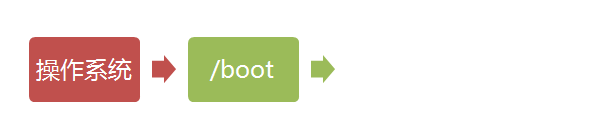
内核引导

当计算机打开电源后，首先是BIOS开机自检，按照BIOS中设置的启动设备（通常是硬盘）来启动。

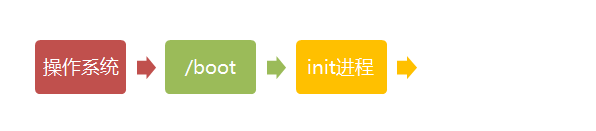
操作系统接管硬件以后，首先读入 /boot 目录下的内核文件。



运行init

init 进程是系统所有进程的起点，你可以把它比拟成系统所有进程的老祖宗，没有这个进程，系统中任何进程都不会启动。

init 程序首先是需要读取配置文件 /etc/inittab。



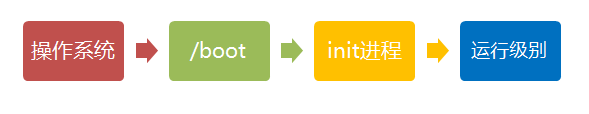
运行级别

许多程序需要开机启动。它们在Windows叫做"服务"（service），在Linux就叫做"守护进程"（daemon）。

init进程的一大任务，就是去运行这些开机启动的程序。

但是，不同的场合需要启动不同的程序，比如用作服务器时，需要启动Apache，用作桌面就不需要。

Linux允许为不同的场合，分配不同的开机启动程序，这就叫做"运行级别"（runlevel）。也就是说，启动时根据"运行级别"，确定要运行哪些程序。



Linux系统有7个运行级别(runlevel)：

运行级别0：系统停机状态，系统默认运行级别不能设为0，否则不能正常启动

运行级别1：单用户工作状态，root权限，用于系统维护，禁止远程登陆

运行级别2：多用户状态(没有NFS)

运行级别3：完全的多用户状态(有NFS)，登陆后进入控制台命令行模式

运行级别4：系统未使用，保留

运行级别5：X11控制台，登陆后进入图形GUI模式

运行级别6：系统正常关闭并重启，默认运行级别不能设为6，否则不能正常启动

系统初始化

在init的配置文件中有这么一行： si::sysinit:/etc/rc.d/rc.sysinit　它调用执行了/etc/rc.d/rc.sysinit，而rc.sysinit是一个bash shell的脚本，它主要是完成一些系统初始化的工作，rc.sysinit是每一个运行级别都要首先运行的重要脚本。

它主要完成的工作有：激活交换分区，检查磁盘，加载硬件模块以及其它一些需要优先执行任务。

l5:5:wait:/etc/rc.d/rc 5

这一行表示以5为参数运行/etc/rc.d/rc，/etc/rc.d/rc是一个Shell脚本，它接受5作为参数，去执行/etc/rc.d/rc5.d/目录下的所有的rc启动脚本，/etc/rc.d/rc5.d/目录中的这些启动脚本实际上都是一些连接文件，而不是真正的rc启动脚本，真正的rc启动脚本实际上都是放在/etc/rc.d/init.d/目录下。

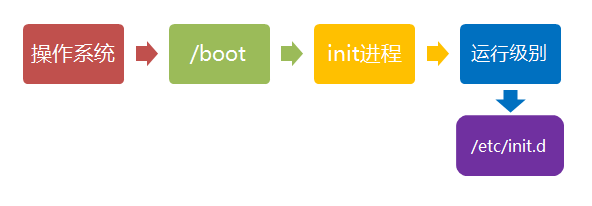
而这些rc启动脚本有着类似的用法，它们一般能接受start、stop、restart、status等参数。

/etc/rc.d/rc5.d/中的rc启动脚本通常是K或S开头的连接文件，对于以以S开头的启动脚本，将以start参数来运行。

而如果发现存在相应的脚本也存在K打头的连接，而且已经处于运行态了(以/var/lock/subsys/下的文件作为标志)，则将首先以stop为参数停止这些已经启动了的守护进程，然后再重新运行。

