第6章　串行链路网际协议

串行链路网际协议SLIP和PPP为“穷人”提供了接入因特网的可能。只需要一个 Modem和一个配备FIFO缓存的串行卡，除此以外，不再需要别的硬件。它的使用非常简单，而且费用低廉，越来越多的私人企业，以普通用户能够承受的价格，开始提供拨号 IP服务。

本章和第9章，我们将为大家介绍 SLIP和PPP驱动程序。SLIP驱动程序的存在有相当长一段时间了，而且其运行相当可靠。 PPP驱动程序是迈克·克拉翰和艾尔·依伊尔近来开发出来的。我们将在下一章详细讨论。

6.1　常规需求

要想使用SLIP或PPP，必须像前面几章描述的那样，配置一些基本的网络特性。比如，起码应该安装LoopBack接口，以及提供名字解析。在连上因特网时，人们肯定想使用 DNS。最简单的方法是把某个域名服务器的地址放入自己的 resolv.conf文件中：只要SLIP链接一激活，就会对这个服务器进行查询。该名字服务器离你上网的地方越近，越好。

但是，上面的方法不是最佳解决之道，因为所有服务器名查找仍然通过你自己的SLIP/PPP链接来完成。如果担心因此而造成的带宽开销，最好安装一个 caching-only域名服务器。它不能真正地充当一个域，对发自你的主机的所有 DNS查询来说，它只是一个中间站。不过，它的好处在于建立了一个缓冲区，所以所有的查询都只能在这个串行链接上发送一次。caching-only服务器所用的named.boot文件像下面这样：

```
: named.boot file for caching-only server
directory                       /var/named
primary     0.0.127.in-addr.arpa    db.127.0.0      : loopback net
cache       .                   db.cache        : root servers
```

除了named.boot文件外，还必须用一个有效的根域名服务器清单来安装 db.cache文件。

6.2　SLIP的工作原理

拨号IP服务器通过特定的用户账号，不间断地提供 SLIP服务。用户登录后，执行的不是一个常见的外壳，而是一个程序或外壳脚本，之后，再启用串行线路 SLIP驱动程序，并配置相应的网络接口。

有些操作系统上， SLIP驱动程序是一个用户空间程序；在Linux操作系统上，这个驱动程序则集成在内核中，因此，其运行速度快得多。但是，它要求串行线路必须被显式转换为SLIP模式。这是通过一个特殊的 tty线路法，即SLIPDISC来完成的。在tty处于普通线路法（DISC0）时，只采用普通的读(2)和写(2)调用与用户进程交换数据， SLIP驱动程序不能对tty进行读写操作。SLIPDISC中，角色发生了变化：所有用户空间进程都被封锁，不得对 tty进行读写操作，而来自串行端口的所有数据都直接传送到 SLIP驱动程序。

SLIP驱动程序本身能够识别 SLIP协议的各种变形。除了常规的 SLIP外，它还能识别

CSLIP，显著提升交互式对话的流通量。CSLIP对输出IP包进行所谓的Van Jacobson报头压缩。至于Van Jacobson报头压缩的详情，可参见RFC-1441。另外，每个SLIP协议的变形都有6位版本。

要把串行线路转换为SLIP模式，最简单的方式是利用slattach工具。假设Modem已经存在于/dev/cua3目录中，而且已经成功登录到SLIP服务器。那么，执行下面的语句：

```
# slattach /dev/cua3 &
```

就会把cua3的线路切换为SLIPDISC，并把它附着在其中一个SLIP网络接口上。如果是初次激活SLIP链接，该链接就会附在sl0接口上；如果是第二次激活SLIP链接，就会附在sl1接口上，以此类推。目前，内核能支持的并发SLIP链接多达8条。

slattach选择的默认封装是CSLIP。大家可利用-p交换，选用其他模式。要想采用普通SLIP（无压缩的），就应该用

```
# slattach -p slip /dev/cua3 &
```

其他模式是：cslip、slip6、cslip6（6位版本的SLIP）和适用于SLIP的自适应（adaptive）模式。后者留在内核中，用于查找远程终端采用的SLIP封装类型。

注意，你采用的封装模式必须和你的对等体一样。比如，如果 cowslip采用的是CSLIP，你也必须如此。如果不一样，发给远程主机的 ping就不能收到自远程主机返回的包。如果其他主机ping你，你的控制台上可能会出现这样的信息：“不能建ICMP报头”。为避免此类情况的发生，必须采用自适应SLIP模式。

事实上，slattach不仅允许你启用SLIP，还允许其他协议利用串行线路，比如PPP和KISS（“火腿”无线电发烧友使用的另一个协议）。有关详情，请参阅slattach手册。

