第三部分　登 录 环 境

第13章　登 录 环 境

登录系统时，在进入命令提示符前，系统要做两个工作。键入用户名和密码后，系统检查是否为有效用户，为此需查询 /etc/passwd文件。如果登录名正确并且密码有效，开始下一步过程，即登录环境。

本章内容有：

- 登录过程。
- 文件/etc/passwd。
- $HOME.profile。
- 定制$HOME.profile。

在进行下一步处理之前，先看看文件 /etc/passwd。这是一个文本文件，可以任意修改其中的文本域，但要小心。此文本有 7个域，并用冒号作分隔符，以下是其部分文件内容列表。在顶端加有列号，这样各域标识得更加清晰。

```
[ 1 ][          2          ][ 3 ][ 4 ][      5      ][          6          ][   7   ]
kvp:JFqMmk9.uRioA:405:413:K.V.Pally:/home/sysdev/kvp:/bin/sh
dhw:hi/G4U1CUd9aI,B/0J:407:401:D.Whitely:/home/dept47/dhw:/bin/sh
aec:ILgHtxJ9kXtSc,B/GI:408:401:A.E.Cloody:/b_user/dept47/aec:/bin/sh
gdw:iLFu9BB8RNjpc,B/MK:409:401:G.D.Wilcom:/b_user/dept47/gdw:/bin/sh
```

现在来看看各域，第 1域是登录名，第 2域是加密的密码，第 5域是用户全名。第 6域是用户根目录，第7域是用户使用的shell。这里/bin/sh意即缺省为常规Bourne Shell。

Passwd文件可能还有其他格式。其中的一个版本即为实际 passwd域保存在另一个文件中。以上即为最普通格式。

登录成功后，系统执行两个环境设置文件，第一个是 /etc/profile，第二个是 .profile，位于用户根目录下。

系统还会处理其他的初始化文件。这里只涉及 profile文件。

13.1　/etc/profile

用户登录时，自动读取/etc目录下profile文件，此文件包含：

- 全局或局部环境变量。
- PATH信息。
- 终端设置。
- 安全命令。
- 日期信息或放弃操作信息。

下面就来详细解释上述各项内容。设置全局环境变量便于用户及其进程和应用访问它。

PATH定位包含可执行文件，库文件及一般文本文件的目录位置，便于用户快速访问。终端设置使系统获知用户终端的一般特性。安全命令包括文件创建模式或敏感区域的双登录提示。日期信息是一个文本文件，保存用户登录时即将发生事件的记录或放弃登录的信息文件。

以下是文件/etc/profile，列表后将予以讨论。

```
$ pg /etc/profile
#!/bin/sh
#
trap "" 2 3
#  Set LOGNAME
export LOGNAME

# set additional MAN paths.
MANPATH=/usr/opt/sybase/man
export MANPATH

#  Set TZ.
if [ -f /etc/TIMEZONE ]
then
.  /etc/TIMEZONE
fi

#  Set TERM.
if [ -z "$TERM" ]
then
  TERM=vt220    # for standard async terminal/console
  export TERM
fi

  #  Allow the user to break the Message-Of-The-Day only.
  trap "trap "2"  2
  if [ -f /usr/bin/cat ] ; then
    cat -s /etc/motd
  fi
  trap "" 2
  if [ -f /usr/bin/mail ] ; then
    if mail -e ; then
      echo "Hey guess what? you have mail"
    fi
  fi
  ;;
esac

#  set the umask for more secure operation
umask 022
fi
#  set environments
SYSHOME=/appdvb/menus
ASLBIN=/asl_b/bin
UDTHOME=/dbms_b/ud
UDTBIN=/dbms_b/ud/bin
PAGER=pg
NOCHKLPREQ=1
PATH=$PATH:$UDBIN:$ASLBIN
```

```
export PATH UDTHOME UDTBIN PAGER NOCHKLPREQ SYSHOME

trap  2 3
#  Set variable SAVEDSTTY so that it can be used to recover the
#  stty settings on coming out of the audit system.
SAVEDSTTY=‘stty -g‘
export SAVEDSTTY

# log all connections to syslog
logger -p local7.info -t login  $LOGNAME `tty`
trap 'logger -p local7.info -t logout $LOGNAME `tty`' 0

# suppress creation of core dumps
ulimit -c 0
#
# check if users are logged in more than twice apart from...
case $LOGNAME in
idink | psalon | dave)
    ;;
  *)
    PID=${$};export PID
    Connected=`who | awk '{print $1}' | fgrep -xc ${LOGNAME}`
    if [ "$Connected" -gt  2 ]
    then
      echo
      echo 'You are logged in more than twice.'
      echo
      who -u | grep $LOGNAME
      echo
      echo 'Enter <CR> to exit \c'
      read FRED
      kill -15 $PID
    fi
    ;;
esac
# set the prompt to hold the user id
PS1="$LOGNAME >"
```

