

理解 Linux 配置文件

介绍

每个 Linux 程序都是一个可执行文件，它含有操作码列表，CPU 将执行这些操作码来完成特定的操作。例如，ls 命令是由 /bin/ls 文件提供的，该文件含有机器指令的列表，在屏幕上显示当前目录中文件的列表时需要使用这些机器指令。几乎每个程序的行为都可以通过修改其配置文件来按照您的偏好或需要去定制。

Linux 中有没有一个标准的配置文件格式？

一句话，没有。不熟悉 Linux 的用户（一定）会感到沮丧，因为每个配置文件看起来都象是一个要迎接的新挑战。在 Linux 中，每个程序员都可以自由选择他或她喜欢的配置文件格式。可以选择的格式很多，从 /etc/shells 文件（它包含被一个换行符分开的 shell 的列表），到 Apache 的复杂的 /etc/httpd.conf 文件。

什么是系统配置文件？

内核本身也可以看成是一个“程序”。为什么内核需要配置文件？内核需要了解系统中用户和组的列表，进而管理文件权限（即根据权限判定特定用户（UNIX-USERS）是否可以打开某个文件）。注意，这些文件不是明确地由程序读取的，而是由系统库所提供的一个函数读取，并被内核使用。例如，程序需要某个用户的（加密过的）密码时不应该打开 /etc/passwd 文件。相反，程序应该调用系统库的 getpw() 函数。这种函数也被称为系统调用。打开 /etc/passwd 文件和之后查找那个被请求的用户的密码都是由内核（通过系统库）决定的。除非另行指定，Red Hat Linux 系统中大多数配置文件都在 /etc 目录中。配置文件可以大致分为下面几类：

访问文件

<b>/etc/host.conf</b>	告诉网络域名服务器如何查找主机名。（通常是 /etc/hosts，然后就是名称服务器；可通过 netconf 对其进行更改）
<b>/etc/hosts</b>	包含（本地网络中）已知主机的一个列表。如果系统的 IP 不是动态生成，就可以使用它。对于简单的主机名解析(点分表示法)，在请求 DNS 或 NIS 网络名称服务器之前，/etc/hosts.conf 通常会告诉解析程序先查看这里。
<b>/etc/hosts.allow</b>	请参阅 hosts_access 的联机帮助页。至少由 tcpd 读取。
<b>/etc/hosts.deny</b>	请参阅 hosts_access 的联机帮助页。至少由 tcpd 读取。

引导和登录 / 注销

<b><i>/etc/issue &amp; /etc/issue.net</i></b>	这些文件由 mingetty（和类似的程序）读取，用来向从终端（issue）或通过 telnet 会话（issue.net）连接的用户显示一个“welcome”字符串。它们包括几行声明 Red Hat 版本号、名称和内核 ID 的信息。它们由 rc.local 使用。
<b><i>/etc/redhat-release</i></b>	包括一行声明 Red Hat 版本号和名称的信息。由 rc.local 使用。
<b><i>/etc/rc.d/rc</i></b>	通常在所有运行级别运行，级别作为参数传送。例如，要以图形（Graphics）模式（X-Server）引导机器，请在命令行运行下面的命令：init 5。运行级别 5 表示以图形模式引导系统。
<b><i>/etc/rc.d/rc.local</i></b>	非正式的。可以从 rc、rc.sysinit 或 /etc/inittab 调用。
<b><i>/etc/rc.d/rc.sysinit</i></b>	通常是所有运行级别的第一个脚本。
<b><i>/etc/rc.d/rc/rcX.d</i></b>	从 rc 运行的脚本（X 表示 1 到 5 之间的任意数字）。这些目录是特定“运行级别”的目录。当系统启动时，它会识别要启动的运行级别，然后调用该运行级别的特定目录中存在的所有启动脚本。例如，系统启动时通常会在引导消息之后显示“entering run-level 3”的消息；这意味着 /etc/rc.d/rc3.d/ 目录中的所有初始化脚本都将被调用。

