



# linux 文件管理

## 目录篇

作者：七王爷

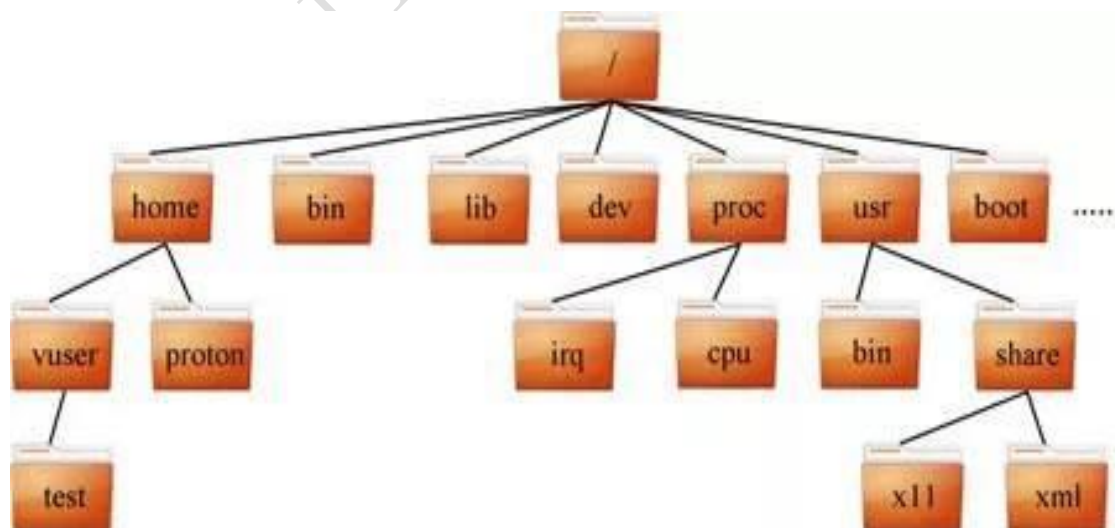
## 文件系统

对于计算机来说，所谓的数据就是 0 和 1 的序列。这样的一个序列可以存储在内存中，但内存中的数据会随着关机而消失。为了将数据长久保存，我们把数据存储在光盘或者硬盘中。根据我们的需要，我们通常会将数据分开保存到文件这样一个个的小单位中(所谓的小，是相对于所有的数据而言)。但如果数据只能组织为文件的话，而不能分类的话，文件还是会杂乱无章。每次我们搜索某一个文件，就要一个文件又一个文件地检查，太过麻烦。文件系统(file system)就是文件在逻辑上组织形式，它以一种更加清晰的方式来存放各个文件。

## Linux 的树结构

文件被组织到文件系统(file system)中，通常会成为一个树状(tree)结构。Linux 有一个根目录 /，也就是树状结构的最顶端。这个树的分叉的最末端都代表一个文件，而这个树的分叉处则是一个目录(directory，相当于我们在 windows 界面中看到的文件夹)。

要找到一个文件，除了要知道该文件的文件名，还需要知道从树根到该文件的所有目录名。从根目录开始的所有途径的目录名和文件名构成一个路径(path)。比如说，我们在 Linux 中寻找一个文件 file.txt，不仅要知道文件名(file.txt)，还要知道完整路径，也就是绝对路径 (/home/vamei/doc/file.txt)。



## Linux 目录标准（FHS）

由于 linux 是开放源代码，各大公司和团体根据 linux 的核心代码做各自的操作，编程。这样就造成在根下的目录的不同。这样就造成个人不能使用他人的 linux 系统的 PC

为了定义 Linux 目录规范，FHS（Filesystem Hierarchy Standard）机构诞生。该机构是 linux 爱好者自发的组成的一个团体，主要是对 linux 做一些基本的要求

根据 FHS(<http://www.pathname.com/fhs/>)的官方文件指出，他们的主要目的是希望让使用者可以了解到已安装软件通常放置于那个目录下，所以他们希望独立的软件开发商、操作系统制作者、以及想要维护系统的用户，都能够遵循 FHS 的标准。也就是说，FHS 的重点在于规范每个特定的目录下应该要放置什么样子的数据而已。这样做好处非常多，因为 Linux 操作系统就能够在既有的面貌下(目录架构不变)发展出开发者想要的独特风格。事实上，FHS 是根据过去的经验一直再持续的改版的，FHS 依据文件系统使用的频繁与否与是否允许使用者随意更动，而将目录定义成为四种交互作用的形态。

### 四种规范

	可分享的(shareable)	不可分享的(unshareable)
不变的(static)	/usr (软件放置处)	/etc (配置文件)
	/opt (第三方协力软件)	/boot (开机与核心档)
可变动的(variable)	/var/mail (使用者邮件信箱)	/var/run (程序相关)
	/var/spool/news (新闻组)	/var/lock (程序相关)

#### 🌈 可分享的：

可以分享给其他系统挂载使用的目录，所以包括执行文件与用户的邮件等数据，是能够分享给网络上其他主机挂载用的目录；

#### 🌈 不可分享的：

自己机器上面运作的装置文件或者是与程序有关的 socket 文件等，由于仅与自身机器有关，所以当然就不适合分享给其他主机了。

#### 🌈 不变的：

有些数据是不会经常变动的，跟随着 distribution 而不变动。例如函式库、文件说明文件、系统管理员所管理的主机服务配置文件等等；

✚ 可变动的：

经常改变的数据，例如登录文件、一般用户可自行收受的新闻组等。

事实上，FHS 针对目录树架构仅定义出三层目录底下应该放置什么数据而已，分别是底下这三个目录的定义：

**/ (root, 根目录)：与开机系统有关；**

**/usr (unix software resource)：与软件安装/执行有关；**

**/var (variable)：与系统运作过程有关。**

## Linux 文件系统基本常识

### 基本文件系统类型：

linux 有四种基本文件系统类型：

✚ 普通文件：

如文本文件、c 语言源代码、shell 脚本等，可以用 cat、less、more、vi 等来察看内容，用 mv 来改名；

✚ 目录文件：

包括文件名、子目录名及其指针，可以用 ls 列出目录文件；

✚ 链接文件：

是指向一索引节点的那些目录条目，用 ls 来查看时，链接文件的标志用 l 开头，而文件后以 "->" 指向所链接的文件；

✚ 特殊文件：如磁盘、终端、打印机等都在文件系统中表示出来，常放在 /dev 目录内；

### 可以用 file 命令来识别。

路径：

linux 系统中，所有的文件与目录都是由根目录/开始，不是以/开头的就是相对路径；

· ：表示当前目录，也可以用 ./ 表示；

.. ：表示上一级目录，也可以用 ../ 表示；

~ ：代表用户自己的宿主目录；

## 目录详细解析

### 根目录 (/) 的意义与内容:

根目录是整个系统最重要的一个目录，因为不但所有的目录都是由根目录衍生出来的，同时根目录也与开机/还原/系统修复等动作有关。由于系统开机时需要特定的开机软件、核心文件、开机所需程序、函式库等等文件数据，若系统出现错误时，根目录也必须要包含有能够修复文件系统的程序才行。因为根目录是这么的重要，所以在 FHS 的要求方面，他希望根目录不要放在非常大的分割槽内，因为越大的分割槽你会放入越多的数据，如此一来根目录所在分割槽就可能会有较多发生错误的机会。

