第7章　登录和注销

本章将说明用户在登录和注销时发生的行为。另外，还要详细介绍后台进程、日志文件、配置文件等的交互过程。

7.1　通过终端登录

图7-1展示了如何通过终端进行登录。首先， init要确定有一个供该终端连接（或控制台）使用的getty程序。getty在终端上监听并等待用户通知它“他／她准备登录了”（这通常意味着用户必须输入点东西）。它注意到用户后， getty就输出一条欢迎消息（保存在/etc/issue）内，提示用户输入用户名，最后才运行 login（登录）程序。login得到作为参数的用户名后，提示用户输入密码。如果两者相符， login便开始启动为该用户配置的外壳脚本；如果两者不符， login只好退出并中断登录进程（也许要求用户再次提供用户名和密码之后）。init注意到登录进程中断后，就为终端启动一个新的getty。

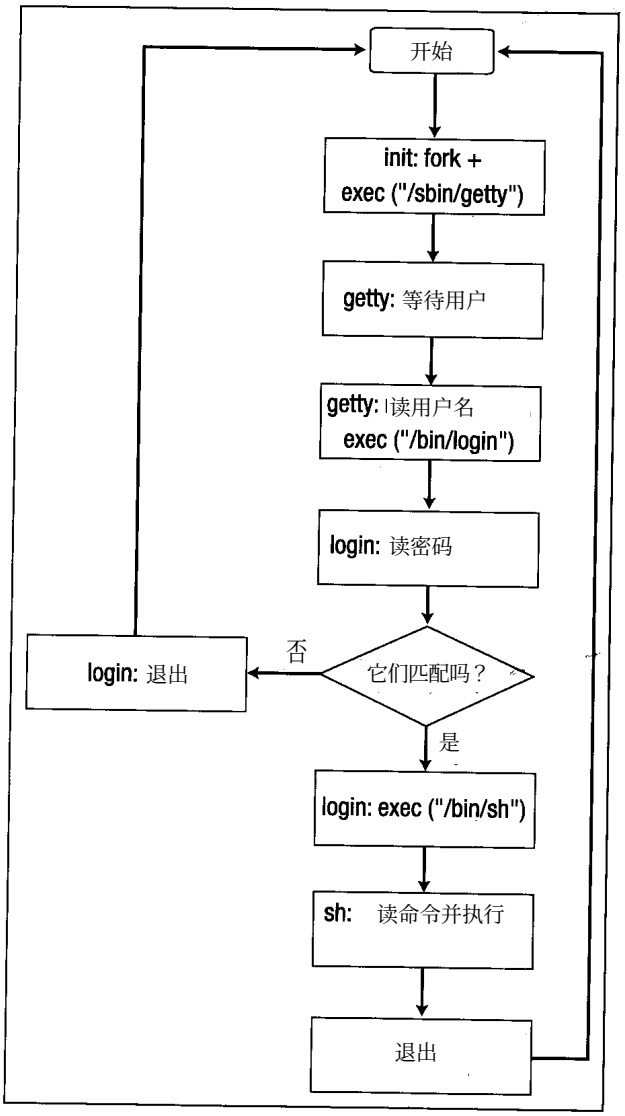

图7-1　通过终端登录：init、getty、login和shell之间的交互活动

注意，唯一的新进程是由 init创建的那个（利用 fork系统调用创建的）；getty和login只能替换进程中正在运行的程序（利用 exec系统调用）。

值得用户注意的是，串行线路需要一个单独的程序，因为在终端进入活动状态时，这个程序可能（过去一直如此）非常复杂。getty还可适应于连接的速率和其他参数设置的变化，这些对拨入连接来说，是特别重要的，因为对拨入连接来说，参数可能会不

时地发生变化（即每次呼叫时的参数都可能不一样）。

getty和init都有好几个版本，它们都有各自的优缺点。因此，你最好要知道自己系统内采用的是什么版本，与此同时，还需要对其他的版本有所了解（可利用 Linux Software Map来搜索它们）。如果没有拨入连接，大可不必太多关注 getty，但仍需了解init。

7.2 通过网络登录

同一个网络内的两台计算机通常通过一条单独的物理线缆连接在一起。当它们通过网络进行通信时，对参与通信的每台计算机上的程序来说，都会通过一条虚拟的连接（一条假想的线缆）链接在一起。只要虚拟连接任何一端上的程序被包含在内，都会有自己专用的“线缆”。但是，由于线缆不是真的，只是假想的，两台计算机上的操作系统都可能出现数条虚拟连接共享同一条物理线缆的情况。通过这种只采用一条物理线缆的方式，数个程序可以在不知道或不担心其他通信的情况下，进行通信。甚至于数台计算机采用同一条物理线缆也是可能的；虚拟连接存在于两台计算机之间，而其他计算机忽略那些它们自己没有参与的连接。

