# Linux背景

## Linux发展史

Linux是一种自由和开放源代码的类UNIX操作系统，该操作系统的内核由林纳斯托瓦兹在1991年首次发布，之后，在加上用户空间的应用程序之后，就成为了Linux操作系统。严格来讲，Linux只是操作系统内核本身，但通常采用“Linux内核”来表达该意思。而Linux则常用来指基于Linux内核的完整操作系统，它包括GUI组件和许多其他实用工具。

GNU通用公共许可协议（GNU General Public License，简称GNU GPL或GPL），是一个广泛被使用的自由软件许可协议条款，最初由理查德斯托曼为GNU计划而撰写，GPL给予了计算机程序自由软件的定义，任何基于GPL软件开发衍生的产品在发布时必须采用GPL许可证方式，且必须公开源代码。

Linux是自由软件和开放源代码软件发展中最著名的例子。只要遵循GNU通用公共许可证，任何个人和机构都可以自由地使用Linux的所有底层源代码，也可以自由地修改和再发布。随着Linux操作系统飞速发展，各种集成在Linux上的开源软件和实用工具也得到了应用和普及，因此，Linux也成为了开源软件的代名词。

## Linux应用现状

* + - 1. Linux在服务器领域的发展

随着开源软件在世界范围内影响力日益增强，Linux服务器操作系统在整个服务器操作系统市场格局中 占据了越来越多的市场份额，已经形成了大规模市场应用的局面。并且保持着快速的增长率。尤其在政府、金融、农业、交通、电信等国家关键领域。此外，考虑到Linux的快速成长性以及国家相关政策的扶持力度，Linux服务器产品一定能够冲击更大的服务器市场。

据权威部门统计，目前Linux在服务器领域已经占据75%的市场份额，同时，Linux在服务器市场的迅速崛起，已经引起全球IT产业的高度关注，并以强劲的势头成为服务器操作系统领域中的中坚力量。

* + - 1. Linux在桌面领域的发展

近年来，特别在国内市场，Linux桌面操作系统的发展趋势非常迅猛。国内如中标麒麟Linux、红旗Linux、深度Linux等系统软件厂商都推出的Linux桌面操作系统，目前已经在政府、企业、OEM等领域得到了广泛应用。另外SUSE、Ubuntu也相继推出了基于Linux的桌面系统，特别是Ubuntu Linux，已经积累了大量社区用户。但是，从系统的整体功能、性能来看，Linux桌面系统与Windows系列相比还有一定的差距，主要表现在系统易用性、系统管理、软硬件兼容性、软件的丰富程度等方面。

* + - 1. Linux在移动嵌入式领域的发展

Linux的低成本、强大的定制功能以及良好的移植性能，使得Linux在嵌入式系统方面也得到广泛应用，目前Linux以广泛应用于手机、平板电脑、路由器、电视和电子游戏机等领域。在移动设备上广泛使用的Android操作系统就是创建在Linux内核之上的。目前，Android已经成为全球最流行的智能手机操作系统，据2015年权威部门最新统计，Android操作系统的全球市场份额已达84.6%。

此外，思科在网络防火墙和路由器也使用了定制的Linux，阿里云也开发了一套基于Linux的操作系统“YunOS”，可用于智能手机、平板电脑和网络电视；常见的数字视频录像机、舞台灯光控制系统等都在逐渐采用定制版本的Linux来实现，而这一切均归功与Linux与开源的力量。

* + - 1. Linux在云计算/大数据领域的发展

互联网产业的迅猛发展，促使云计算、大数据产业的形成并快速发展，云计算、大数据作为一个基于开源软件的平台，Linux占据了核心优势；据Linux基金会的研究，86%的企业已经使用Linux操作系统进行云计算、大数据平台的构建，目前，Linux已开始取代Unix成为最受青睐的云计算、大数据平台操作系统。

## 发行版本

* + - 1. Debian

Debian运行起来极其稳定，这使得它非常适合用于服务器。debian这款操作系统无疑并不适合新手用户，而是适合系统管理员和高级用户。

* + - 1. Ubuntu

Ubuntu是Debian的一款衍生版，也是当今最受欢迎的免费操作系统。Ubuntu侧重于它在这个市场的应用，在服务器、云计算、甚至一些运行Ubuntu Linux的移动设备上很常见。Ubuntu是新手用户肯定 爱不释手的一款操作系统。

* + - 1. 红帽企业级Linux

这是第一款面向商业市场的Linux发行版。它有服务器版本，支持众多处理器架构，包括x86和 x86\_64。红帽公司通过课程红帽认证系统管理员/红帽认证工程师（RHCSA/RHCE），对系统管理员进行培训和认证。

* + - 1. CentOS

CentOS是一款企业级Linux发行版，它使用红帽企业级Linux中的免费源代码重新构建而成。这款重构版完全去掉了注册商标以及Binary程序包方面一个非常细微的变化。有些人不想支付一大笔钱，又能领略红帽企业级Linux；对他们来说，CentOS值得一试。此外，CentOS的外观和行为似乎与母发行版红帽企业级Linux如出一辙。CentOS使用YUM来管理软件包。

