在linux命令中reboot是重新启动，****shutdown -r now是立即停止然后重新启动，都说他们两个是一样的，其实是有一定的区别的****。

****shutdown****命令可以****安全地关闭****或重启Linux系统，它在系统关闭之前给系统上的所有登录用户提示一条警告信息。该命令还允许用户指定一个时间参数，可以是一个精确的时间，也可以是从现在开始的一个时间段。

精确时间的格式是hh:mm，表示小时和分钟，时间段由+ 和分钟数表示。系统执行该命令后会自动进行数据同步的工作。

该命令的一般格式: shutdown [选项] [时间] [警告信息]

命令中各选项的含义为:

- k 并不真正关机而只是发出警告信息给所有用户

- r 关机后立即重新启动

- h 关机后不重新启动

- f 快速关机重启动时跳过fsck

- n 快速关机不经过init 程序

- c 取消一个已经运行的shutdown

需要特别说明的是该命令只能由超级用户使用。

例1，系统在十分钟后关机并且马上重新启动: # shutdown –r +10

例2，系统马上关机并且不重新启动:# shutdown –h now

****halt****是最简单的关机命令，其****实际上是调用shutdown -h命令****。halt执行时，杀死应用进程，文件系统写操作完成后就会停止内核。

halt命令的部分参数如下：

[-f] 没有调用shutdown而强制关机或重启

[-i] 关机或重新启动之前，关掉所有的网络接口

[-p] 关机时调用poweroff，此选项为缺省选项

****reboot****的工作过程与halt类似，其作用是重新启动，而halt是关机。其参数也与halt类似。reboot命令重启动系统时是删除所有的进程，而不是平稳地终止它们。因此，****使用reboot命令可以快速地关闭系统，但如果还有其它用户在该系统上工作时，就会引起数据的丢失。所以使用reboot命令的场合主要是在单用户模式****。

****init****是所有进程的祖先，其进程号始终为1。init用于切换系统的运行级别，切换的工作是立即完成的。****init 0命令用于立即将系统运行级别切换为0，即关机；init 6命令用于将系统运行级别切换为6，即重新启动****

Samba是在Linux和UNIX系统上实现[SMB协议](https://baike.baidu.com/item/SMB%E5%8D%8F%E8%AE%AE" \t "/home/ts/Documents\\x/_blank)的一个免费软件，由服务器及客户端程序构成。[SMB](https://baike.baidu.com/item/SMB" \t "/home/ts/Documents\\x/_blank)（Server Messages Block，信息服务块）是一种在[局域网](https://baike.baidu.com/item/%E5%B1%80%E5%9F%9F%E7%BD%91" \t "/home/ts/Documents\\x/_blank)上共享文件和打印机的一种[通信协议](https://baike.baidu.com/item/%E9%80%9A%E4%BF%A1%E5%8D%8F%E8%AE%AE" \t "/home/ts/Documents\\x/_blank)。

Linux文件权限一共10位长度，分成四段，第三段表示的内容是\_（） \_\_ 。

* 1，代表文件类型，一般有三类，- 代表普通的二进制文件，l 代表符号链接文件（软链接），d 代表目录文件
* 2-4，代表文件所有者的权限表示情况，分别对应 可读(r) 可写(w) 可执行(w)，- 代表没有该项权限
* 5-7，代表文件所有这所属组的权限表示情况。分别对应 可读(r) 可写(w) 可执行(w)，- 代表没有该项权限
* 8-10,代表除文件所有者和所属组的其他人所拥有的权限表示情况。分别对应 可读(r) 可写(w) 可执行(w)，- 代表没有该项权限

如；umask 的值为022

1. 创建一个文件的默认权限应该就是777-022=755
2. 由于 执行权限(x) 比较危险，Linux默认不会赋予此权限，所以实际创建文件的默认权限是 755-111=644
3. 如果此时创建的是一个软链接文件，则权限位为 lrx-r--r-；如果此时创建的是一个普通文件，则权限位为 -rx-r--r--；如果此时创建的是一个目录文件，则权限位为 drx-r--r--

linux下文件系统包括以下几种类型

    1. 普通文件。即一般意义上的文件

    2. 目录文件。它用来包含文件，文件一定在某个目录下。

    3. 管道文件。

        管道式进程间通信的一种机制，通常，一个进程写一些数据到管道中，这些数据就可以被另一个进程从这个管道中读取出来。

        管道分为两种类型：无名管道与命名管道。

        无名管道：由进程在使用时创建，读写结束关闭文件后消失，它们并不存在于文件系统中，没有文件名称

        命名管道：在形式上就是文件系统中的一个文件，虽然不占用内存空间，但有自己的文件名。命名管道通常称为FIFO

    4. 设备文件。

    5. 符号链接。

        符号链接是一种特殊的文件，它的内容是指向另一个文件的路径。对符号链接进行操作时，系统会根据情况对这个操作转移到它

        所指向的文件上去，而不是对它本身进行操作。

    6. socket。socket也是进程间通信的方式，与管道不同的是，它可以在不同主机上的进程间通信。socket在linux上是以文件的方式

        进行操作的