从串行线路上转入SLIP驱动程序之后，还需要配置网络接口。再次提醒大家注意，我们是利用ifconfig和route命令来完成这一配置的。假设现在已经从 vlager，拨号连接到一个名为cowslip的服务器，然后执行

```
# ifconfig sl0 vlager pointopoint cowslip
# route add cowslip
  # route add default gw cowslip
```

第一个命令把网络接口配置为到 cowslip服务器的点到点链接，第二和第三个命令将cowslip作为一个网关，增加了一条到cowslip的路由和一条默认路由。

取消SLIP链接时，必须先用带有del选项的route命令，删除所有通过cowslip服务器的路由，然后再取消网络接口，发送slattach挂起信号。最后，再次利用终端程序，将Modem挂起。

```
# route del default
# route del cowslip
# ifconfig s10 down
# kill -HUP 516
```

6.3 dip的使用

接下来的事情非常简单。但是，人们常常希望上述步骤能够自动进行，比如说，只须调用一个简单的命令，便可执行上面的所有步骤。这种想法可行吗？当然，这正是 dip的绝活（dip的意思是拨号IP，由弗雷德·范·肯蓬编写）。当前发布的dip版本是3.3.7。

dip为一种处理Modem的简单脚本编写语言提供了一个翻译器，能把串行线路转换为 SLIP

模式，并配置网络接口。这是非常原始而非常受限的，但对大多数情况来说，却是行之有效的。dip的下一个版本可能会扩展到其他语言。

为了能配置SLIP接口，dip提出“ root ”权限要求。现在，它开始试着建立到“ root ”的dip setuid，所以所有的用户都可以在没有root访问权的情况下，拨号连到特定的SLIP服务器。这是非常危险的，因为利用dip安装的bogus接口和路由可能导致网络上分发的路由混乱不堪。更为糟糕的是，可能会造成有些用户随意连接到任何一台 SLIP服务器，并向你的网络发动恶意攻击。所以，如果你想允许自己的用户向各个 SLIP服务器发起SLIP连接，并编写交换程序的话，就应该利用建立连接的特定脚本，让这些交换器来调用 dip。然后，这些程序就可以成为setuid root了（diplogin也可以，同时也必须运行setuid，参见本章的最后一节）。程序6-1中包含了一个dip脚本示例。

程序6-1 dip脚本示例

```
# Sample dip script for dialing up cowslip
# Set local and remote name and address
get $local vlager
get $remote cowslip

port cua3                # choose a serial port
speed 38400              # set speed to max
modem HAYES              # set modem type
reset                    # reset modem and tty
flush                    # flush out modem response

# Prepare for dialing.
send ATQ0V1E1X1\r
wait OK 2
if $errlvl != 0 goto error
dial 0123456789
if $errlvl != 0 goto error
wait CONNECT 60
if $errlvl != 0 goto error

# Okay, we're connected now
sleep 3
send \r\n\r\n
wait ogin: 10
if $errlvl != 0 goto error
send Cvlager\r
wait ssword: 5
if $errlvl != 0 goto error
#better not leave your password in ascii (thanx noud)
password
wait running 30
if $errlvl != 0 goto error
#to set up your remote and local IP
get $remote remote
print remote = $remote
if $errlvl != 0 goto error
```

```
wait to 3
get $local remote
print local = $local
if $errlvl != 0 goto error


# We have logged in, and the remote side is firing up CSLIP.
print Connected to $remote with address $rmtip
default                    # Make this link our default route
mode CSLIP                  # We go to CSLIP mode, too
# fall through in case of error
error:
print CSLIP to $remote failed.
```

在上面的示例中可产生一个样本脚本。该脚本可被用来连接到 cowslip，方法是用脚本名作参数引发dip。

```
# dip cowslip.dip
DIP: Dialup IP Protocol Driver version 3.3.7 (12/13/93)
Written by Fred N. van Kempen, MicroWalt Corporation.

connected to cowslip.moo.com with addr 193.174.7.129
#
```

连接到cowslip服务器，启用CSLIP链接之后，dip便离开终端，转入后台运行。之后，你就可以开始利用CSLIP链接上的普通网络服务。如果想中止这条链接，只须调用带有 -k选项的dip。该调用利用/etc/dip.pid:中的进程ID dip记录，向dip进程发出一个“挂起”信号（关于三字缩写词的详情，参阅newsgroup alt.tla）。

```
# kill -k
```

dip脚本编写语言中，前缀是 $符号的关键字是变量名。 dip有一个预先定义好的变量集（参见下面）。比如，$remote和$local这两个变量中，包含的是 SLIP链接所涉及的本地主机和远程主机名。