* + - 1. Fedora

小巧的Fedora适合那些人：想尝试最先进的技术，等不及程序的稳定版出来。其实，Fedora就是红帽公司的一个测试平台；产品在成为企业级发行版之前，在该平台上进行开发和测试。Fedora是一款非常好的发行版，有庞大的用户论坛，软件库中还有为数不少的软件包。

* + - 1. Kali Linux

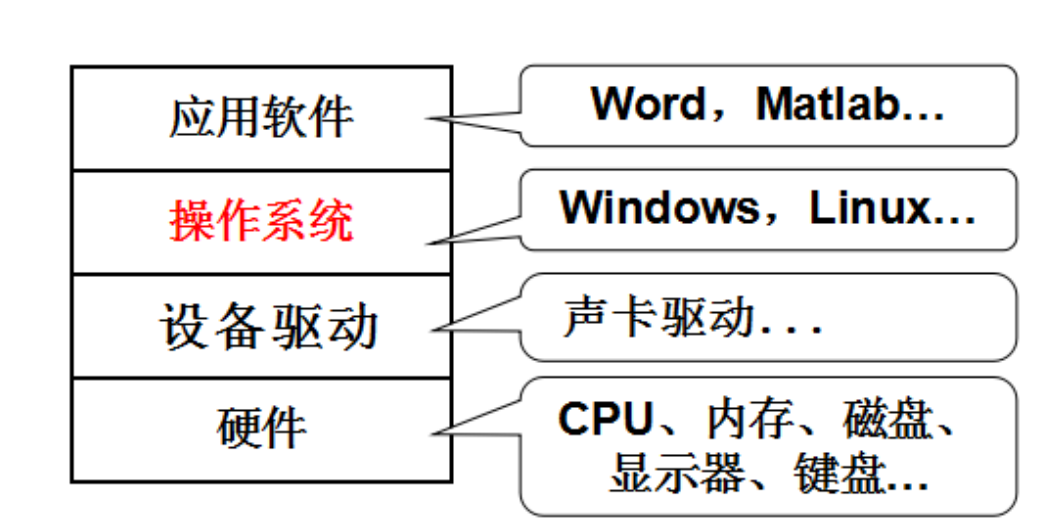
Kali Linux是Debian的一款衍生版。Kali旨在用于渗透测试。Kali的前身是Backtrack。用于Debian的所有Binary软件包都可以安装到Kali Linux上，而Kali的魅力或威力就来自于此。此外，支持Debian的用户论坛为Kali加分不少。Kali随带许多的渗透测试工具，无论是Wifi、数据库还是其他任何工具，都设计成立马可以使用。Kali使用APT来管理软件包。

毫无疑问，Kali Linux是一款渗透测试工具，或者是文明黑客青睐的操作系统。



## 定位

操作系统是一款管理软件，用来管理软硬件。对上（应用软件/用户）提供良好的运行环境，对下（设备驱动/硬件）管理好软硬件资源。使计算机更好用!这是操作系统的根本要义!



服务器为什么没有图形化界面？

进入服务器没有界面的原因是因为服务器通常是以命令行界面（CLI）的方式工作的，而不是图形用户界面（GUI）。

首先，服务器是为了处理大量的数据和资源，因此需要一个高效、稳定的操作环境。相比于图形界面，命令行界面更加轻量级、占用资源更少，可以提供更好的性能和可靠性。

其次，命令行界面提供了更强大的控制和管理功能。通过命令行可以执行各种操作，如安装软件、配置网络、管理用户等。命令行界面还可以自动化批量操作，提高工作效率。

另外，服务器通常是通过远程连接方式进行管理和访问的，而不是直接连接到服务器的物理界面。这意味着管理员可以通过终端或远程桌面等方式远程访问服务器，并通过命令行界面进行管理和操作。

当然，如果你希望在服务器上使用图形界面，也是可以的。可以通过安装图形界面的桌面环境，如Gnome、KDE等，来实现在服务器上使用带有界面的操作系统。但需要注意的是，这样会增加服务器的资源消耗，降低性能和可靠性。在大多数情况下，服务器管理员更倾向于使用命令行界面来管理服务器，以获得更好的性能和安全性。

总之，进入服务器没有界面是正常的，这是为了提供高效、稳定和强大的操作环境。管理员可以通过命令行界面来进行各种操作和管理任务，远程访问服务器并保持对服务器的全面控制。

# Linux常见指令及权限理解

## Linux下基本指令

注：linux下写指令操作等同于windows的基本操作。linux是命令行（CLI：Command-Line Interface）操作方式，windows是图形化用户界面（GUI：Graphical User Interface）操作方式，而这两种方式在操作系统层面来说没有区别。