## 文件系统

内核提供了一个接口，用来显示一些它的数据结构，这些数据结构对于决定诸如使用的中断、初始化的设备和内存统计信息之类的系统参数可能很有用。这个接口是作为一个独立但虚拟的文件系统提供的，称为 /proc 文件系统。很多系统实用程序都使用这个文件系统中存在的值来显示系统统计信息。例如，/proc/modules 文件列举系统中当前加载的模块。lsmod 命令读取此信息，然后将其以人们可以看懂的格式显示出来。下面表格中指定的 mtab 文件以同样的方式读取包含当前安装的文件系统的 /proc/mount 文件。

<b><i>/etc/mtab</i></b>	这将随着 /proc/mount 文件的改变而不断改变。换句话说，文件系统被安装和卸载时，改变会立即反映到此文件中。
<b><i>/etc/fstab</i></b>	列举计算机当前“可以安装”的文件系统。这非常重要，因为计算机引导时将运行 mount -a 命令，该命令负责安装 fstab 的倒数第二列中带有“1”标记的每一个文件系统。
<b><i>/etc/mtools.conf</i></b>	DOS 类型的文件系统上所有操作（创建目录、复制、格式化等等）的配置。

## 系统管理

<b><i>/etc/group</i></b>	包含有效的组名称和指定组中包括的用户。单一用户如果执行多个任务，可以存在于多个组中。例如，如果一个“用户”是“project 1”工程组的成员，同时也是管理员，那么在 group 文件中他的条目看起来就会是这样的：user: * : group-id : project1
<b><i>/etc/nologin</i></b>	如果有 /etc/nologin 文件存在，login(1) 将只允许 root 用户进行访问。它将对其它用户显示此文件的内容并拒绝其登录。
<b><i>etc/passwd</i></b>	请参阅“man passwd”。它包含一些用户帐号信息，包括密码（如果未被 shadow 程序加密过）。
<b><i>/etc/rpmrc</i></b>	rpm 命令配置。所有的 rpm 命令行选项都可以在这个文件中一起设置，这样，当任何 rpm 命令在该系统中运行时，所有的选项都会全局适用。
<b><i>/etc/securetty</i></b>	包含设备名称，由 tty 行组成（每行一个名称，不包括前面的/dev/），root 用户在这里被允许登录。
<b><i>/etc/usertty</i> <i>/etc/shadow</i></b>	包含加密后的用户帐号密码信息，还可以包括密码时效信息。包括的字段有： <ul style="list-style-type: none"> <li>• 登录名</li> <li>• 加密后的密码</li> <li>• 从 1970 年 1 月 1 日到密码最后一次被更改的天数</li> <li>• 距密码可以更改之前的天数</li> <li>• 距密码必须更改之前的天数</li> <li>• 密码到期前用户被警告的天数</li> <li>• 密码到期后帐户被禁用的天数</li> <li>• 从 1970 年 1 月 1 日到帐号被禁用的天数</li> </ul>
<b><i>/etc/shells</i></b>	包含系统可用的可能的“shell”的列表。
<b><i>/etc/motd</i></b>	每日消息；在管理员希望向 Linux 服务器的所有用户传达某个消息时使用。

## 联网

<b><i>/etc/gated.conf</i></b>	gated 的配置。只能被 gated 守护进程所使用。
<b><i>/etc/gated.version</i></b>	包含 gated 守护进程的版本号。
<b><i>/etc/gateway</i></b>	由 routed 守护进程可选地使用。
<b><i>/etc/networks</i></b>	列举从机器所连接的网络可以访问的网络名和网络地址。通过路由命令使用。允许使用网络名称。
<b><i>/etc/protocols</i></b>	列举当前可用的协议。请参阅 NAG（网络管理员指南，Network Administrators Guide）和联机帮助页。