前面的脚本示例中，前两个语句是 get命令。dip中，get命令是用来设立变量的。我们的示例中，本地和远程主机名分别被设为 vlager和cowslip。

随后的五条语句设立了终端线路和 Modem。reset命令向Modem发出一个reset字串；采用贺氏标准的Modem使用的则是ATZ命令。下一个语句直接作出 Modem应答，所以下面几行的登录对话将正常进行。这里的登录对话相当直接：简单地拨叫 41988（cowslip的电话号码），再利用“ hey-jude ”这个密码，登录到 Svlager账号。wait命令使dip等待作为其第一个变量的字串；如果没有收到此类的字串，作为第二个参数的指定编号将等待数秒，直到超时为止。 if命令分布在登录过程中，以复查执行命令期间是否有错误发生。

登录后执行的命令都是 default和mode，前者令SLIP链接到通向所有主机的默认路由；后者则在线路上启用SLIP模式，并开始配置接口和路由表。

dip参考

尽管dip的应用非常广泛，但目前尚未出台标准文档。所以我们将在本小节向大家介绍一些常用的dip命令。如果在测试模式下调用 dip，并输入help命令，就能看到关于dip所有命令的

总述。如果想了解各个命令的相关语法，输入该命令时，不带任何参数即可；当然，不带参数的命令是没有用的。

```
DIP help
DIP knows about the following commands:

databits default  dial     echo     flush
get      goto     help     if       init
mode     modem    parity   print    port
reset    send     sleep    speed    stopbits
term     wait

DIP echo
Usage: echo on|off
DIP
```

随后的描述中，显示DIP>提示行的示例将向大家展示如何在测试模式下输入命令，以及该命令产生的输出又是什么。没有提示行的示例则当作脚本引用使用。

1. Modem命令

许多dip命令都提供了对串行线路和 Modem的配置。有的是显式的，比如用于选定串行端口的port命令，以及用于设立命令行参数的 speed、databits、stopbits和parity命令。

modem命令用于选定 Modem类型。目前为止，只提供了对贺氏 (HAYES，要求大写）标准的支持。执行dip时，必须为它提供一个 Modem类型，不然，它会拒绝执行 dial和reset命令。reset命令向Modem发出一个reset字串；这个字串的用法和选定的 Modem类型有关。对采用贺氏标准的Modem来说，则是ATZ。

刷新（Flush）代码可用来刷新 Modem发出的所有应答信号。不要把它和 reset后面的对话脚本搞混淆了，因为后者读取的是从前一条命令返回的 OK应答。

拨号之前，须通过 init命令选定一个准备传递给 Modem的初始化字串。贺氏 Modem的默认字串是“ ATE0 Q0V1 X1 ”，它将打开命令和长结果代码应答，并选定屏蔽拨号（即不复查拨号声音）。

最后，dial命令把选定的初始化字串发给 Modem，并拨叫远程系统。贺氏 Modem的默认dial命令是ATD。

2. echo和term命令

echo命令实际上是一个调试助理，利用 Echo On能使dip把发给串行设备的全部内容都反映到控制台。如果想关闭反映功能，调用 Echo Off即可。

dip还允许暂时性离开脚本模式，进入终端模式。在终端模式下，可以像使用其他普通终端程序一样，使用 dip，在串行线路上实施读写操作。要退出终端模式，输入Ctrl+]即可。

3. get命令

dip方法中，get命令用于设立变量。最简单的方式是把变量设为常量，就像前一个示例那样。但是，也可以指定关键字询问来代替值，提示用户输入变量值：

```
DIP get $local ask
Enter the value for $local:
```

第三种方式是试着从远程主机中取得一个变量值。虽然这种方式有些离谱，但在某些情

况下，却是非常有用的：比如，有的 SLIP服务器禁止在SLIP链接上使用你自己的IP地址，只要你一拨号，它就会为你分配一个取自地址缓冲池的地址，屏幕上还会提示你已经分配地址。如果屏幕上的提示消息像这样：“ 你的地址： 193.174.7.202 ”，下面的dip代码段就会让你选定这个地址；

```
wait address: 10
get $locip remote
```

4. print命令

这个命令用于向控制台反映 “ 已经开始使用dip ”。任何dip变量都可用于print命令中，比如

```
DIP print Using port $port at speed $speed
Using port cua3 at speed 38400
```

5. 变量名

dip只能理解预先定义好的变量集。所以变量名总是以美元符号（ $）为前缀，而且必须采用小写形式。

$local和$locip变量中包含本地主机名和IP地址。主机名的设立会使dip将合法的主机名保存在$local内，与此同时，为$locip分配相应的IP地址。类似的情形常发生于设立$locip变量时。