* + - 1. ls指令（全称：list）

语法：ls [选项][目录或文件]

功能：对于目录，该命令列出该目录下的所有子目录与文件。对于文件，将列出文件名以及其他信息。

常用选项：

* -a 列出目录下的所有文件，包括以 . 开头的隐含文件。
* -d 将目录像文件一样显示，而不是显示其下的文件。如：ls –d 指定目录
* -i 输出文件的 i 节点的索引信息。 如 ls –ai 指定文件
* -k 以 k 字节的形式表示文件的大小。ls –alk 指定文件
* -l 列出文件的详细信息。
* -n 用数字的 UID,GID 代替名称。 （介绍 UID， GID）
* -F 在每个文件名后附上一个字符以说明该文件的类型，“\*”表示可执行的普通文件；“/”表示目录；“@”表 示符号链接；“|”表示FIFOs；“=”表示套接字(sockets)。（目录类型识别）
* -r 对目录反向排序。
* -t 以时间排序。
* -s 在I文件名后输出该文件的大小。（大小排序，如何找到目录下最大的文件）
* -R 列出所有子目录下的文件。(递归)
* -1 一行只输出一个文件

注：关于文件

* 文件=文件内容（数据）+文件属性（元数据）
* 操作文件：操作文件内容和操作文件属性
  + - 1. pwd命令（全称：print working directory）