	C 接口是 getprotoent。绝不能更改。
<b>/etc/resolv.conf</b>	在程序请求“解析”一个 IP 地址时告诉内核应该查询哪个名称服务器。
<b>/etc/rpc</b>	包含 RPC 指令/规则, 这些指令/规则可以在 NFS 调用、远程文件系统安装等中使用。
<b>/etc/exports</b>	要导出的文件系统(NFS)和对它的权限。
<b>/etc/services</b>	将网络服务名转换为端口号/协议。由 inetd、telnet、tcpdump 和一些其它程序读取。有一些 C 访问例程。
<b>/etc/inetd.conf</b>	inetd 的配置文件。请参阅 inetd 联机帮助页。包含每个网络服务的条目, inetd 必须为这些网络服务控制守护进程或其它服务。注意, 服务将会运行, 但在 /etc/services 中将它们注释掉了, 这样即使这些服务在运行也将不可用。格式为: <service-name> <sock-type> <proto> <flags> <user> <server-path> <args>
<b>/etc/sendmail.cf</b>	邮件程序 sendmail 的配置文件。比较隐晦, 很难理解。
<b>/etc/sysconfig/network</b>	指出 NETWORKING=yes 或 no。至少由 rc.sysinit 读取。
<b>/etc/sysconfig/network-scripts/if*</b>	Red Hat 网络配置脚本。

## 系统命令

系统命令要独占地控制系统, 并让一切正常工作。所有如 login (完成控制台用户身份验证阶段) 或 bash (提供用户和计算机之间交互) 之类的程序都是系统命令。因此, 和它们有关的文件也特别重要。这一类别中有下列令用户和管理员感兴趣的文件。

<b>/etc/lilo.conf</b>	包含系统的缺省引导命令行参数, 还有启动时使用的不同映象。您在 LILO 引导提示的时候按 Tab 键就可以看到这个列表。
<b>/etc/logrotate.conf</b>	维护 /var/log 目录中的日志文件。
<b>/etc/identd.conf</b>	identd 是一个服务器, 它按照 RFC 1413 文档中指定的方式实现 TCP/IP 提议的标准 IDENT 用户身份识别协议。identd 的操作原理是查找特定 TCP/IP 连接并返回拥有此连接的进程的用户名。作为选择, 它也可以返回其它信息, 而不是用户名。请参阅 identd 联机帮助页。
<b>/etc/ld.so.conf</b>	“动态链接程序” (Dynamic Linker) 的配置。

<b><i>/etc/inittab</i></b>	按年代来讲,这是 UNIX 中第一个配置文件。在一台 UNIX 机器打开之后启动的第一个程序是 init,它知道该启动什么,这是由于 inittab 的存在。在运行级别改变时,init 读取 inittab,然后控制主进程的启动。
<b><i>/etc/termcap</i></b>	一个数据库,包含所有可能的终端类型以及这些终端的性能。

## 守护进程

守护进程是一种运行在非交互模式下的程序。一般来说,守护进程任务是和联网区域有关的:它们等待连接,以便通过连接提供服务。Linux 可以使用从 Web 服务器到 ftp 服务器的很多守护进程。

<b><i>/etc/syslogd.conf</i></b>	syslogd 守护进程的配置文件。syslogd 是一种守护进程,它负责记录(写到磁盘)从其它程序发送到系统的消息。这个服务尤其常被某些守护进程所使用,这些守护进程不会有另外的方法来发出可能有问题存在的信号或向用户发送消息。
<b><i>/etc/httpd.conf</i></b>	Web 服务器 Apache 的配置文件。这个文件一般不在 /etc 中。它可能在 /usr/local/httpd/conf/ 或 /etc/httpd/conf/ 中,但是要确定它的位置,您还需要检查特定的 Apache 安装信息。
<b><i>/etc/conf.modules</i> or <i>/etc/modules.conf</i></b>	kernelld 的配置文件。有意思的是,kernelld 并不是“作为守护进程的”内核。它其实是一种在需要时负责“快速”加载附加内核模块的守护进程。