因此 FHS 标准建议：根目录(/)所在分割槽应该越小越好，且应用程序所安装的软件最好不要与根目录放在同一个分割槽内，保持根目录越小越好。如此不但效能较佳，根目录所在的文件系统也较不容易发生问题。

/: 处于 Linux 文件系统树形结构的最顶端，我们称它为 Linux 文件系统的 root，它是 Linux 文件系统的入口。所有的目录、文件、设备都在/-之下，它是 Linux 文件系统最顶层的唯一的目录；

一般建议在根目录下面只有目录，不要直接存放文件；根目录是 linux 系统启动时系统第一个载入的分区，所以启动过程中用到的文件应该都放在这个分区中，其中/etc、/bin、/dev、/lib、/sbin 这 5 个子目录都应该要与根目录连在一起，不可独立成为某个分区；

### /bin:

存放所有用户都可以使用的 linux 基本操作命令；(目录中多是可执行的二进制文件)  
系统有很多放置执行文件的目录，但/bin 比较特殊。因为/bin 放置的是在单人维护模式下还能够被操作的指令。在/bin 底下的指令可以被 root 与一般账号所使用，主要有：cat, chmod, chown, date, mv, mkdir, cp, bash 等等常用的指令。

### /dev:

在 Linux 系统上，任何装置与接口设备都是以文件的型态存在于这个目录当中的。你只要透过存取这个目录底下的某个文件，就等于存取某个装置。比要重要的文件有/dev/null, /dev/zero, /dev/tty, /dev/lp\*, /dev/hd\*, /dev/sd\*等等

设备文件目录，虚拟文件系统，主要存放所有系统中 device 的相关信息，不论是使用的或未使用的设备，只要有可能使用到，就会在/dev 中建立一个相对应的设备文件；设备文件分为 2 种类型：字符设备文件和块设备文件(目录中基本上都是设备文件，如硬盘设备文件 /dev/sda)

/dev/console: 系统控制台，也就是直接和系统连接的监视器；  
/dev/hd: IDE 设备文件；  
/dev/sd: sata、usb、scsi 等设备文件；  
/dev/fd: 软驱设备文件；  
/dev/tty: 虚拟控制台设备文件；  
/dev/pty: 提供远程虚拟控制台设备文件；  
/dev/null: 所谓"黑洞"，所有写入该设备的信息都将消失，如当想要将屏幕上的输出信息隐藏起来时，只要将输出信息输入到/dev/null 中即可；

### **/boot:**

主要放置开机会使用到的档案，包括 Linux 核心档案以及开机选单与开机所需设定档等等。Linux kernel 常用的档名为: vmlinuz，如果使用的是 grub 这个开机管理程式，则还会存在/boot/grub/这个目录。

/boot: 存放开机启动加载程序的核心文件；(如 kernel 和 grub)

config-2.6.18-164.el5: 系统 kernel 的配置文件，内核编译完成后保存的就是这个配置文件；

lost+found: 说明/boot 是一个独立的 ext3 文件系统；

vmlinuz-2.6.18-164.el5: 系统使用 kernel，非常重要；

grub: 多系统启动管理程序 grub 的目录，里面存放的都是 grub 在启动时所需要的画面、配置及各阶段的配置文件；其中 grub.conf 是 grub 的配置文件；

symvers-2.6.18-164.el5.gz

initrd-2.6.18-164.el5.img: 此文件是 linux 系统启动时的模块供应主要来源，initrd 的目的就是在 kernel 加载系统识别 cpu 和内存等核心信息之后，让系统进一步知道还有那些硬件是启动所必须使用的；

System.map-2.6.18-164.el5: 是系统 kernel 中的变量对应表；(也可以理解为是索引文件)

### **/home:** 默认存放用户的宿主目录(除了 root 用户)

/home/~/.bashrc: 提供 bash 环境中所需使用的别名；

/home/~/.bash\_profile: 提供 bash 环境所需的变量；一般先执行.bashrc 后，才会再执行.bash\_profile；

/home/~/.bash\_history: 用户历史命令文件，记录用户曾经输入过的所有命令；(默认为 1000 条，可以通过 HISTSIZE 变量更改)

/home/~/.bash\_logout: 当用户注销的同时，系统会自动执行.bash\_logout 文件，

如果管理员需要记录用户注销的一些额外记录、动作或其他信息，就可以利用这个机制去完成；

这是系统默认的用户家目录(home directory)。在你新增一个一般使用者账号时，默认的用户家目录都会规范到这里来。比较重要的是，家目录有两种代号：

~：代表目前这个用户的家目录，而

~king：则代表 king 的家目录！

### /etc:

主机、系统或网络配置文件存放目录；

系统主要的配置文件几乎都放置在这个目录内，例如人员的账号密码文件、各种服务的启动档等等。一般来说，这个目录下的各文件属性是可以让一般使用者查阅的，但是只有 root 有权力修改。FHS 建议不要放置可执行文件(binary)在这个目录中。