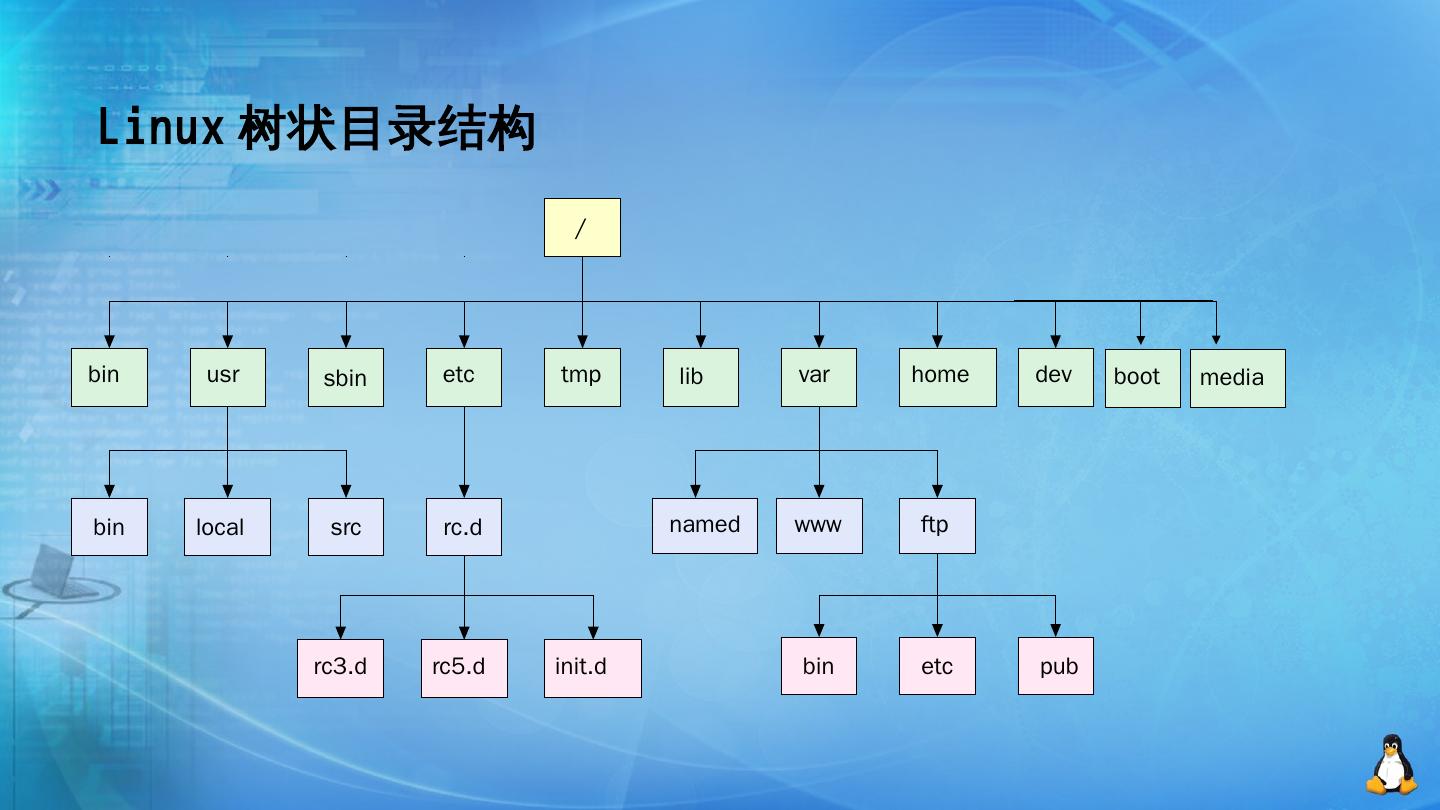
语法: pwd

功能：显示用户当前所在的目录

常用选项：无

* + - 1. cd指令（全称：change directory）

Linux系统中，磁盘上的文件和目录被组成一棵目录树，每个节点都是目录或文件。



注：几乎任何操作系统文件的目录组织都是一颗多叉树

* 叶子节点：普通文件或空目录
* 路上节点：只能是目录
* 路径+文件名 = 具有唯一性

语法:cd 目录名

功能：改变工作目录。将当前工作目录改变到指定的目录下。

举例：

|  |
| --- |
| cd .. : 返回上级目录  cd /home/litao/linux/ : 绝对路径  cd ../day02/ : 相对路径  cd ~：进入用户家目  cd -：返回最近访问目录 |

* + - 1. touch指令

语法:touch [选项] ...文件...

功能：touch命令参数可更改文档或目录的日期时间，包括存取时间和更改时间，或者新建一个不存在的文件。

常用选项：

* -a 或--time=atime或--time=access或--time=use只更改存取时间。
* -c 或--no-create 不建立任何文档。
* -d 使用指定的日期时间，而非现在的时间。
* -f 此参数将忽略不予处理，仅负责解决BSD版本touch指令的兼容性问题。
* -m 或--time=mtime或--time=modify 只更改变动时间。
* -r 把指定文档或目录的日期时间，统统设成和参考文档或目录的日期时间相同。
* -t 使用指定的日期时间，而非现在的时间。
  + - 1. mkdir指令（全称：make directory）

语法：mkdir [选项] dirname...

功能：在当前目录下创建一个名为 “dirname”的目录

常用选项：

* -p, --parents 可以是一个路径名称。此时若路径中的某些目录尚不存在,加上此选项后,系统将自动建立好那些尚不存在的目录,即一次可以建立多个目录;

举例：

|  |
| --- |
| mkdir –p test/test1 : 递归建立多个目录 |

* + - 1. rmdir指令 && rm 指令（重要）（全称：remove）

rmdir是一个与mkdir相对应的命令。mkdir是建立目录，而rmdir是删除命令。

语法：rmdir [-p][dirName]

适用对象：具有当前目录操作权限的所有使用者

功能：删除空目录

常用选项：

|  |
| --- |
| -p 当子目录被删除后如果父目录也变成空目录的话，就连带父目录一起删除。 |

rm命令可以同时删除文件或目录

语法：rm [-f-i-r-v][dirName/dir]

适用对象：所有使用者

功能：删除文件或目录

常用选项：

* -f 即使文件属性为只读(即写保护)，亦直接删除
* -i 删除前逐一询问确认
* -r 删除目录及其下所有文件
  + - 1. man指令（重要）：

Linux的命令有很多参数，我们不可能全记住，我们可以通过查看联机手册获取帮助。访问Linux手册页的命令是 man

语法: man [选项] 命令

常用选项：

* -k 根据关键字搜索联机帮助
* num 只在第num章节找
* -a 将所有章节的都显示出来，比如 man printf 它缺省从第一章开始搜索，知道就停止，用a选项，当按下q退出，他会继续往后面搜索，直到所有章节都搜索完毕。

注：解释一下,面手册分为8章。1是普通的命令；2是系统调用，如open、write之类的(通过这个，至少可以很方便的查到调用这个函数，需要加什么头文件)；3是库函数，如printf、fread；4是特殊文件，也就是/dev下的各种设备文件；5是指文件的格式，比如passwd，就会说明这个文件中各个字段的含义；6是给游戏留的，由各个游戏自己定义；7是附件还有一些变量，比如向environ这种全局变量在这里就有说明；8是系统管理用的命令，这些命令只能由root使用，如ifconfig。

* + - 1. cp指令（重要）（全称：copy）

语法：cp [选项] 源文件或目录 目标文件或目录

功能: 复制文件或目录

说明: cp指令用于复制文件或目录，如同时指定两个以上的文件或目录，且最后的目的地是一个已经存在的目录，则它会把前面指定的所有文件或目录复制到此目录中。若同时指定多个文件或目录，而最后的目的地并非一个已存在的目录，则会出现错误信息

常用选项：

* -f 或 –force 强行复制文件或目录，不论目的文件或目录是否已经存在
* -i 或 --interactive 覆盖文件之前先询问用户
* -r递归处理，将指定目录下的文件与子目录一并处理。若源文件或目录的形态，不属于目录或符号链接，则一律视为普通文件处理
* -R 或 --recursive递归处理，将指定目录下的文件及子目录一并处理
  + - 1. mv指令（重要）（全称：move）

mv命令是move的缩写，可以用来移动文件或者将文件改名，是Linux系统下常用的命令，经常用来备份文件或者目录。

语法: mv [选项] 源文件或目录 目标文件或目录

功能:

