。

硬链接是指多个文件名指向同一物理文件, 同一个i节点 (inode), 文件内容都是相同的。**硬链接只能在同一文件系统中创建**, 因为文件系统是基于i节点的, 而不是文件名的。

软链接是一种文件, 包含一个指向另一个文件的路径名。类似于 Windows 中的快捷方式。软链接可以**跨越文件系统边界**, 因为它们只是简单地指向另一个路径名, 而不是一个i节点。

因此, 硬链接和软链接的区别在于:

- **硬链接是指向同一个i节点, 而软链接是指向另一个路径名。**
- **硬链接只能在同一文件系统中使用, 而软链接可以跨越文件系统边界。**
- **在删除原文件时, 硬链接只有当所有链接都被删除后, 才会真正删除文件的内容; 而软链接会失效, 成为一个死链接, 不再指向任何文件。**

# 虚拟文件系统

---

虚拟文件系统是Linux内核的一个软件层, 为用户空间的程序提供文件系统接口, 并允许不同的文件系统共存。它在内存中创建, 不需要保存在永久存储介质中。虚拟文件系统本身没有文件, 只是用户与实际文件系统之间的接口。Linux支持众多不同类型的文件系统共存, 并且跨文件系统的操作也得益于虚拟文件系统的存在。因此, 所有的文件系统都依赖和协同工作于虚拟文件系统之上。

# 补充

---

1. 超级块：是文件系统的元数据信息，包含了文件系统的总体信息，如文件系统的类型、大小、块大小、i 节点数量、数据块数量等。超级块只有一个，存储在文件系统的特定位置，用于描述整个文件系统的特征和状态。
2. i 节点表：记录了文件和目录的元数据信息，包括文件的权限、拥有者、创建时间、修改时间、访问时间等，还包括了文件数据块在外存的物理地址信息。i 节点表是一个数组，每个元素代表一个文件或目录，每个 i 节点都有一个唯一的编号（inode number），用于在文件系统中标识一个文件或目录。
3. 数据块：是存储文件实际数据的区域，它们根据文件系统的块大小进行分配和管理。数据块也是一个数组，每个元素代表一个物理块，每个块大小通常为 4KB 或 8KB，用于存储文件的实际数据。
4. 目录项：记录了目录中所有文件和子目录的元数据信息，包括文件名、i 节点编号等。目录项也是一个数组，每个元素代表一个文件或目录，它们按照一定的顺序排列，用于在目录中查找特定的文件或目录。

## df 和 du 两者区别

---

### 1、df—disk free

可以快速获取 磁盘 被占用了多少空间，目前还剩下多少空间等信息。

### 2、du—disk usage

显示磁盘空间的使用情况，统计 目录（或文件）所占磁盘空间的大小。  
是不是感觉字面看起来没啥区别，实际有很多不同哦

df 和 du 的不同点：

#### （1）统计的范围不同

df是从总体上统计系统各磁盘的占用情况，不能统计具体的文件夹或文件的大小。

du既可以从总体上统计，又可以统计具体的某个文件夹或文件的大小。

## SystemV通信方式：共享内存，消息队列，信号量

---

## Top Half和Bottom Half

---

在Linux内核中，当设备接收到外部事件时，会触发中断。中断处理程序可以分为两个部分：Top Half和Bottom Half。

Top Half（顶半部分）是中断处理程序的第一部分，它必须在最短时间内完成。Top Half通常用于完成一些紧急的任务，如禁用中断、处理硬件设备的状态、启动数据传输等。Top Half处理完成后，中断处理程序就会退出。

Bottom Half（底半部分）是中断处理程序的第二部分，它可以延迟执行，不需要在最短时间内完成。Bottom Half通常用于完成一些非紧急的任务，如处理数据、唤醒等待队列、启动新的进程等。Bottom Half是在Top Half之后异步执行的。

需要Bottom Half机制的原因是，一些中断处理程序可能需要花费大量时间来完成一些耗时的操作，这可能会导致中断处理程序长时间占用CPU资源，影响系统的响应能力。使用Bottom Half机制可以将这些耗时的操作放到Top Half之后异步执行，让中断处理程序尽快退出，释放CPU资源，提高系统的响应能力。

在Linux内核中，实现Bottom Half机制的方法有多种，包括软中断（softirq）、任务队列（tasklet）、工作队列（workqueue）等。软中断是最基本的Bottom Half实现机制，它可以在中断处理程序退出后立即执行，但它只能在本地CPU上执行。任务队列（tasklet）是一种轻量级的Bottom Half实现机制，它可以在所有CPU上执行，但不能被睡眠进程调用。工作队列（workqueue）是一种更加灵活的Bottom Half实现机制，它可以在所有CPU上执行，并且可以被睡眠进程调用，但它的执行时间相对较长。

在Linux内核中，中断处理程序分为Top Half和Bottom Half。Top Half用于紧急任务，必须在最短时间内完成；Bottom Half用于非紧急任务，可以在Top Half之后异步执行。Bottom Half机制的作用是提高系统响应能力，因为一些耗时操作可以在Top Half之后异步执行。Bottom Half的实现机制包括软中断、任务队列和工作队列等。

## 用户线程和内核线程

在Linux中，用户线程是在用户空间中实现的，由用户程序自己管理，而内核线程是在内核空间中实现的，由内核管理和调度。用户线程的执行效率较高，但当一个用户线程被阻塞时，整个进程都会被阻塞；而内核线程可以直接访问内核空间的资源，可以提高系统的并发处理能力。每个进程都有一个或多个用户线程，但不是每个进程都有内核线程。

## 内核模块的作用

Linux内核模块是一种动态加载的内核代码，可以在运行时插入或卸载，用于扩展内核功能、优化内核性能、修改内核行为以及调试内核问题。使用内核模块可以使内核更加灵活，避免了重新编译整个内核的复杂性和风险。内核模块的开发需要考虑内核编程规范、安全性和稳定性。

## uname

`uname` 是一个命令行工具，用于显示当前系统的基本信息，例如操作系统类型、主机名、内核版本等等。它可以在Linux、Unix、macOS等操作系统上使用。

在Linux中，使用 `uname` 命令时，常用的参数及其含义如下：

- `-a`：显示所有信息，相当于同时使用 `-s`、`-n`、`-r`、`-v` 和 `-m` 参数；
- `-s`：显示操作系统类型；
- `-n`：显示主机名；
- `-r`：显示内核版本号；
- `-v`：显示内核发布日期和时间；
- `-m`：显示计算机硬件架构。

例如，要显示系统的内核版本号，可以在终端中输入以下命令：

```
uname -r
```

执行后，终端会输出当前系统的内核版本号。