### 🌈 基本文件

所有直接放在/etc 目录下的文件归类为基本文件；

aliases：用于设置邮件别名；

auto.\*：代表的是一系列 autofs 服务所需要的配置文件，这个服务主要是让管理员可以事先定义出一些网络、本机或光驱等默认的路径；

auto.master：负责规划目录的分配与使用，目前默认提供三种自动挂载模式；

auto.misc：文件中的配置都以实体连接本机的磁盘驱动器为主；

auto.net：并不是一个配置文件，而是一个脚本文件，在使用上其实不须做任何调整；；

auto.smb：与 auto.net 一样，都是以个脚本文件；

bashrc：用户登录功能配置，全局配置，对所有用户生效，主要配置别名；

profile：与系统环境配置或初始化软件的相关配置，全局配置，对所有用户生效，主要配置变量；

DIR\_COLORS：用于配置 ls 命令的颜色，主要针对 tty 登录的用户；

DIR\_COLORS.xterm：用于配置 ls 命令的颜色，主要针对 xterm 登录的用户；

fstab：系统启动时自动挂载文件系统的配置文件；

inittab：启动时系统所需要的第一个配置文件；也即是 init 进程的配置文件；

issue：用户本机登录时，看到的欢迎信息；

issue.net：用户网络登录时，看到的欢迎信息；

ld.so.conf：包含 ld.so.conf.d/\*.conf 配置；主要是 ld.so.conf.d/\*.conf 目录的作用；

localtime：系统所使用的时区对应的配置文件；对应的时区文件都存在于 /usr/share/zoneinfo/

motd：登录成功的用户显示的信息对应的配置文件；

mtab：可以当做是检查当前文件系统挂载情况的配置文件；与 mount 命令结果一致；

prelink.conf：定义哪些执行文件和函数库是需要预先连接的；

securetty：主要是 login 程序在使用的，只要是列在该文件中的接口，就表示是可以使用的接口，相反，若从列表中删除，则无法使用该接口；

shells：记录目前系统所拥有 shell 种类的路径，通过 cssh 命令使用；

sudoers：sudo 命令对应的配置文件，用于配置权限的分配方式；

sysctl.conf：主要是帮助用户配置/proc/sys 目录下所有文件的值，与 sysctl 命令对应；

syslogd.conf：是 syslogd 服务的配置文件

host.conf：主机名解析配置文件，主要说明解析的方式及顺序；

hosts：主机名解析配置文件，主要列出所有需要本地解析的主机名与 IP 地址的对应关系；

hosts.allow 和 hosts.deny：linux 网络安全机制 TCP Wrapper 对应的配置文件；

nsswitch.conf：主要记录系统应如何查询主机名、密码、用户组、网络等，或是查询顺序的编排；

resolv.conf：记录 DNS 服务器地址，用于 DNS 域名解析；

services：定义了网络服务的默认端口号；

xinetd.conf：xinetd 的主配置文件，目的是为 xinetd.d 下的所有子服务建立一个标准的规范使其可以遵循；

anacrontab：属于一种任务计划软件的配置文件，anacrontab 软件和 crond 其实有点相辅相成，crond 负责任务计划，而 anacrontab 则是负责以“间隔多久”为主要的目标；

at.deny：该文件属于拒绝列表，只要被记录在其中的用户，就无法使用 at 所提供的任务计划服务；

at.allow：与 at.deny 刚好相反；

crontab：crontab 的主配置文件，crond 默认会执行的文件可以参考此配置文件；

cron.deny：该文件属于拒绝列表，只要被记录在其中的用户，就无法使用 crond 所提供的任务计划服务；



cron.allow: 与 cron.deny 刚好相反;

exports: 是 NFS 服务的主配置文件, 主要目的就是为本机的目录共享到网络上, 供其他人使用;

group 与 gshadow: 用户组配置文件, group 主要保存用户组信息, gshadow 主要保存群组密码;

login.defs: 设置系统在建立账号时所参考的配置;

passwd: 主要保存系统用户账号的信息;

shadow: linux 系统通常包经过 "hash" 处理后的密码存储在这个文件中;

protocols: 通信协议对应端口号的一个对照表, 包含协议名称、协议号码、注释等;

wgetrc: wget 程序对应的配置文件, 其中有 quota、mail header、重传文件的预设次数、firewall 和 proxy 等相关设置;

init.d: RHEL 中所有服务的默认启动脚本都存放在这里; 这个是链接文件, 链接到/etc/rc.d/init.d;

csh.cshrc 和 csh.login: 用户启动 c shells 执行的初始化配置文件;

printcap: linux 系统中打印机设备对应的配置文件;

### 服务器目录

如 samba、http、vsftpd 等服务器配置相关目录;

cups: linux 下的打印机服务器, 目录下存放的是打印机服务的配置文件;

dnsmasq.d: dnsmasq 是一种 DNS 的 "轻薄机种", 转为区域或小型网络所设计, 拥有比一般 DNS 更为方便简易的配置;

httpd: apache 网页服务器的配置文件所在目录;

mail: Mail Server 组件的主要配置目录, 如 sendmail;

ntp: 网络时间服务器的配置目录, 其主要配置文件为/etc/ntp.conf;

openldap: 目录明显是 LDAP 的配置目录, 软件名称为 OpenLDAP;

postfix: postfix 组件所提供的主要配置文件目录;

samba: 文件共享服务 samba 的主要配置文件目录;

smrsh: 这是 sendmail 为了限制用户可使用的命令设计的程序, 将原本用户所使用的/bin/sh 替换为/usr/sbin/smrsh;

snmp: 简单网络管理软件的配置文件目录, 存在 snmpd.conf 主配置文件;

squid: 这是 linux 下的代理服务器 squid 的配置文件目录, 主配置文件是 squid.conf;

ssh: SSH 服务的主要配置目录, 主配置文件是 sshd\_config;

vsftpd: vsftpd 服务器的主要配置目录，主配置文件是 vsftpd.conf;

xinetd.d: xinetd 是一个管理多个服务的 daemon，这个目录下列出的服务都是由 xinetd 进程管理的，其主配置文件是/etc/xinetd.conf;

### 🌈 系统目录:

如 sysconfig、xen 或网络配置等与系统运行相关的目录;

blkid: 此目录所存放的其实是一个块设备 ID 的临时文件，主要是记录系统中所有区块设备的标签名称、硬件的唯一识别码、文件系统的格式等基本信息;

bluetooth: linux 下使用蓝牙设备所需的配置文件; 启动蓝牙检测的主要服务仍是/etc/rc.d/init.d/bluetooth，该程序使用的是 hcid.conf 配置文件;

cron.X: cron.X 的目录都是给 cron 软件存放其需要任务计划的文件所使用的，按任务计划时间的长短及配置特性分为 cron.d、cron.daily、cron.hourly、cron.monthly、cron.weekly 五个主要目录;

dbus-1: D-BUS 的主要配置目录，D-BUS 也是一种 IPC 交流的方式;

default: 这里是存放一些系统软件默认值的目录，存放某些软件执行时的基本参数;

firmware: 这个目录所存放的东西是非常底层的信息，是 CPU 所需的 microcode 的实体文件;

foomatic: 与打印机相关的配置目录，实现打印一对多的方式，在 foomatic 中，可以记录多条打印机数据，让用户只在使用前先行配置所有需要使用的打印机即可;