1. 视mv命令中第二个参数类型的不同（是目标文件还是目标目录），mv命令将文件重命名或将其移至一个新的目录中。
2. 当第二个参数类型是文件时，mv命令完成文件重命名，此时，源文件只能有一个（也可以是源目录名），它将所给的源文件或目录重命名为给定的目标文件名。
3. 当第二个参数是已存在的目录名称时，源文件或目录参数可以有多个，mv命令将各参数指定的源文件均移至目标目录中。

常用选项：

* -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(destination)已经存在时，就会询问是否覆盖！
  + - 1. cat指令（全称：concatenate）

语法：cat [选项][文件]

功能：查看目标文件内容（连接文件并打印到标准输出设备上）

常用选项：

* -b 对非空输出行编号
* -n 对输出的所有行编号
* -s 不输出多行空行
  + - 1. more指令

语法：more [选项][文件]

功能：more命令，功能类似 cat

常用选项：

* -n 对输出的所有行编号
* q 退出more
  + - 1. less指令（重要）

less工具也是对文件或其它输出进行分页显示的工具，应该说是linux正统查看文件内容的工具，功能极其强大。less的用法比起more更加的有弹性。在more的时候，我们并没有办法向前面翻，只能往后面看但若使用了less时，就可以使用[pageup][pagedown]等按键的功能来往前往后翻看文件，更容易用来查看一个文件的内容！除此之外，在less里头可以拥有更多的搜索功能，不止可以向下搜，也可以向上搜。

语法： less [参数] 文件

功能： less与more类似，但使用less可以随意浏览文件，而more仅能向前移动，却不能向后移动，而且less在查看之前不会加载整个文件。

选项：

* -i 忽略搜索时的大小写
* -N 显示每行的行号
* /字符串：向下搜索“字符串”的功能
* ?字符串：向上搜索“字符串”的功能
* n：重复前一个搜索（与/或?有关）
* N：反向重复前一个搜索（与/或?有关）
* q:quit
  + - 1. head指令

head与tail就像它的名字一样的浅显易懂，它是用来显示开头或结尾某个数量的文字区块，head用来显示档案的开头至标准输出中，而tail想当然尔就是看档案的结尾。

语法： head[参数]...[文件]...

功能： head用来显示档案的开头至标准输出中，默认head命令打印其相应文件的开头10行。

选项： -n显示的行数

* + - 1. tail指令

tail命令从指定点开始将文件写到标准输出。使用tail命令的-f选项可以方便的查阅正在改变的日志文件，tail -f filename会把filename里最尾部的内容显示在屏幕上,并且不但刷新,使你看到最新的文件内容。

语法：tail[必要参数][选择参数][文件]

功能：用于显示指定文件末尾内容，不指定文件时，作为输入信息进行处理。常用查看日志文件。

选项：

* -f 循环读取
* -n 显示行数
  + - 1. 时间相关的指令

date显示

date 指定格式显示时间： date +%Y:%m:%d

date 用法：date [OPTION]... [+FORMAT]

1. 在显示方面，使用者可以设定欲显示的格式，格式设定为一个加号后接数个标记，其中常用的标记列表如下

* %H : 小时(00..23)
* %M : 分钟(00..59)
* %S : 秒(00..60)
* %X : 相当于 %H:%M:%S
* %d : 日 (01..31)
* %m : 月份 (01..12)
* %Y : 完整年份 (0000..9999)
* %F : 相当于 %Y-%m-%d

1. 在设定时间方面

* date -s //设置当前时间，只有root权限才能设置，其他只能查看。
* date -s 20080523 //设置成20080523，这样会把具体时间设置成空00:00:00
* date -s 01:01:01 //设置具体时间，不会对日期做更改
* date -s “01:01:01 2008-05-23″ //这样可以设置全部时间
* date -s “01:01:01 20080523″ //这样可以设置全部时间
* date -s “2008-05-23 01:01:01″ //这样可以设置全部时间
* date -s “20080523 01:01:01″ //这样可以设置全部时间

1. 时间戳

* 时间->时间戳：date +%s
* 时间戳->时间：date -d@1508749502

注：Unix时间戳（英文为Unix epoch, Unix time, POSIX time 或 Unix timestamp）是从1970年1月1日（UTC/GMT的午夜）开始所经过的秒数，不考虑闰秒。

* + - 1. Cal指令（全称：Calendar）

cal命令可以用来显示公历（阳历）日历。公历是现在国际通用的历法，又称格列历，通称阳历。“阳历”又名“太阳 历”，系以地球绕行太阳一周为一年，为西方各国所通用，故又名“西历”。

命令格式：cal [参数][月份][年份]