$remote和$rmtip变量和上面的两个变量差不多，只不过对象换为远程主机名及其地址。$mtu变量中有针对连接的MTU值。

上面这5个变量是唯一能用get命令直接进行分配的变量。主机的其他变量只能通过相应的命令来设立，但这些变量也可当作打印语句来使用，它们是： $modem、$port和$speed。

至于$errlvl，我们可通过该变量访问上一条命令的执行结果。如果错误级为 0，则表示命令成功，若是非零值，则表示有错。

6. if和goto命令

至于if命令，与其说它是通常调用 if条件的方法，倒不如说它是一个条件方法。它的语法如下：

```
if var op number goto label
```

上面的表达式中，必须对 $errlvl、$locip或$rmtip变量进行一个简单的比较。第二个运算对象必须是一个整数；运算符可以是： ==、!=、<、>、<=和>=。

goto命令的作用是令脚本在出现标签的那一行之后，继续执行下去。标签必须作为首要的令牌出现在行上，而且后面必须紧跟一个冒号（ :)。

7. send、wait和sleep命令

这三个命令有助于实施 dip中的简单对话脚本。 send向串行线输出自己的参数。该命令不支持变量，但能够识别全部 C样式的反斜杠字符序列，比如 n和b。平铺字符（~）用作回车／新行的缩写。

wait采用一个词来作为自己的参数，并对串行线路上的所有输入进行查看，直到认出自己需要的那个词为止。这个词本身不含任何空格。还可以为 wait指定一个超时值，作为它的第二个参数；如果规定时间内没有收到 wait需要的词，该命令就会随值为 1的$errlvl变量返回。

8. mode和default命令

这两个命令用于将串行线路置入 SLIP模式，并配置接口。

mode命令是dip在进入daemon模式之前所用的最后一个命令。如果不出错的话，这个命令

就不会返回。

mode采用一个协议名作为自己的参数。目前的 dip能够把SLIP和CSLIP识别为合法的协议名。但dip的当前版本还不能识别自适应 SLIP。

在串行线路上启用SLIP模式之后，dip便执行ifconfig，把接口配置为点到点链接，并调用route，设立到远程主机的路由。

另外，如果在对话脚本执行 mode命令之前，就执行default命令，dip同样会令默认路由指向SLIP链接。

6.4 运行于服务器模式

SLIP客户机的安装是最困难的一步。相对而言，把自己的主机配置为 SLIP服务器，要简单的多。

安装SLIP的方法之一：在服务器模式下使用 dip，这是通过把它作为 diplogin，再对它进行调用的方式来完成的。其主要配置文件是 /etc/diphosts，该文件把登录名和为该主机分配的地址关联在一起。还有一种方法是：利用 sliplogin，它是一个衍生于BSD的工具，其特色在于它是一种更为灵活的配置方案，允许你在主机连接或取消连接时，执行外壳脚本。不过，这一方案正处于测试阶段。

两种方法都要求你为每个 SLIP客户机设立一个登录账号。比方说，如果你为dent.beta.com的Authur Dent提供了SLIP服务，就要在你的passwd文件中添加下面一行，创建一个名字为dent的账号：

```
dent:* 501:60:Authur Dent ' s SLIP account:/tmp:/usr/sbin/diplogin
```

然后，再用passwd工具设立dent的密码。

现在，dent登录时，dip就会作为它的服务器，开始启动。为查看他是否拥有 SLIP使用许可，它会在/etc/diphosts文件中查找这个用户名。该文件详细记载了每个 SLIP用户的访问权限和连接参数。比如，dent的登录条目是这样的：

```
dent::dent.beta.com:Authur Dent:SLIP,296
```

用冒号隔开的字段中，第一个是用户必须采用的登录名。第二个字段中有一个额外的密码（见下面）。第三个字段是呼叫方主机的主机名或 IP地址。下一个字段是一个没有（目前还没有）特别含义的信息字段。最后一个字段描述的是连接参数。它用逗号隔开，逗号前指定的是协议名（目前，只能是SLIP或CSLIP），逗号后则是MTU（最大传输单元）。

dent登录时，diplogin便从diphosts文件中抽出关于他的相关信息，如果第二个字段非空，就会提示用户输入外部安全密码。然后，它会将用户输入的字串与 diphosts文件中的（未加密）密码进行比较。如果不相同，就会拒绝该用户登录。

另一种方法是：将串行线路置入 CSLIP或SLIP模式，diplogin继续进行，并开始配置接口和路由。这条链接会一直存在，直到用户取消链接或 Modem掉线为止。然后，diplogin将串行线返回普通线约束，并退出。

diplogin要求用户拥有超级用户特权。如果没有运行 diplogin setuid根的dip，就应该令diplogin做一份独立的dip备份，而不是作一个简单的链接。然后，diplogin就可以在不影响dip本身状态的前提下，安全地进行 setuid。