这样做是为了保证是当init改变运行级别时，所有相关的守护进程都将重启。

至于在每个运行级中将运行哪些守护进程，用户可以通过chkconfig或setup中的"System Services"来自行设定。



建立终端

rc执行完毕后，返回init。这时基本系统环境已经设置好了，各种守护进程也已经启动了。

init接下来会打开6个终端，以便用户登录系统。在inittab中的以下6行就是定义了6个终端：

1:2345:respawn:/sbin/mingetty tty1

2:2345:respawn:/sbin/mingetty tty2

3:2345:respawn:/sbin/mingetty tty3

4:2345:respawn:/sbin/mingetty tty4

5:2345:respawn:/sbin/mingetty tty5

6:2345:respawn:/sbin/mingetty tty6

从上面可以看出在2、3、4、5的运行级别中都将以respawn方式运行mingetty程序，mingetty程序能打开终端、设置模式。

同时它会显示一个文本登录界面，这个界面就是我们经常看到的登录界面，在这个登录界面中会提示用户输入用户名，而用户输入的用户将作为参数传给login程序来验证用户的身份。

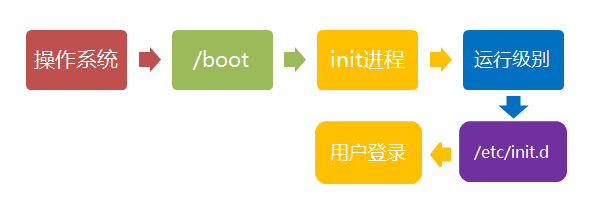
用户登录系统

一般来说，用户的登录方式有三种：

（1）命令行登录

（2）ssh登录

（3）图形界面登录



对于运行级别为5的图形方式用户来说，他们的登录是通过一个图形化的登录界面。登录成功后可以直接进入KDE、Gnome等窗口管理器。

而本文主要讲的还是文本方式登录的情况：当我们看到mingetty的登录界面时，我们就可以输入用户名和密码来登录系统了。

Linux的账号验证程序是login，login会接收mingetty传来的用户名作为用户名参数。

然后login会对用户名进行分析：如果用户名不是root，且存在/etc/nologin文件，login将输出nologin文件的内容，然后退出。

这通常用来系统维护时防止非root用户登录。只有/etc/securetty中登记了的终端才允许root用户登录，如果不存在这个文件，则root可以在任何终端上登录。

/etc/usertty文件用于对用户作出附加访问限制，如果不存在这个文件，则没有其他限制。

<="" p="">

图形模式与文字模式的切换方式

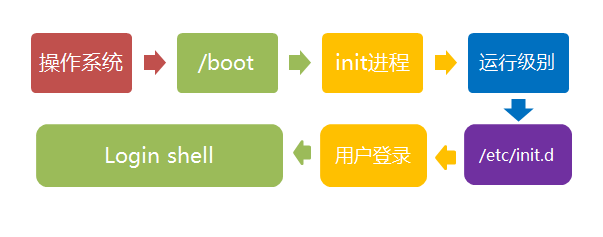
Linux预设提供了六个命令窗口终端机让我们来登录。

默认我们登录的就是第一个窗口，也就是tty1，这个六个窗口分别为tty1,tty2 … tty6，你可以按下Ctrl + Alt + F1 ~ F6 来切换它们。

如果你安装了图形界面，默认情况下是进入图形界面的，此时你就可以按Ctrl + Alt + F1 ~ F6来进入其中一个命令窗口界面。

当你进入命令窗口界面后再返回图形界面只要按下Ctrl + Alt + F7 就回来了。

如果你用的vmware 虚拟机，命令窗口切换的快捷键为 Alt + Space + F1~F6. 如果你在图形界面下请按Alt + Shift + Ctrl + F1~F6 切换至命令窗口。