功能：用于查看日历等时间信息，如只有一个参数，则表示年份(1-9999)，如有两个参数，则表示月份和年份

常用选项：

* -3显示系统前一个月，当前月，下一个月的月历
* -j 显示在当年中的第几天（一年日期按天算，从1月1号算起，默认显示当前月在一年中的天数）
* -y 显示当前年份的日历
  + - 1. find指令

Linux下find命令在目录结构中搜索文件，并执行指定的操作。Linux下find命令提供了相当多的查找条件，功能很强大。由于find具有强大的功能，所以它的选项也很多，其中大部分选项都值得我们花时间来了解一下。即使系统中含有网络文件系统( NFS)，find命令在该文件系统中同样有效，只你具有相应的权限。在运行一个非常消耗资源的find命令时，很多人都倾向于把它放在后台执行，因为遍历一个大的文件系统可能会花费很长的时间(这里是指30G字节以上的文件系统)。

语法：find pathname -options

功能：用于在文件树种查找文件，并作出相应的处理（可能访问磁盘）

常用选项：-name 按照文件名查找文件。

* + - 1. grep指令（全称：global regular expression）

语法： grep [选项] 搜寻字符串文件

功能： 在文件中搜索字符串，将找到的行打印出来

常用选项：

* -i ：忽略大小写的不同，所以大小写视为相同
* -n ：顺便输出行号
* -v ：反向选择，亦即显示出没有'搜寻字符串'内容的那一行
  + - 1. zip/unzip指令

语法：zip 压缩文件.zip目录或文件

功能：将目录或文件压缩成zip格式

常用选项：-r 递归处理，将指定目录下的所有文件和子目录一并处理

* + - 1. tar指令（重要）（全称：tape archive）

语法：tar [-cxtzjvf] 文件与目录....

功能：打包/解包，不打开它，直接看内容

常用选项：

* -c：建立一个压缩文件的参数指令(create 的意思)；
* -x：解开一个压缩文件的参数指令！
* -t：查看tarfile里面的文件！
* -z：是否同时具有gzip的属性？亦即是否需要用gzip压缩？
* -j：是否同时具有bzip2的属性？亦即是否需要用bzip2压缩？
* -v：压缩的过程中显示文件！这个常用，但不建议用在背景执行过程！
* -f：使用档名，请留意，在f之后要立即接档名喔！不要再加参数！
* -C：解压到指定目录
  + - 1. bc指令（全称：basic calculator）

语法：bc [options] 文件

功能：进行浮点运算

常用选项：

* -l mathlib, 使用标准数学库, 例如使用内置函数就需要使用这个参数
* -q 退出，不显示欢迎信息
  + - 1. uname -r指令（全称：unix name）

语法：uname [选项]

功能：uname用来获取电脑和操作系统的相关信息。

补充说明：uname可显示linux主机所用的操作系统的版本、硬件的名称等基本信息。

常用选项：-a或–all详细输出所有信息，依次为内核名称，主机名，内核版本号，内核版本，硬件名，处理器类型，硬件平台类型，操作系统名称

* + - 1. 重要的热键
* [Tab]按键---具有『命令补全』和『档案补齐』的功能
* [Ctrl]-c按键---让当前的程序『停掉』
* [Ctrl]-d按键---通常代表着：『键盘输入结束(End Of File, EOF 或 End OfInput)』的意思；另外，他也可以用来取代exit
  + - 1. shutdown

语法：shutdown [选项]

常见选项：

* -h：将系统的服务停掉后，立即关机
* -r：在将系统的服务停掉之后就重新启动
* -t sec：-t 后面加秒数，亦即『过几秒后关机』的意思
  + - 1. 其他扩展命令
* 安装和登录命令：login、shutdown、halt、reboot、install、mount、umount、chsh、exit、last；
* 文件处理命令：file、mkdir、grep、dd、find、mv、ls、diff、cat、ln；
* 系统管理相关命令：df、top、free、quota、at、lp、adduser、groupadd、kill、crontab；
* 网络操作命令：ifconfig、ip、ping、netstat、telnet、ftp、route、rlogin、rcp、finger、mail、 nslookup；
* 系统安全相关命令：passwd、su、umask、chgrp、chmod、chown、chattr、sudo ps、who
* 其它命令：tar、unzip、gunzip、unarj、mtools、man、unendcode、uudecode。

## Shell命令及其运行原理

Linux严格意义上说的是一个操作系统，我们称之为“核心（kernel）“ ，但我们一般用户，不能直接使用kernel。而是通过kernel的“外壳”程序，也就是所谓的shell，来与kernel沟通。如何理解？为什么不能直接使用kernel？

从技术角度，Shell的最简单定义：命令行解释器（command Interpreter）主要包含：

* 将使用者的命令翻译给核心（kernel）处理。
* 同时，将核心的处理结果翻译给使用者。

对比windows GUI，我们操作windows 不是直接操作windows内核，而是通过图形接口，点击，从而完成我们的 操作（比如进入D盘的操作，我们通常是双击D盘盘符或者运行起来一个应用程序）。

shell 对于Linux，有相同的作用，主要是对我们的指令进行解析，解析指令给Linux内核。反馈结果在通过内核运行出结果，通过shell解析给用户。

## Linux权限管理

Linux下有两种用户：超级用户（root）、普通用户。

* 超级用户：可以再linux系统下做任何事情，不受限制
* 普通用户：在linux下做有限的事情
* 超级用户的命令提示符是“#”，普通用户的命令提示符是“$”。