hal: 全名 Hardware Abstraction Layer，是 linux 一种管理硬件的机制，它会帮所有的应用程序或用户搜集所有 PCI 及 USB 等硬件信息，因此，用户可以很简单并实时地通过 HAL 的方式取得硬件的相关数据;

isdn: ISDN 服务的主要配置目录，里面包含可拨号的用户、电话、联机方式等;

ld.so.conf.d: 这个目录是 ldconfig 所使用的，更准确的说，它是由 /etc/ld.so.conf 文件所决定的; ldconfig 命令的目的在于将系统中的一些函数库预先存放到内存中，让系统使用时可以比以往通过硬盘的读取速度来的更快，这样可以大幅提高系统性能，尤其当要重复读取时更明显; ldconfig 要将哪些函数库丢到内存中，则须看 /etc/ld.so.conf 文件中所记录的信息;

logrotate.d: 此目录对系统管理员来说，是十分重要的一个目录，因为目录中的文件，记录了如何定期备份系统所需要备份的系统或软件日志文件及备份方式，目录是由 logrotate 组件所提供的，而里面所有文件是由各软件各自产生的; 其主要配置文件是 /etc/logrotate.conf;

logwatch: logrotate 主要是实现如何备份日志文件，这个目录就是记载如何分析日志文件并告诉用户的软件 logwatch 的配置目录;

lsb-release.d: LSB 是一个由很多人所执行的项目，其目的是将所有的 Linux 发行版定义为一些共同的标准；

lvmm: 这个目录是 LVM 的基本配置文件，但配置或操作一般都只需要通过 LVM 提供的命令，而不会用到这个目录，除非要使用到很高级的配置才会更改此文件；

makedev.d: MAKEDEV 软件对应的配置文件目录，MAKEDEV 主要用来产生设备文件，也就是说，在/dev 目录下的文件都由这个命令产生的，此目录下的文件主要是针对设备文件的定义或属性，目录中存在的设备文件可以由 MAKEDEV 来创建，否则需要使用 mknod 命令了；

modprobe.d: 是 modprobe 命令的住配置目录，一般系统启动默认要加载的模块放在/etc/modprobe.conf 中；

netplug 和 netplug.d: 这两个目录和网络接口的联机与否由直接关系，因为主要是控制联机时的接口操作；

opt: 此目录原本是定义为存放所有额外安装软件的主机配置文件，但目前并没有被使用到，此目录为空；

pcmcia: 这是 PCMCIA 的配置文件目录，PCMCIA 是笔记本电脑不可或缺的接口，需要即插即用的方式，此接口使用较少；

pm: 由 pm-utils 组件所提供的目录，pm-utils 是一套电源管理的工具软件，其中/usr/lib/pm-utils 也是主要目录之一；

ppp: ppp 相关的配置文件都放在这个目录中；

profile.d: 这个目录存放的是系统部分的软件配置，但会按不同的 shell 执行不同的文件，默认所使用的 bash 会直接执行该目录下所有扩展名为.sh 的文件；

rc.d: 主要用来定义在每一个执行阶段必须要执行哪些系统服务或程序，在目录中主要分为三个重要的部分：

--rc.sysinit: 系统一开始启动时所遇到的第一个文件，此脚本文件记录服务启动之前所需准备的所有事情，包括启动时看到的欢迎画面；

--rcX.d: 在 rc.sysinit 文件之后所要执行的，X 是系统启动时的 initdefault 值，值为几则会转到那个目录下，并执行其中的所有文件，在此目录中，文件一律都由两个英文字母开始 K 和 S，K 代表 kill，S 代表 Start；

--rc.local: 系统初始化过程中最后一个执行的脚本文件，可以将需要开机启动的程序或脚本放置在这个脚本文件中，以实现自动运行的目的；

readahead.d: 是 readahead 程序的主要配置目录，为了加速操作系统的使用速度，readahead\_early 和 readahead\_later 这两个进程在系统加载时，直接将日常所需要的一些文件，全部先放到硬盘的高速缓存中；

redhat-lsb: 都 lsb-release.d 目录都是由程序 redhat-lsb 所提供的；

rwtab.d: 这个目录是一个在启动时会去参考的目录, 主要的文件在/etc/rwtab; 这是一个系统初期的备份机制;

sane.d: 这是在系统下要使用扫描仪所需的配置目录, 主要配置文件是 sane.conf, sane 为了方便用户在各式的扫描仪连接时都可以使用, 因此, 在这一目录中放置了很多种不同类型扫描仪的硬件信息, 让系统在检测到扫描仪时可以直接使用;

setuptool.d: 这个目录是"setup"系统配置工具的主要配置目录;

skel: 用于初始化用户宿主目录的配置目录, 当建立一个用户时, 会把此目录下的所有文件复制一份到用户的宿主目录, 作为用户的初始化配置;

sysconfig: 非常重要的系统配置文件的存放目录, 里面放置了大量系统启动及运行相关的配置文件;

sysconfig/network-scripts/ifcfg-eth0: 网卡 eth0 对应的配置文件, 设置内容包括设备名称、IP 地址、广播地址、网关地址、网段、开机是否激活等参数

udev: udev 程序本身是一套设备的管理机制, udev 通过 sysfs 的文件系统, 可以正确地掌握目前系统上存在的硬件设备, 以及针对每一个硬件设备做出不同的判断与执行;

yum 和 yum.repos.d: 这两个都是 yum 的配置目录, 是一套在 linux 下可以自动帮助用户安装、更新、移除等的管理组件, 可用来替代 rpm 包管理方式, 主配置文件是 /etc/yum.conf; yum 是更新方式及外挂程序的配置目录, yum.repos.d 是存放定期更新组件内容的信息;

### 安全性目录

如 selinux 或 pam.d 等管理系统安全性的目录;

audit: 这个目录所代表的是一种和目录名称一致的 audit 安全机制, 主要以服务的方式协助管理员持续监控各文件被存取的情况; 目录下的 audit.rules 文件主要是定义一些必要的监控规则;

pam.d: 此目录是 Linux-PAM 的所有配置文件, 配合/lib/security 目录中所有觉得函数库, 提供 Linux 下的应用程序认证的机制;

pam\_pkcs11: PAM 机制中的一种登录模块, 可以让用户通过 smart card 做登录的操作;

pki: PKI 是一种公开密钥的管理方式, 通过这样的管理模式, 可以让所有网络传输有更多保障;

racoon: 这个目录是由 ipsec-tools 组件所提供的, ipsec 的主要目的是让系统实现 VPN 的网路技术, 在 racoon 目录的主配置文件 racoon.conf 中, 定义在 ipsec 操作中所需要的加密算法种类以及其他细节的配置;

security: 与 pam.d 目录相辅相成, pam.d 中的所有 PAM 的规则都要用到 /lib/security 下的 PAM 函数库, 而/etc/security 目录中, 就是针对这些函数库, 提供以配置文件的方式进行细节配置, 对希望调整系统安全性部分增加了非常大的方便性;

selinux: selinux 是一个很新的安全性方案, 它是一种针对各种文件、目录、设备或 daemon 等在 linux 所需使用到的安全性机制, 而且其安全性的数据时直接记录在文件系统中;

wpa\_supplicant: 这个目录被归类到安全性目录中, 是因为其属于无线中安全认证的部分, 存在 wpa\_supplicant.conf 配置文件, 用户可以在这个目录中加入已知可登陆的 AP;