Linux 关机

在linux领域内大多用在服务器上，很少遇到关机的操作。毕竟服务器上跑一个服务是永无止境的，除非特殊情况下，不得已才会关机。

正确的关机流程为：sync > shutdown > reboot > halt

关机指令为：shutdown ，你可以man shutdown 来看一下帮助文档。

例如你可以运行如下命令关机：

sync 将数据由内存同步到硬盘中。

shutdown 关机指令，你可以man shutdown 来看一下帮助文档。例如你可以运行如下命令关机：

shutdown –h 10 ‘This server will shutdown after 10 mins’ 这个命令告诉大家，计算机将在10分钟后关机，并且会显示在登陆用户的当前屏幕中。

Shutdown –h now 立马关机

Shutdown –h 20:25 系统会在今天20:25关机

Shutdown –h +10 十分钟后关机

Shutdown –r now 系统立马重启

Shutdown –r +10 系统十分钟后重启

reboot 就是重启，等同于 shutdown –r now

halt 关闭系统，等同于shutdown –h now 和 poweroff

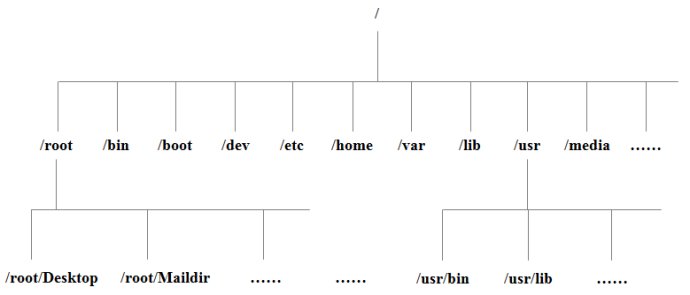
最后总结一下，不管是重启系统还是关闭系统，首先要运行 **sync** 命令，把内存中的数据写到磁盘中。

关机的命令有 **shutdown –h now halt poweroff** 和 **init 0** , 重启系统的命令有 **shutdown –r now reboot init 6**。

你会看到如下图所示:

[4_20](http://www.runoob.com/wp-content/uploads/2014/06/4_20.png)

树状目录结构：



以下是对这些目录的解释：

**/bin**：  
bin是Binary的缩写, 这个目录存放着最经常使用的命令。

**/boot：**  
这里存放的是启动Linux时使用的一些核心文件，包括一些连接文件以及镜像文件。

**/dev ：**  
dev是Device(设备)的缩写, 该目录下存放的是Linux的外部设备，在Linux中访问设备的方式和访问文件的方式是相同的。

**/etc：**  
这个目录用来存放所有的系统管理所需要的配置文件和子目录。

**/home**：  
用户的主目录，在Linux中，每个用户都有一个自己的目录，一般该目录名是以用户的账号命名的。

**/lib**：  
这个目录里存放着系统最基本的动态连接共享库，其作用类似于Windows里的DLL文件。几乎所有的应用程序都需要用到这些共享库。

**/lost+found**：  
这个目录一般情况下是空的，当系统非法关机后，这里就存放了一些文件。

**/media**：  
linux系统会自动识别一些设备，例如U盘、光驱等等，当识别后，linux会把识别的设备挂载到这个目录下。

**/mnt**：  
系统提供该目录是为了让用户临时挂载别的文件系统的，我们可以将光驱挂载在/mnt/上，然后进入该目录就可以查看光驱里的内容了。

**/opt**：  
 这是给主机额外安装软件所摆放的目录。比如你安装一个ORACLE数据库则就可以放到这个目录下。默认是空的。

**/proc**：  
这个目录是一个虚拟的目录，它是系统内存的映射，我们可以通过直接访问这个目录来获取系统信息。  
这个目录的内容不在硬盘上而是在内存里，我们也可以直接修改里面的某些文件，比如可以通过下面的命令来屏蔽主机的ping命令，使别人无法ping你的机器：