## 用户程序

在 Linux (和一般的 UNIX) 中,有无数的“用户”程序。最常见的一种用户程序配置文件是 /etc/lynx.cfg。这是著名的文本浏览器 lynx 的配置文件。通过这个文件,您可以定义代理服务器、要使用的字符集等等。下面的代码样本展示了 lynx.cfg 文件的一部分,修改这部分代码可以改变 Linux 系统的代理服务器设置。缺省情况下,这些设置适用于在各自的 shell 中运行 lynx 的所有用户,除非某个用户通过指定 --cfg = "mylynx.cfg" 重设了缺省的配置文件。

### **/etc/lynx.cfg 中的代理服务器设置**

```
. h1 proxy
. h2 HTTP_PROXY
. h2 HTTPS_PROXY
. h2 FTP_PROXY
. h2 GOPHER_PROXY
. h2 NEWS_PROXY
```

```
.h2 NNTP_PROXY
# Lynx version 2.2 and beyond supports the use of proxy servers that
# can act as
# firewall gateways and caching servers. They are preferable to the
# older
# gateway servers. Each protocol used by Lynx can be mapped
# separately using
# PROTOCOL_proxy environment variables (see Lynx Users Guide). If you
# have
# not set them externally, you can set them at run time via this
# configuration file.
# They will not override external settings. The no_proxy variable can
# be used
# to inhibit proxying to selected regions of the Web (see below).
# Note that on
# VMS these proxy variables are set as process logicals rather than
# symbols, to
# preserve lowercasing, and will outlive the Lynx image.
#
.ex 15
http_proxy:http://proxy3.in.ibm.com:80/
ftp_proxy:http://proxy3.in.ibm.com:80/
#http_proxy:http://penguin.in.ibm.com:8080
#ftp_proxy:http://penguin.in.ibm.com:8080/
.h2 NO_PROXY
# The no_proxy variable can be a comma-separated list of strings
# defining
# no-proxy zones in the DNS domain name space. If a tail substring
# of the
# domain-path for a host matches one of these strings, transactions
# with that
# node will not be proxied.
.ex
no_proxy:demiurge.in.ibm.com, demiurge
```

### 更改配置文件

在更改配置文件时，如果程序不是由系统管理员或内核控制的，就要确保重新启动过使用该配置的程序。普通用户通常没有启动或停止系统程序和/或守护进程的权限。

### 内核



更改内核中的配置文件会立即影响到系统。例如，更改 passwd 文件以增加用户将立即使该用户变为可用。而且任何 Linux 系统的 /proc/sys 目录中都有一些内核可调参数。只有超级用户可以得到对所有这些文件的写访问权力；其它用户只有只读访问权力。此目录中文件的分类的方式和 Linux 内核源代码的分类方式一样。此目录中的每个文件都代表一个内核数据结构，这些数据结构可以被动态地修改，从而改变系统性能。

**注意：**在更改其中任何文件的任何值之前，您应该确保自己全面了解该文件，以避免对系统造成不可修复的损害。

**/proc/sys/kernel/ 目录中的文件**

文件名	描述
threads-max	内核可运行的最大任务数。
ctrl-alt-del	如果值为 1，那么顺序按下这几个键将“彻底地”重新引导系统。
sysrq	如果值为 1，Alt-SysRq 则为激活状态。
osrelease	显示操作系统的发行版版本号
ostype	显示操作系统的类型。
hostname	系统的主机名。
domainname	网络域，系统是该网络域的一部分。
modprobe	指定 modprobe 是否应该在启动时自动运行并加载必需的模块。