### X Windows 目录

如 X11 或 gdm 管理 X windows 启动或使用上的配置目录;

alternatives: linux 下可辨识扩展名的"文件类型"选项, 可以针对同一类型的文件, 选出一个默认用户所要使用的程序去执行; /etc/alternatives 目录下有所有目前已经定义的程序名称, 都以软链接的方式存在, 里面每一个文件其实都有定义好的默认执行程序, 可以使用 alternatives 命令查看及修改配置;

fonts: 这个目录就是 fontconfig 软件的最主要配置目录, 其中 /etc/fonts/fonts.conf 就是对应的配置文件, /etc/fonts 目录下的配置都是以 XML 的方式配置的;

gconf: 这一目录是 GConf2 的组件所建立的, GConf 的作用就是提供 GNOME 下的应用程序注册的机制, 有些类似于 windows 下的 regedit;

gdm: 全名为 GNOME Display Manger, 也就是协助 X Windows 启动的管理软件, 在 GDM 中的主配置文件是 custom.conf, 在 X windows 下可以利用 gdmsetup 命令对这个文件进行配置;

gnome-vfs-2.0: GNOME VFS 机制, 让 GNOME 的系统可以知道每一种文件格式要如何开启或浏览, 而所有的配置都需要有相对应的函数库;

gtk-2.0: 由 gtk+ 组件提供的目录, 主要是提供 X Windows 窗口的颜色、按钮或图案, 包含软件选项的画面、选项的按钮、滚动轴的样式等;

kde: KDE Desktop Manager 的主要配置目录; 主配置文件是 kdmrc;

NetworkManager: 此目录的目的是让用户不需要做任何操作和配置, 只要用户曾经登陆过无线 AP, 系统就可以记录下来, 以后再次登陆时就可以方便的登陆;

pango: pango 是一套协助 GTK+ 将字体描绘出来的函数库, 不论任何的字体或语言, 都可以通过 pango 描绘出来;

rhgb: 系统在进入 X Windows 之前, 有一个前置配置的图形接口, 这个接口

就是 rhgb, 其主要目的是让系统启动变得漂亮;

scim: 是 Linux 下目前很好用的输入法;

sound: GNOME 下有许多的应用软件, 很多都会有其特殊的声音, 这个目录中存放所有声音的命令路径;

X11: X windows 的核心配置目录; 该目录下比较重要的文件有 prefddm(判断 X windows 使用哪一个 Display Manager)、主配置文件 xorg.conf(定了 X windows 所需使用的键盘、鼠标、显卡等相关硬件设备, 重点是关于显卡的配置)、xinit 子目录(里面都是一些 X windows 资源相关的配置)

xdg: X windows 上的菜单画面, 就是从这里出来的, 所有在 X windows 中使用的菜单文字及分类, 都可以在这个目录下做配置, 其下的子目录 menu, 可以通过配置里面的文件自定义应用程序、系统管理、外观等菜单内容

### 其他目录

针对单一特殊软件的配置或未能按以上分类方式则放在此目录中;

a2ps.cfg 和 a2ps-site.cfg: 用于将一份文件格式转换为 postscript 的格式, 在某些打印机或要将文件输出成一份标准格式的文件时, 它会被用到;

alsa: 主要任务在于提供 linux 声音及声音的功能, 并试着让其性能达到最佳化;

ghostscript: 在 linux 下要读取 Adobe 格式文件(如 pdf), 最方便的方式就是使用 ghostscript 命令, 这个目录主要用于设置在显示时使用哪种字体作为默认字体;

gre.d: GRE 是 Mozilla 注册的一种机制, 目录中的配置文件 gre.conf 会注明所使用的 Mozilla 软件的路径和版本;

iproute2: iproute2 是一套非常强大的网络管理软件, iproute2 提供的功能有很多种, 此目录中存放一些网络的基本配置值;

java: 这个目录是由 jpackage-utils 软件提供的, 这个目录是这个软件的主要配置目录, 除此之外还有 maven、jvm、jvm-common 都是由 jpackage-utils 软件产生的, jpackage 是一个专门为了提供 java 程序与函数库所存在的软件;

mgetty+sendfax: 主要用于使用 linux 架构一台 fax server, 可以使用 mgetty.config 来配置需要有关传真接收和发送的操作;

php.d: 主要存放的各软件(如 dbase、ldap、mysql 等)与 php 相关的配置文件;

reader.conf.d: 存放 smart card 配置文件的目录, 由程序 pcsc-lite 提供, 这个程序的主配置文件是/etc/reader.conf;

dumpdates: 存放 dump 命令的执行日期, dump 命令可以对 ext2/ext3 文件系统进行检查备份;



### **/media:**

移动存储设备默认挂载点；(如光盘)

media 是『媒体』的英文，顾名思义，这个/media 底下放置的就是可移除的装置！包括软盘、光盘、DVD 等等装置都暂时挂载于此。常见的档名有：/media/floppy, /media/cdrom 等等。

### **/mnt:**

临时挂载用的设备挂载点；(如磁盘分区，网络共享)

如果你想要暂时挂载某些额外的装置，一般建议你可以放置到这个目录中。最开始，这个目录的用途与/media 相同！只是有了/media 之后，这个目录就用来暂时挂载用了。

### **/lib:**

需要共享的函数库与 kernel 模块,系统 kernel 启动所使用的函数库,或者当执行一些在/bin 和/sbin 中的命令时使用的函数库;

系统的函式库非常的多,而/lib 放置的则是在开机时会用到的函式库,以及在/bin 或/sbin 底下的指令会呼叫的函式库而已。什么是函式库呢?可以将他想成是『外挂』,某些指令必须要有这些『外挂』才能够顺利完成程序的执行之意。尤其重要的是/lib/modules/这个目录,因为该目录会放置核心相关的模块(驱动程序)!