以上说明复杂而且过于抽象。但是，仍然非常易于理解，我们从中能看出网络登录和普通登录之所以不同的重要原因。虚拟连接是在有两个程序希望通信时建立的，这两个程序分别在不同计算机上。由于网络内的任何一台计算机都有可能想登录到另一台计算机，因此可能存在数目较大的虚拟通信。正由于此，要对每一次潜在的登录启动 getty几乎是不现实的。

所以，一个能处理所有网络登录的单独进程应运而生，它就是 inetd（对应于getty）。它注意到进入的网络登录（也就是说，它注意到自己与另一台计算机建立了新的虚拟连接），就启动一个新进程来处理单独的登录。原来的进程仍然存在，并继续监听新的登录。

为了能处理更多的事情，网络登录使用的通信协议不止一种。最重要的两个是 telnet和rlogin。除了登录以外，还可能实行其他许多虚拟连接（针对 FTP、Gopher、HTTP和其他一些网络服务），有一个单独的进程来监听特定类型的连接是非常有效的，所以就出现了一个专门的监听者，它能够识别连接类型并启动相应类型的程序来提供服务。这个单独的监听者就是/cmd{inetd}；有关详情，请参考本书第一部分。

7.3 登录的意义

login程序负责对用户进行身份验证（确定用户名和密码是否匹配），以及为用户设置初始化环境，这是通过为串行线路设置访问许可和启动 shell（外壳）来完成的。

部分初始化设置将输出文件 /etc/motd（message of the day的缩写，意为日期消息）的内容，并检查电子邮件。这些设置是可以禁用的，具体做法是在用户的根目录内，创建一个名为.hushlogin的文件。

如果/etc/nologin文件存在，登录就会被取消。这个文件一般是由 shutdown及其关系词创建的。login检查这个文件，如果这个文件存在的话，它将拒绝接受登录请求。如果这个文件不存在，login就会在退出之前，把自己的内容输出到终端。

login把所有失败的登录尝试记录在系统日志文件中（通过 syslog）。另外，它还记录了root的每次登录。在跟踪入侵者时，这些记录都是非常有用的。

当前已经登录的用户都列在 /var/run/utmp内。只有在系统下一次重启或关闭之前，这个文件才是有效的；系统启动时，就会被清除。该文件列出了每个用户以及他们使用的终端（或

网络连接），另外还有其他有用的信息。 who、w和其他类似的命令将在 utmp内查看登录的用户。

所有成功的登录都记录在 /var/log/wtmp文件内。该文件将无限制地增长，所以必须定期对它进行清除，最好每周执行一次 cron任务，清除它。最后一个命令用于浏览 wtmp文件。

utmp和wtmp均是二进制格式（参见 utmp手册页）；但遗憾的是，没有特定的程序，是不方便对它们进行检查的。

7.4 访问控制

过去，用户数据库包含在 /etc/passwd文件内。有的系统采用影子密码，并且已经把密码挪到/etc/shadow文件内。对具有多台计算机（它们共享账号）的站点来说，利用 NIS或别的方法来保存用户数据库；它们还可能自动把数据库从一个集中地点复制到所有的计算机。

用户数据库内不仅包含密码，还有一些和用户有关的额外信息，比如用户的真名、根目录和登录外壳等。这些额外的信息需要公开，以便任何人都可读取它们。因此，密码要实行加密保存。这样做的确有不足的地方，因为任何访问加密密码的人可采用各种解密方法来猜测真正的密码，根本无须尝试实际登录计算机。影子密码试图通过将密码移入另一个文件（只有root才能读取它，但密码仍采用加密方式保存）内的方式，来避免此类情况的发生。但是，在一个不支持影子密码的系统上安装它不是件简单的事。

不管有无密码，重要的是要确定系统内的所有密码都是可靠的，也就是说，不是轻易就能猜出来的。解密程序能够用来对密码进行解密；它能找出的任何密码都被认为是不可靠的。解密程序可以由入侵者实施，但也可由系统管理员来实施，以避免采用不可靠密码。可靠的密码也可以由 passwd程序来执行；实际上，这在 CPU周期中更为有效，因为解密密码需要进行大量的计算。

用户组数据库保存在 /etc/group内；对安装了影子密码的系统来说，用户组数据库也可能是/etc/shadow.group。

root通常不能通过大多数终端或网络登录，它只能通过 /etc/securetty文件内列出的终端进行登录。这样一来，就有必要与这些终端之一建立物理连接，以便进行访问。但是，它也可以作为任何一名其他的用户，通过任何一个终端进行登录，然后再用 su命令，恢复 root的身份。

7.5 外壳的启动

一个交互性的登录外壳启动时，它将自动执行一个或多个预先定义好的文件。不同的外壳，执行的文件也是不同的；有关详情，参见各个外壳的文档。

许多外壳首先运行某个全局文件，比如 Bourne外壳（/bin/sh）及其衍生程序执行的是/etc/profile；另外，它们还将执行用户根目录下的 .profile文件。/etc/profile允许系统管理员设置好一个普通的用户环境，特别是设置路径，使其除了包含普通目录外，还要包括本地命令目录。另一方面， .profile还允许用户在必要时，通过改写默认设置的方式，自定义他／她的用户环境。