命令：su [用户名]

功能：切换用户。 例如，要从root用户切换到普通用户user，则使用 su user。 要从普通用户user切换到root用户则使用 su root（root可以省略），此时系统会提示输入root用户的口令。

### 文件访问者的分类（人）

文件和文件目录的所有者：u---User

文件和文件目录的所有者所在的组的用户：g---Group

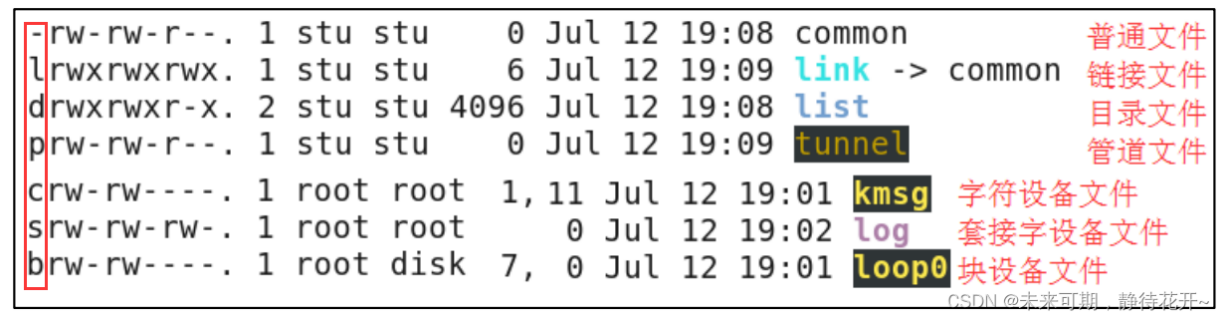
其它用户：o---Others

### 文件类型和访问权限（事务的属性）

Linux下所有的东西都可以看做文件，并且Linux下不以文件的扩展名区别文件类型，而是在文件属性中有一列专门记录文件类型。

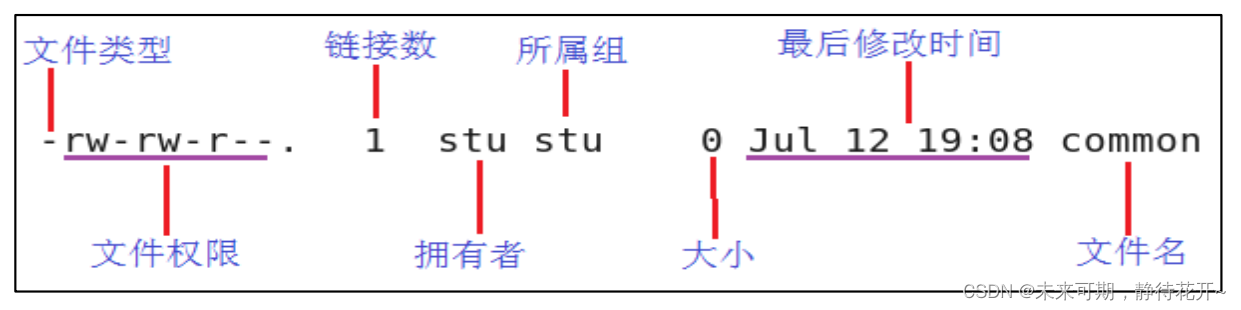
* d：目录/文件夹
* -：普通文件（源代码、库文件、可执行程序、文档压缩包等）
* l：链接文件（类似Windows的快捷方式）
* b：块设备文件（例如硬盘、光驱等）
* p：管道文件
* c：字符设备文件（键盘、显示器）
* s：套接口文件

这7种文件在Linux上呈现如下图所示：



注：Linux区分文件与后缀无关，后缀的本质是文件名的一部分

上图中列出的文件信息发别为：



文件的基本权限如下：

* 读：Read对文件而言，具有读取文件内容的权限；对目录来说，具有浏览该目录信息的权限
* 写：Write对文件而言，具有修改文件内容的权限；对目录来说具有删除移动目录内文件的权限
* 执行：execute对文件而言，具有执行文件的权限；对目录来说，具有进入目录的权限
* “—”：表示不具有该项权限

注：文件的读/写/执行权限只与文件的内容有关，与文件自己无关。删除一个文件不需要文件自身的读/写/执行权限，而需要它所在目录的读/写/执行权限。

### 文件权限值的表示方法

* + - 1. 字符表示方法

|  |  |  |  |
| --- | --- | --- | --- |
| Linux表示 | 说明 | Linux表示 | 说明 |
| r-- | 只读 | -w- | 只写 |
| --x | 仅可执行 | rw- | 可读可写 |
| -wx | 可写和可执行 | r-x | 可读可执行 |
| rwx | 可读可写可执行 | --- | 无权限 |