其中一些命令可能不好理解，不必担心，本书以后将陆续予以介绍。如果愿意，可以参照这个列表建立自己的profile文件。

第一行捕获两个信号，即使用QUIT退出用户或<Ctrl-c>键停止文件执行。

接下来导出LOGNAME；然后指定系统额外增加的man页查询的位置。MANPATH将此位置加入存在的man页查询列表中。

检查时区文件，如果存在，指定它作为时区源，设置终端类型为 vt220。

重新设置捕获信号，以便于用户读取日期文件信息，但此后必须再重新设置它。

建立邮件信息（当有新邮件到达时显示此信息）。

设置umask值，使文件创建时带有一定的缺省权限位集。

初始化环境变量，设置路径并导出，以便于用户使用。

重新设置捕获信号<Ctrl-C>和QUIT。

保存缺省的stty设置，便于用户退出查询系统时能够重新初始化终端设置。

将所有连接注册到文件/var/adm/messages，即缺省系统注册文件中。

使用ulimit命令限制内存溢出或十六进制溢出数目。

下面的一小段脚本限制用户最多同时登录两次，但不包括三个人（ idnk，psalom，dave），如果有人试图登录超过两次，则令其退出登录进程。

最后设置命令提示符到登录名。

此环境设置为全局使用，下面在用户自己的 profile文件中定制环境。

13.2 用户的$HOME.profile

/etc/profile文件执行时，用户将被放入到自己的 $HOME目录中，回过头来观察 passwd文件，用户的$HOME目录在倒数第2列。

可以将之看作用户根目录，因为正是在这里存储了所有的私有信息。

如果.profile已经存在，系统将参照此文件，意即对此过程并不创建另一个 shell，因而在/etc/profile下设置的环境不做改动，除非在 .profile中强制改动它。如果创建另一个进程，用户本地的shell变量将被覆盖。

回到 .profile，一般来说创建帐户时，一个 profile文件的基本框架即随之创建。不要忘了在 .profile文件中可以通过设置相关条目以不同的值或使用 uset命令来覆盖/etc/profile文件中的设置。如果愿意，可以定制用户自己的 .profile文件。先来看看标准的.profile文件。

```
$ pg .profile
#.profile
set -a
MAIL=/usr/mail/${LOGNAME:?}
PATH=$PATH
export PATH
#
```

现在改动此文件。

现在加入两个环境变量，如 EDITOR，以使cron或其他应用获知正在使用的编辑器；将TERM变量设置为vt100，而不是vt220。

也可以创建bin目录，将之加入路径（ path），目录结构中加一个 bin目录是一个好习惯。在这里可以保存所有脚本，将之加入 PATH后，就不必写入脚本的文件路径名全称，只键入脚本名即可。

几乎没有人想在命令提示符中显示自己的登录名，而宁愿使用现在的目录路径或是正在使用的系统主机名做提示符。例如，下面显示了在命令提示符中如何设置主机名：

```
$ PS1="`hostname`>"
dns-server>
```

如果用户位于当前目录下：

```
$ PS1="\`pwd\`>"
/home/dave>
```

如果上面的命令返回pwd，可使用如下命令：

```
PS1='$PWD >';
```

我通常设置辅助命令提示符（一般用于命令提示符里的多行命令）为符号 ©，它的ASCII代码值八进制数为251，十进制为169。

```
$ PS2="`echo "\251"`:"
/home/dave> while read line
©:do
©:echo $line
©:done
```

如果是LINUX，那么……

在echo命令中使用八进制值，方法为：

```
$ PS2="`echo -e "\251"`:"
```

如果需要访问管理区/usr/admin，可将之加入环境变量，这样可以很容易地进入此目录。

```
ADMIN=/usr/adm
```

如果要知道用户本身登录后系统用户数，使用 who和wc命令。

```
$ echo "`who|wc -l` users are on today"

19 users are on today.
```

将上述设置加入.profile文件。如果要使.profile或/etc/profile文件改动生效，可以退出登录然后再登入，或者参照此文件设置。要参照此文件设置，格式为：

```
./pathname/filename
```