echo 1 > /proc/sys/net/ipv4/icmp\_echo\_ignore\_all

**/root**：  
该目录为系统管理员，也称作超级权限者的用户主目录。

**/sbin**：  
s就是Super User的意思，这里存放的是系统管理员使用的系统管理程序。

**/selinux**：  
 这个目录是Redhat/CentOS所特有的目录，Selinux是一个安全机制，类似于windows的防火墙，但是这套机制比较复杂，这个目录就是存放selinux相关的文件的。

**/srv**：  
 该目录存放一些服务启动之后需要提取的数据。

**/sys**：  
 这是linux2.6内核的一个很大的变化。该目录下安装了2.6内核中新出现的一个文件系统 sysfs 。

sysfs文件系统集成了下面3种文件系统的信息：针对进程信息的proc文件系统、针对设备的devfs文件系统以及针对伪终端的devpts文件系统。

该文件系统是内核设备树的一个直观反映。

当一个内核对象被创建的时候，对应的文件和目录也在内核对象子系统中被创建。

**/tmp**：  
这个目录是用来存放一些临时文件的。

**/usr**：  
 这是一个非常重要的目录，用户的很多应用程序和文件都放在这个目录下，类似于windows下的program files目录。

**/usr/bin：**  
系统用户使用的应用程序。

**/usr/sbin：**  
超级用户使用的比较高级的管理程序和系统守护程序。

**/usr/src：**内核源代码默认的放置目录。

**/var**：  
这个目录中存放着在不断扩充着的东西，我们习惯将那些经常被修改的目录放在这个目录下。包括各种日志文件。

在linux系统中，有几个目录是比较重要的，平时需要注意不要误删除或者随意更改内部文件。

/etc： 上边也提到了，这个是系统中的配置文件，如果你更改了该目录下的某个文件可能会导致系统不能启动。

/bin, /sbin, /usr/bin, /usr/sbin: 这是系统预设的执行文件的放置目录，比如 ls 就是在/bin/ls 目录下的。

值得提出的是，/bin, /usr/bin 是给系统用户使用的指令（除root外的通用户），而/sbin, /usr/sbin 则是给root使用的指令。

/var： 这是一个非常重要的目录，系统上跑了很多程序，那么每个程序都会有相应的日志产生，而这些日志就被记录到这个目录下，具体在/var/log 目录下，另外mail的预设放置也是在这里。

在 Linux 或 Unix 操作系统中，所有的文件和目录都被组织成以一个根节点开始的倒置的树状结构。

文件系统的最顶层是由根目录开始的，系统使用 / 来表示根目录。在根目录之下的既可以是目录，也可以是文件，而每一个目录中又可以包含子目录文件。如此反复就可以构成一个庞大的文件系统。

在Linux文件系统中有两个特殊的目录，一个用户所在的工作目录，也叫当前目录，可以使用一个点 . 来表示；另一个是当前目录的上一级目录，也叫父目录，可以使用两个点 .. 来表示。

. ：代表当前的目录，也可以使用 ./ 来表示；

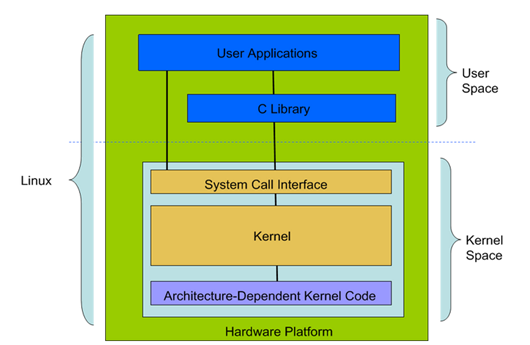
.. ：代表上一层目录，也可以 ../ 来代表。

如果一个目录或文件名以一个点 . 开始，表示这个目录或文件是一个隐藏目录或文件(如：.bashrc)。即以默认方式查找时，不显示该目录或文件

**Linux内核学习：详解内核的工作原理与关系**

内核是操作系统的核心部分，包含了系统运行的核心过程，决定系统的性能，操作系统启动内核被装入到RAM中;

　　操作系统与底层硬件设备交互和为运行应用程序提供执行环境。



[**Linux**](http://www.kokojia.com/list/273.html)**内核与微内核比较：**

　　微内核：内核只需要一个很小的函数集，通常包括几个同步原语，一个简单的调度程序和进程间通信机制。

　　运行在微内核之上的几个系统进程实现系统级功能：内存分配，设备驱动程序……完全的模块化进程。任何操作

　　系统层都是独立的程序模块，通过模块化的方法定义明确清晰的软件接口与其它层交互。内核中暂且不需要执行

　　的系统进程可以被调出或者撤销。微内核便于移植和充分利用RAM，但开销大效率是个问题。

　　宏内核：Linux内核：单块结构。内核的全部代码，包括所有子系统(如内存管理、文件系统、设备驱动程序)