**守护进程和系统程序**

守护进程是永远运行在后台的程序，它默默地执行自己的任务。常见的守护进程有 in.ftpd (ftp 服务器守护进程)、in.telnetd (telnet 服务器守护进程) 和 syslogd (系统日志记录守护进程)。有些守护进程在运行时会严密监视配置文件，在配置文件改变时就会自动重新加载它。但是大多数守护进程并不会自动重新加载配置文件。我们需要以某种方式“告诉”这些守护进程配置文件已经被发生了改变并应该重新加载。可以通过使用服务命令重新启动服务来达到这个目的（在 Red Hat Linux 系统上）。

例如，如果我们更改了网络配置，就需要发出：

service network restart 。

注意：这些服务最常见的是 /etc/rc.d/init.d/\* 目录中存在的脚本，在系统被引导时由 init 启动。所以，您也可以执行如下操作来重新启动服务：

/etc/rc.d/init.d/<script-for-the-service> start | stop | status

start、stop 和 status 是这些脚本接受的输入值，用来执行操作。

**用户程序**

用户或系统程序在每次启动时都会读取其配置文件。尽管如此，请记住，有些系统程序在计算机打开时情况不一样，它们的行为依赖于在 /etc/ 中的配置文件中读到的内容。所以，用户程序第一次启动时将从 /etc/ 目录中存在的文件读取缺省配置。然后，用户可以通过使用 rc 和 . (点) 文件来定制程序，正如下面一节所示。

## 用户配置文件：.（点）文件和 rc 文件

我们已经看到怎样容易地配置程序。但是如果有的人不喜欢在 /etc/ 中配置程序的方式该怎么办呢？“普通”用户不能简单地进入 /etc 然后更改配置文件；从文件系统的角度来看，配置文件的所有者是 root 用户！这就是大多数用户程序都定义两个配置文件的原因：第一个是“系统”级别的，位于 /etc/；另一个属于用户“专用”，可以在他或她的主目录中找到。

例如，我在我的系统中安装了非常有用的 wget 实用程序。/etc/ 中有一个 /etc/wgetrc 文件。在我的主目录中，有一个名为 .wgetrc 的文件，它描述了我定制的配置（只有在我，也就是用户运行 wget 命令时，才会加载这个配置文件）。其它用户在他们自己的主目录（/home/other）中也可以有 .wgetrc 文件；当然，只有这些用户运行 wget 命令时，才会读取这个文件。换句话说，/etc/wgetrc 文件为 wget 提供了“缺省”值，而 /home/xxx/.wgetrc 文件列举了某个用户的“定制项”。重要的是这只是“一般规则”，并非所有情况都如此。例如，一个象 pine 一样的程序，在 /etc/ 中并没有任何文件，它只在用户主目录中有一个定制配置文件，名为 .pinerc。其它程序可能只有 /etc/ 中的缺省配置文件，而且可能不允许用户“定制”这些配置文件（/etc 目录中只有少数 config. 文件是这种情况）。

### 通常使用的 rc 和 .（点）文件

文件名	描述
~/.bash-login	请参考“man bash”。如果 ~/.bash-profile 不存在，bash 则将 ~/.bash-login 作为 ~/.bash-profile 处理。
~/.bash-logout	请参考“man bash”。在退出时由 bash 登录 shell 引用。
~/.bash-profile	由 bash 登录 shell 引用 /etc/profile 之后引用。
~/.bash-history	先前执行的命令的列表。
~/.bashrc	请参考“man bash”。由 bash 非登录交互式 shell 引用（没有其它文件）。除非设置了 BASH_ENV 或 ENV，非交互式 shell 不引用任何文件。
~/.emacs	启动时由 emacs 读取。
~/.forward	如果这里包含一个电子邮件地址，那么所有发往 ~ 的所有者的邮件都会被转发到这个电子邮件地址。