**/opt:** 额外所安装的应用程序目录,有些软件包我们可以将它安装在该目录中;(一般为空,某些应用软件安装需要这个目录)

这个是给第三方协力软件放置的目录。什么是第三方协力软件?举例来说,KDE 这个桌面管理系统是一个独立的计划,不过他可以安装到 Linux 系统中,因此 KDE 的软件就建议放置到此目录下了。另外,如果你想要自行安装额外的软件(非原本的 distribution 提供的),那么也能够将你的软件安装到这里来。不过,以前的 Linux 系统中,我们还是习惯放置在 /usr/local 目录下!

### **/root:**

管理员 root 的宿主目录

系统管理员(root)的家目录。之所以放在这里,是因为如果进入单人维护模式而仅挂载根目录时,该目录就能够拥有 root 的家目录,所以我们会希望 root 的家目录与根目录放置在同一个分割槽中。

## **/sbin:**

系统管理相关的二进制文件存放在这个目录下,一些可执行文件普通用户只具备较小的权限;(多数管理命令默认只有管理员可以使用)

Linux 有非常多指令是用来设定系统环境的,这些指令只有 root 才能够利用来『设定』系统,其他用户最多只能用来『查询』而已。放在/sbin 底下的为开机过程中所需要的,里面包括了开机、修复、还原系统所需要的指令。至于某些服务器软件程序,一般则放置到/usr/sbin/当中。至于本机自行安装的软件所产生的系统执行文件(system binary),则放置到/usr/local/sbin/当中了。常见的指令包括:fdisk, fsck, ifconfig, init, mkfs 等等。

## **/srv:**

默认为空,主要用于存放一些软件的配置文件,某些软件可能会把配置文件默认存放在这个目录下,多数都是/etc 目录下,此目录没有被具体的定义;

srv 可以视为『service』的缩写,是一些网络服务启动之后,这些服务所需要取用的数据目录。常见的服务例如 WWW, FTP 等等。举例来说,WWW 服务器需要的网页数据就可以放置在/srv/www/里面。

## **/tmp:**

临时文件存放区域;(默认被设置了粘滞位)

这是让一般使用者或者是正在执行的程序暂时放置文件的地方。这个目录是任何人都能够存取的,所以你需要定期的清理一下。当然,重要数据不可放置在此目录! 因为 FHS 甚至建议在开机时,应该要将/tmp 下的数据都删除!

## **Linux 目录拓展标准**

### **/sys:**

虚拟文件系统,被建立在内存中,是在 2.6 版的 kernel 之后才被加入到正式的文件系统中,以分类的方式将系统的信息存放在这个目录中,以方便 linux 用户通过不同的分类找出系统相关的信息;

这个目录其实跟/proc 非常类似,也是一个虚拟的文件系统,主要也是记录与核心相关的信息。包括目前已加载的核心模块与核心侦测到的硬件装置信息等等。这个目录同样不占硬盘容量

### **/lost+found**

这个目录是使用标准的 ext2/ext3 文件系统格式才会产生的一个目录,目的在于当文件系统发生错误时,将一些遗失的片段放置到这个目录下。这个目录通常会在分割槽的最顶层存



在，例如你加装一颗硬盘于/disk 中，那在这个系统下就会自动产生一个这样的目录『/disk/lost+found』

**/proc:** 虚拟文件系统，此目录是 kernel 加载后，

在内存里面建立的一个虚拟目录，有专属的文件系统，主要提供系统一些实时的信息，此目录下不能建立和删除文件；(某些文件可以修改)

这个目录本身是一个『虚拟文件系统(virtual filesystem)』！他放置的数据都是在内存当中，例如系统核心、行程信息(process)、周边装置的状态及网络状态等等。因为这个目录下的数据都是在内存当中，所以本身不占任何硬盘空间！比较重要的文件例如：/proc/cpuinfo, /proc/dma, /proc/interrupts, /proc/ioports, /proc/net/\* 等等。

/proc 主要作用可以整理为：

- 整理系统内部的信息；
- 存放主机硬件信息；
- 调整系统执行时的参数；
- 检查及修改网络和主机的参数；
- 检查及调整系统的内存和性能；

/proc 下常用的信息文件有：

/proc/cpuinfo: cpu 的硬件信息，如类型、厂家、型号和性能等

/proc/devices: 记录所有在/dev 目录中相关的设备文件分类方式

/proc/filesystems: 当前运行内核所配置的文件系统

/proc/interrupts: 可以查看每一个 IRQ 的编号对应到哪一个硬件设备

/proc/loadavg: 系统"平均负载"，3 个数据指出系统当前的工作负载

/proc/dma: 当前正在使用的 DMA 通道

/proc/ioports: 将目前系统上所有可看到的硬件对应到内存位置的分配表的详细信息呈现出来

/proc/kcore: 系统上可以检测到的物理内存，主机内存多大，这个文件就有多大

/proc/kmsg: 在系统尚未进入操作系统阶段，把加载 kernel 和 initrd 的信息先记录到该文件中，后续会将日志信息写入/var/log/message 文件中

/proc/meminfo: 记录系统的内存信息

/proc/modules: 与 lsmod 命令查看到的模块信息完全一致

/proc/mtrr: 负责内存配置的机制

/proc/iomem: 主要用于储存配置后所有内存储存的明细信息

/proc/partitions: 这个文件可以实时呈现系统目前看到的分区

/proc/数字目录: 数字目录很多，它们代表所有目前正在系统中运行的所有程序

**/proc/bus:** 有关该主机上现有总线的所有信息，如输入设备、PCI 接口、PCMCIA 扩展卡及 USB 接口信息

**/proc/net** 目录：存放的都是一些网络相关的虚拟配置文件，都是 ASCII 文件，可以查看(与 ifconfig、arp、netstat 等有关)

**/proc/scsi:** 保存系统上所有的 scsi 设备信息(包括 sata 和 usb 设备的信息)

**/proc/sys** 目录：存放系统核心所使用的一些变量，根据不同性质的文件而存放在不同的子目录中，可以通过/etc/sysctl.conf 文件设置和更改其默认值；变量时实时的变更，有很多设置很象是开关，设置后马上生效；

**/proc/tty:** 存放有关目前可用的正在使用的 tty 设备的信息

**/proc/self:** 存放查看/proc 的程序的进程目录的符号连接，当 2 个进程查看 proc 时，这将会是不同的连接；主要便于程序得到它自己的进程目录；

**/proc/stat:** 系统的不同状态信息；

**/proc/uptime:** 系统启动的时间长度；

**/proc/version:** 系统核心版本；

### **/usr:**

安装除操作系统本身外的一些应用程序或组件，一般可以认为 linux 系统上安装的应用程序默认都安装在此目录中；

如果/usr 是安装时会占用较大硬盘容量的目录，那么/var 就是在系统运作后才会渐渐占用硬盘容量的目录。因为/var 目录主要针对常态性变动的文件,包括缓存(cache)、登录档(log file)以及某些软件运作所产生的文件，包括程序文件(lock file, run file), 或者例如 MySQL 数据库的文件等等。常见的次目录有：