　　都打包到一个文件中。内核中的每个函数都可以访问内核中所有其他部分。模块特性依赖于内核与用户层之间设计

　　精巧的通信方法，这使得模块的热插拔和动态装载得以实现。

　　每个内核层都被继承到整个内核程序中，并代表着当前进程在内核态下运行。

　　模块化(非进程)——允许在运行状态下动态的安装。模块是一个目标文件，其代码在运行时链接到内核或从内核解除链接。

　　目标代码通常是一组函数组成，用来实现文件系统，驱动程序……这些模块与其他静态链接内核函数一样，代表着当前进程

　　在内核态下执行，直接函数调用避免进程切换消息传递的开销，效率可能更高。

**Linux用户程序两种状态：**

　　用户态和内核态;用户态切换到内核态：

　　 进程系统调用

　　 CPU异常

　　 中断

　　 内核线程被执行

　　多用户系统：

　　能并发执行和独立的执行多个用户的应用程序，各个用户拥有独立空间。用户组，Root用户。

**Linux进程：**

　　进程Process：

　　 操作系统的基本抽象。

　　 进程是程序执行时的一个实例;一个运行程序的执行上下文。

　　 几个进程能并发的执行同一个程序;而同一个进程能顺序执行几个程序。

　　 具有独立的地址空间;多个进程可以同时执行。

　　进程受内核管理;每个进程由一个进程描述符表示，包含进程当前的状态信息。

　　当内核暂停一个进程的执行时，就把几个相关处理器寄存器的内容保存在进程描述符中。这些寄存器包括：

　　 程序计数器PC和栈指针SP寄存器

　　 通用寄存器

　　 浮点寄存器

　　 包含CPU状态信息的处理控制寄存器

　　 跟踪进程对RAM访问的内存管理寄存器

　　当内核恢复执行进程时：将进程描述符中合适字段来装在CPU寄存器，根据程序计数器指向恢复到程序执行的地方。

**Linux重入内核：**

　　内核可重入：

　　可重入函数：使用局部变量

　　实现同步机制：信号量、锁、关中断

　　进程执行状态切换: 进程在用户态与内核态的转换，Linux是抢占式内核

　　进程地址空间：每个进程运行在似有地址空间

　　同步和临界区：内核数据操作访问。

　　进程间通信IPC：信号量、消息队列、共享内存

　　进程管理：fork与\_exit,exec()，子进程与父进程

**Linux文件系统：**

　　文件系统是对存储设备上的数据和元数据进行组织的机制，以树形结构组织。

　　文件类型：

　　不同文件

　　目录

　　符号链接

　　面向块得设备文件 (设备驱动相关)

　　面向字符的设备文件 (设备驱动相关)

　　管道(pipe)和命名管道(named pipe)(进程间通信相关)

　　套接字(socket) (进程间通信相关)

　　文件访问权限和访问模式

　　文件描述符和索引节点：记录文件的信息数据。

　　文件操作的系统调用：open、read、write……

**内存管理：**

　　虚拟内存：处于应用程序内存请求与硬件内存单元之间的逻辑层。

　　随即访问存储器RAM：一部分用于内核映像，其余虚拟内存处理

　　内核内存分配器：KMA 处理内存请求子系统

　　 速度快

　　 减少内存浪费

　　 减轻内存碎片

　　 与其他内存管理合作(页框)

　　 内存分配算法

　　进程虚拟空间地址处理：内核分配给进程的虚拟地址空间由以下内存区组成：

　　 程序的可执行代码

　　 程序的初始化数据

　　 程序未初始化数据

　　 初始化程序栈

　　 所需共享库的可执行代码和数据

　　 程序动态请求的内存堆

　　高速缓存：

**设备驱动程序：**

　　内核通过设备驱动程序与I/O设备交互，设备驱动程序在内核中，用户程序通过内核访问设备。