<code>~/.fvwmrc</code> <code>~/.fvwm2rc</code>	fvwm 和 fvwm2 (基本的 X Window 管理器) 的配置文件。
<code>~/.hushlogin</code>	请参考“man login”。引起“无提示”登录 (没有邮件通知、上次登录信息或者 MOD 信息)。
<code>~/.mail.rc</code>	邮件程序的用户初始化文件。
<code>~/.ncftp/</code>	ncftp 程序的目录; 包含书签、日志、宏、首选项和跟踪信息。请参阅 man ncftp。ncftp 的目的是为因特网标准文件传输协议 (Internet standard File Transfer Protocol) 提供一个强大而灵活的接口。它旨在替换系统所使用的标准的 ftp 程序。
<code>~/.profile</code>	请参考“man bash”。如果 <code>~/.bash-profile</code> 和 <code>~/.bash-login</code> 文件不存在, bash 则将 <code>~/.profile</code> 作为 <code>~/.bash-profile</code> 处理, 并被其它继承 Bourn 的 shell 使用。
<code>~/.pinerc</code>	Pine 配置
<code>~/.muttrc</code>	Mutt 配置
<code>~/.exrc</code>	这个文件可以控制 vi 的配置。 示例: set ai sm ruler 在此文件中写入上面一行会让 vi 设置自动缩进、匹配括号、显示行号和行-列这几个选项。
<code>~/.vimrc</code>	缺省的“Vim”配置文件。和 <code>.exrc</code> 一样。
<code>~/.gtkrc</code>	GNOME 工具包 (GNOME Toolkit)。
<code>~/.kderc</code>	KDE 配置。
<code>~/.netrc</code>	ftp 缺省登录名和密码。
<code>~/.rhosts</code>	由 r- 工具 (如 rsh、rlogin

	<p>等等)使用。因为冒充主机很容易,所以安全性非常低。</p> <ol style="list-style-type: none"> <li>1. 必须由用户(~/ 的所有者)或超级用户拥有。</li> <li>2. 列出一些主机,用户可以从这些主机访问该帐号。</li> <li>3. 如果是符号链接则被忽略。</li> </ol>
~/ .rpmrc	请参阅“man rpm”。如果/etc/rpmrc 不存在则由rpm 读取。
~/ .signature	消息文本,将自动附加在从此帐号发出的邮件末尾。
~/ .twmrc	twm( <b>T</b> he <b>W</b> indow <b>M</b> anager)的配置文件。
~/ .xinitrc	<p>启动时由 X 读取(而不是由xinit 脚本读取)。通常会启动一些程序。</p> <p>示例: exec /usr/sbin/startkde</p> <p>如果该文件中存在上面这行内容,那么在从这个帐号发出 startx 命令时,这一行就会启动“KDE 视窗管理器”(KDE Window Manager)。</p>
~/ .xmodmaprc	此文件被传送到 xmodmap 程序,而且可以被命名为任何文件(例如 ~/ .Xmodmap 和 ~/ .keymap.km)。
~/ .xserverrc	如果 xinit 可以找到要执行的 X, xinit 就会将该文件作为 X 服务器运行。
~/ News/Sent-Message-IDs	gnus 的缺省邮件历史文件。
~/ .Xauthority	由 xdm 程序读和写,以处理权限。请参阅 X、xdm 和 xauth 联机帮助页。
~/ .Xdefaults, ~/ .Xdefaults-hostname	在主机 hostname 的启动过程中由 X 应用程序读取。如果找不到 -hostname 文件,

	则查找 <code>.Xdefaults</code> 文件。
<code>~/.Xmodmap</code>	指向 <code>.xmodmaprc</code> ; Red Hat 有使用这个名称的 <code>.xinitrc</code> 文件。
<code>~/.Xresources</code>	通常是传送到 <code>xrdb</code> 以加载 X 资源数据库的文件的名称, 旨在避免应用程序需要读取一个很长的 <code>.Xdefaults</code> 文件这样的情况。(有些情况曾经使用了 <code>~/.Xres</code> 。)
<code>~/mbox</code>	用户的旧邮件。