**/usr/bin:** 一般用户有机会使用到的程序，或者该软件默认就是要让所有用户使用才会放在该目录中；

**/usr/sbin:** 一些系统有可能会用到的系统命令，与/sbin 比起来，都是一些较次要的文件；

**/usr/etc:** 自行安装或非系统主要的配置文件目录；

**/usr/games:** 只要是电脑游戏相关的软件，就都安装到这个目录；

**/usr/include:** 存放的文件都是一些系统中用户所会使用到的 C 语言 header 文件，保存的都是".h"的文件；

**/usr/kerberos:** kerberos 是一种安全机制，让用户可以直接使用支持 kerberos 机制系统上的部分资源；

**/usr/lib:** 存放一些函数库、执行文件及连接文件，特别的是，存放在这里面的文件都是不希望直接被用户或 shell 脚本所使用的文件，在/usr/lib 中有非常多的子目录，每一

个软件都有其各自所需的函数库；

/usr/libexec：这个目录下的文件及文件夹应该都可以放置在/usr/lib 下；

/usr/local：linux 系统中安装的共享软件程序最好的方式是安装在/usr/local 下，按照 linux 标准目录结构，新建立的软件都应该放在/usr/local 下；

/usr/local/bin：存放软件执行文件的目录；

/usr/local/sbin：同样存放软件执行文件的目录，但此目录专门针对系统所使用的文件；

/usr/local/lib：软件相关的函数库；

/usr/local/share：当文件性质不好归属时就会放在此，man 手册就放在这个目录下；

/usr/local/src：所安装软件的源代码放置在此；

/usr/share：此目录都是一些共享信息，最常被用到的就是/usr/share/man 这个目录，/usr/share 里的信息时跨平台的；

/usr/share/doc：放置一些系统帮助文件的地方；

/usr/share/man：manpage 的文件存放目录，也是使用 man 查看手册页时查询的路径；

/usr/src：主要储存内核源代码的文件；

/usr/X11R6：存放一些 X windows 系统的相关文件；

**/var**：动态文件或数据存放目录，默认日志文件都存放在这个目录下，一般建议把此目录单独划分一个分区；

依据 FHS 的基本定义，/usr 里面放置的数据属于可分享的与不可变动的(shareable, static)，如果你知道如何透过网络进行分割槽的挂载(例如在服务器篇会谈到的 NFS 服务器)，那么 /usr 确实可以分享给局域网络内的其他主机来使用！

很多读者都会误会/usr 为 user 的缩写，其实 usr 是 Unix Software Resource 的缩写，也就是『Unix 操作系统软件资源』所放置的目录，而不是用户的数据！这点要注意。FHS 建议所有软件开发者，应该将他们的数据合理的分别放置到这个目录下的次目录，而不要自行建立该软件自己独立的目录。

因为是所有系统默认的软件(distribution 发布者提供的软件)都会放置到/usr 底下，因此这个目录有点类似 Windows 系统的『C:\Windows\ + C:\Program files\』这两个目录的综合体，系统刚安装完毕时，这个目录会占用最多的硬盘容量。

/var/account：是 linux 系统下的审核机制(psacct)对应的目录；

/var/cache: 该目录下的文件是所有程序所产生的缓存数据，也就是当应用程序启动时，会将数据留一份在这个目录中；

/var/empty: 默认是 sshd 程序用到的这个目录，当建立 ssh 连接，ssh 服务器必须使用该目录下的 sshd 子目录；

/var/ftp: ftp 服务器软件一般默认会将匿名登陆的用户的宿主目录；

/var/gdm: gdm 所使用的目录，里面存放一些系统当前所占用的 console 记录及通过 gdm 执行的 X windows 记录，只有通过 gdm 窗口的日志才会存放在此；

/var/lib: 该目录下存放很多与应用程序名称同名的子目录，每个子目录下都是应用执行的状态信息；

/var/lock: 每个服务一开始都会在这个目录下产生一个该服务的空文件，主要是避免服务启动冲突；

/var/log: 常用目录，专门用来存放所有日志文件的目录，里面存放很多系统、软件、用户等相关的日志信息；里面有一些文件是比较常用的；

lastlog: 记录用户最后一次登录的信息，使用 lastlog 命令读取；

message: 记录系统的几乎所有信息，主要包括启动信息，syslogd 服务记录的信息等；

wtmp: 记录所有用户登陆及注销的信息，使用 last 命令读取；

secure: 记录登录系统访问数据的文件，如 ssh pop3 telnet ftp 等都会记录在此文件中

/var/log/httpd/access\_log: httpd 访问日志

/var/log/httpd/error\_log: httpd 错误日志

btmp: 记录失败的用户登录

utmp: 纪录当前登录的每个用户

xferlog: ftp 会话日志

boot.log: 记录开机或一些服务启动时所显示的启动和关闭信息

/var/log/maillog 或 /var/log/mail/\*: 记录邮件访问或往来的用户信息

cron: 记录 crontab 例行性服务的内容

dmesg: 开机引导日志信息

sudolog: 纪录使用 sudo 发出的命令

sulog: 纪录使用 su 命令的使用

/var/named: bind 软件实现的 DNS 服务器的区域数据文件都存放在这个目录下；

/var/nis 和 /var/yp: 都是 NIS 服务机制所使用的目录，nis 主要记录所有网络中每一个 client 的连接信息；yp 是 linux 的 nis 服务的日志文件存放的目录；