要参照.profile设置，键入：

```
$. .profile
```

如果未成功，试试：

```
$. ./profile
```

以下为改动过的.profile文件。

修改过.profile文件后使之生效的方法

```
$ pg .profile
#.profile
MAIL=/usr/mail/${LOGNAME:?}
PATH=$PATH:$HOME:bin
#
EDITOR=vi
TERM vt100
ADMIN=/usr/adm
PS1="`hostname`>"
PS2="`echo "\0251"`:"
export EDITOR TERM ADMIN PATH PS1
echo "`who|wc -l` users are on to-day"
```

13.3 stty用法

stty用于设置终端特性。要查询现在的 stty选项，使用 stty -a。

```
$ stty -a
speed 9600 baud; rows 24; columns 80; line = 0;
intr =^C; quit =^\; erase =^?; kill =^U; eof =^D; eol =<undef>;
eol2 =<undef>; start =^Q; stop =^S; susp =^Z; rprnt =^R; werase = ^W;
lnext =^V; flush =^O; min =1; time =0;
-parenb -parodd cs8 -hupcl -cstopb cread -clocal -crtscts .
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon
-ixoff -iuclc -ixany -imaxbel
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0
vt0 ff0 isig icanon iexten echo echoe echok -echonl -noflsh -xcase
```

```
-tostop -echoprt echoctl echoke
```

设置终端时遇到的一个最普遍问题是退格键不起作用。这不是不可挽救的。本机 stty命令中^?即为退格键，使用<Ctrl-H>可能会退格并删除前一个字符。在命令行中设置一个 stty选项，一般格式为：

```
stty name character
```

以下将退格设置为^H：

```
$ stty erase '\^H'
```

在.profile文件中使用上述命令可能会碰到一些问题，因为 stty期望输入一个实际'Control H'序列，在vi编辑器环境下使用下述方法解决它：

按住Ctrl键，同时按下V键，释放V键，再按下H键。

最常用的stty命令使用在下述设置上：

名 称	键	含 义
intr	^C	终止进程
echo		打开echo功能
-echo		关闭echo功能
eof	^D	文件尾；注销
kill	^Y	删除一行
start	^Q	滚动屏幕文本
stop	^S	停止滚动屏幕文本

stty的一个可用选项为：

```
stty -g
```

此选项允许以可读格式保存 stty现有设置，便于以后重置回 stty。正像前面在文件/etc/profile中看到的一样。将stty -g内容放入一个变量中，工作完成后，任何改动的设置将被写回stty。

在改变stty设置值并和终端打交道时，此方法很有用。这样可以很容易地存储其初始设置。下面的例子将stty的现有设置保存。使用 stty -g关掉echo，然后在脚本结尾处保存 stty初始设置。

```
$ pg password
#!/bin/sh
# password
# show use of the restoring stty environment
SAVEDSTTY=`stty -g`
stty -echo
echo "\nGive me that password :\c"
read PASSWD
echo "\nyour password is $PASSWD"
stty $SAVEDSTTY


$ sttypass
Give me that password :
your password is bong
```

如果是LINUX，那么……

要使LINUX知道正在使用字符串中转义字符，echo命令应加入-e，即echo -e。

（续）

```
SAVEDSTTY=`stty -g`
stty -echo
echo "\nGive me that password :\c"
read PASSWD
echo "\nyour password is $PASSWD"
stty $SAVEDSTTY
```

stty命令可以与终端、打印机、调制解调器打交道，功能十分丰富。使用 stty时要慎重，不要使用已经使用的键或无效值。

13.4 创建.logout文件

使用Bourne shell与其他shell不同，其缺点是不包含.logout文件。此文件保存有执行exit命令时，在进程终止前执行的命令。

但是通过使用trap命令（trap和信号将在本书后面讨论），Bourne shell也可以创建自己的.logout文件。方法如下：编辑.profile文件，在最后一行加入下列命令，然后保存并退出。

```
trap "$HOME /.logout"0
```

再键入一个.logout文件，敲入下列执行命令。如果愿意，可以在此脚本中加入任何命令。

```
$ pg .logout
rm -f $HOME/*.log
rm -f $HOME/*.tmp
echo "Bye...bye $LOGNAME"
```

用户退出时，调用.logout文件。过程如下：用户退出一个shell时，传送了一个信号0，意即从现在shell中退出，在控制返回shell继续退出命令前，.profile文件中trap行将捕获此信号并执行.logout。

13.5 小结

可以定制用户本身的.profile以满足需求，本章讲述了如何覆盖系统设置以满足用户需求。从显示友好信息到终端特性设置，定制用户环境可以有许多方式。