* + - 1. 8进制数值表示方法

|  |  |  |
| --- | --- | --- |
| 权限符号（读写执行） | 八进制 | 二进制 |
| r-- | 4 | 100 |
| -w- | 2 | 010 |
| --x | 1 | 001 |
| rw- | 6 | 110 |
| r-x | 5 | 101 |
| -wx | 3 | 011 |
| rwx | 7 | 111 |
| --- | 0 | 000 |

### 文件访问权限的相关设置方法

系统判断文件与用户之间的关系是，从user，group，other依次判别，只要能匹配上，就只能用所匹配的权限。比如一个文件的user有r权限，同时user=group，但是group有rw权限，那么这个user来操作文件时，优先匹配user的权限，就只有user的r权限，而不能用group的w权限。

* + - 1. chmod

功能：设置文件的访问权限

格式：chmod [参数] 权限 文件名

常用选项：

* R -> 递归修改目录文件的权限
* 说明：只有文件拥有者和root才能改变文件权限

chmod命令权限值的方式有两种

①用户表示符+/-=权限字符

* +:向权限范围增加权限代号所表示的权限
* -:向权限范围取消权限代号所表示的权限
* =:向权限范围赋予权限代号所表示的权限
* 用户符号：
* u：拥有者
* g：拥有者所有组
* o：其它用户
* a：所有用户

示例如下：

|  |
| --- |
| #chmod u+x /home/abc.txt //给abc.txt文件拥有者加上执行权限  #chmod o-x /home/abc.txt //给abc.txt文件其他用户取消执行权限  #chmod a=x /home/abc.txt //给abc.txt文件所有用户只设置执行权限  #chmod a=x+r=w file //给file文件所有用户先设置执行权限，再加上读权限，  //最后再只设置为w权限 |

②三位数8进制数字

举例如下：

|  |
| --- |
| #chmod 664 /home/abc.txt //给abc.txt的ugo用户分别设置权限为664 |

* + - 1. chowm

功能：修改文件的拥有者

格式：chown [参数] 用户名 文件名

常用选项：-R 递归修改文件或目录的所属组

实例：

|  |
| --- |
| # chown user1 file1 //改变file1文件所属用户为user1  # chown -R user1 filegroup1 //改变filegroup1目录的所有者为user1 |

注：在系统中，更改文件的拥有者需要root权限。

* + - 1. chgrp

功能：修改文件或目录的所属组

格式：chgrp [参数] 用户组名 文件名

常用选项：-R 递归修改文件或目录的所属组

实例：

|  |
| --- |
| #chgrp users file1 ////改变file1文件所属用户组为users |

注：更改文件的所属组也需要root权限。

* + - 1. umask（权限掩码）

功能：查看或修改文件掩码

新建文件夹默认权限=0666

新建目录默认权限=0777

但实际上你所创建的文件和目录，看到的权限往往不是上面这个值。原因就是创建文件或目录的时候还要受到umask的影响。假设默认权限是mask，则实际创建的出来的文件权限是: mask & ~umask

格式：umask 权限值

说明：将现有的存取权限减去权限掩码后，即可产生建立文件时预设权限。超级用户默认掩码值为0022，普通用户默认为0002。

注：凡是在umask中出现的权限，都必须在起始权限中去掉。

### 粘滞位

首先介绍一下目录的权限

* 可执行权限：如果没有目录的可执行权限，则无法在目录下执行权限，也无法进入目录
* 可读权限：如果没有目录的可读权限，则无法用ls等命令查看目录中的文件
* 可写权限：如果目录没有可写权限，则无法在目录中新增、删除文件、重命名等修改操作的权限

注：只要用户具有目录的写和执行权限，用户就可以随意删除目录中的文件，而不论这个用户是否有这个文件的写权限。

Linux系统中有很多人，我们需要一个公共的目录（比如tmp文件夹），进行临时文件的操作（增删查改），那么用户A在这个目录中创建的文件，用户B可以随意的删除，为了解决这个问题，Linux引入了粘滞位的概念。

|  |
| --- |
| #chmod 777 tmp //将目录的所有权限打开  #chmod +t tmp //加上粘滞位，此时的权限为：drwxrwxrwt |

所以加上粘滞位之后目录的其他人的执行权限变为rwt，所以当一个目录被设置为“粘滞位”时，该目录下的文件只能由

* root用户删除
* 目录所有者删除
* 该文件的所有者删除

# Linux环境基础开发工具

## Linux软件包管理器yum

## Linux编辑器vim

## Linux编译器gcc/g++

## Linux调试器gdb

## Linux项目自动化构建工具make/Makefile

## Linux第一个小程序-进度条

# Linux进程概念