/var/run: 此目录中的大部分文件都记载目前系统正在执行程序的 PID 值，每一个文

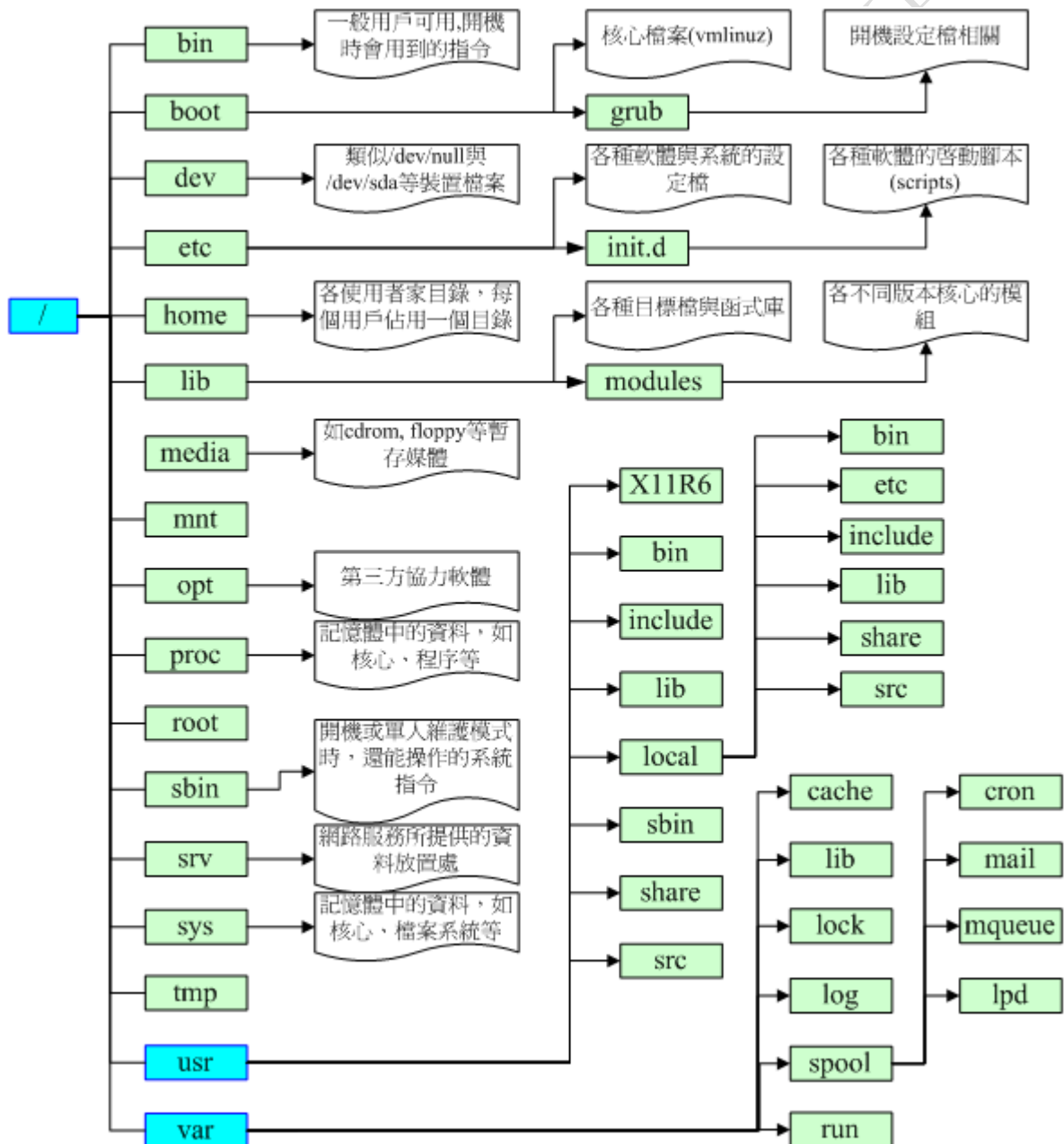
件都是以个独立的 PID 记录；此目录下存放一个特殊文件 utmp，此文件记录目前谁在使用系统，必须使用 utmpdump 命令才能看到其中的内容；

/var/spool：里面主要都是一些临时存放，随时会被用户所调用的数据；打印机、邮件、代理服务器等假脱机目录存放在该目录下；

/var/tmp：专门为了某些应用程序在安装或执行时，需要在重启后使用的某些文件时，能将该文件暂时存放在这个目录中，完成后再行删除；

/var/www：apache 网页服务器的宿主目录；

## 注意事项：



除了这些目录的内容之外，另外要注意的是，因为根目录与开机有关，开机过程中仅有根目录会被挂载，其他分割槽则是在开机完成之后才会持续的进行挂载的行为。就是因为如此，因此根目录下与开机过程有关的目录，就不能够与根目录放到不同的分割槽去！那哪些目录不可与根目录分开呢？有底下这些：

- /etc: 配置文件
- /bin: 重要执行档
- /dev: 所需要的装置文件
- /lib: 执行档所需的函式库与核心所需的模块
- /sbin: 重要的系统执行文件

在 Linux 底下，所有的文件与目录都是由根目录开始的！那是所有目录与文件的源头，然后再一个一个的分支下来，有点像是树枝状。因此，我们也称这种目录配置方式为：『目录树(directory tree)』这个目录树有什么特性呢？他主要的特性有：

目录树的启始点为根目录 (/ , root)；

每一个目录不止能使用本地端的 partition 的文件系统，也可以使用网络上的 filesystem。举例来说，可以利用 Network File System (NFS) 服务器挂载某特定目录等。

每一个文件在此目录树中的文件名(包含完整路径)都是独一无二的。

比较特殊的应该是/selinux 这个目录了，这个目录的内容数据也是在内存中的信息，同样的不会占用任何的硬盘容量。这个/selinux是 Secure Enhance Linux(SELinux)的执行目录，而 SELinux 是 Linux 核心的重要外挂功能之一，他可以用来作为细部权限的控管，主要针对程序(尤其是网络程序)的访问权限来限制

### 绝对路径与相对路径

除了需要特别注意的 FHS 目录配置外，在文件名部分我们也要特别注意！因为根据档名写法的不同，也可将所谓的路径(path)定义为绝对路径(absolute)与相对路径(relative)。这两种文件名/路径的写法依据是这样的：

绝对路径：由根目录(/)开始写起的文件名或目录名称，

例如 /home/dmtsai/.bashrc；

相对路径：相对于目前路径的文件名写法。

例如 ./home/dmtsai 或 http://www.cnblogs.com/home/dmtsai/ 等等。

反正开头不是 / 就属于相对路径的写法

而必须要了解，相对路径是以『你当前所在路径的相对位置』来表示的。举例来说，目前在 /home 这个目录下，如果想要进入 /var/log 这个目录时，可以怎么写呢？

```
cd /var/log (absolute)
```

```
cd ../var/log (relative)
```

因为你在 /home 底下，所以要回到上一层 (../) 之后，才能继续往 /var 来移动的！特别注意这两个特殊的目录：

· : 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。

这个 · 与 .. 目录概念是很重要的，你常常会看到 cd .. 或 ./command 之类的指令下达方式，就是代表上一层与目前所在目录的工作状态！

视野为王，技术为军

---

QQ:575119655 我爱七王爷