

## 附录B Linux

作者：Tim Parker

本附录内容包括：

- 准备配置TCP/IP
- 网络接口访问
- 名字服务与域名解析
- 网关
- 配置SLIP和PPP

Linux是基于Intel芯片流行的Unix版本。目前大多数Linux版本都得到广泛应用，因为Linux根植于Unix，所以在TCP/IP系统中，它既可以当作客户，也可以是服务器。本章不对Linux的操作系统原理做深入讲解，仅讲述如何配置Linux，使之成为基于TCP/IP网络的客户或服务器，同时讲述如何配置Linux系统的PPP和SLIP。

本章使用的Linux版本为Slackware，它可在SlackWare光盘中找到，同时，本书讲述的过程也适用于大多数Linux版本如RedHat等。我们假定用户已安装了Linux及其网络组件。

### B.1 准备配置TCP/IP

在配置系统的TCP/IP之前，用户需要预先执行一些操作以确保系统准备就绪。第一步是检查网络软件是否正确安装。用户可以在安装程序中安装网络包，如图B-1所示。在Linux安装过程中选择网络选项安装TCP/IP应用。在网络组件安装后，需要重启系统。



图B-1 在Linux安装程序中安装网络软件

在某些版本的Linux(使用Net-2内核或许多较新的内核)需要/proc文件系统保证网络正常工作。许多Linux内核可以支持网络功能，它们在操作系统安装时自动创建 /proc文件系统。因此用户无需做任何操作，只需查看内核是否正确加载即可。( /proc文件系统是内核获取网络信息的快速接口点，同时 /proc/net子目录帮助内核维护各种表)。通过访问 /proc文件系统查看它是

否存在，如图B-2所示。

```
merlin:~# cd /proc
merlin:/proc# ls
1/          46/          73/          80/          kcore        poi
174/        49/          74/          opuinfo      kmsg         self/
24/         51/          75/          devices      ksyms        stat
38/         55/          76/          dma          loadavg      uptime
40/         6/           77/          filesystems  meminfo      version
42/         7/           78/          interrupts   modules
44/         72/          79/          ioports      net/
merlin:/proc#
```

图B-2 如果用户可以进入/proc目录且列出目录下的文件，表明文件系统存在，TCP/IP可正确配置

如果用户不能访问/proc目录(假设用户有访问权限)，表明/proc不存在。如果在Linux安装过程中未创建/proc文件系统，就需要重新编译内核并选择 /proc选项。进入Linux资源目录(/usr/src/linux)运行下述命令重新配置内核：

```
make config
```

当系统询问是否需要支持 procfs时(或类似的问题)，选择yes。如果系统未提出此类问题，且文件系统中没有/proc目录，则用户需要升级内核以支持网络。

当Linux系统启动时，系统应自动加载/proc文件系统。为了强制系统自动加载/proc文件系统，用户需编辑/etc/fstab文件，在文件中加入下述行(如果不存在)：

```
none /proc proc defaults
```

在配置TCP/IP之前，用户还需设置系统主机名。通过以下命令设置主机名：

```
hostname name
```

其中名字为用户为本地机取的名称。如果用户有全称域名，就可以使用此名称作为本地机名称。例如，如果用户主机处在 yacht.com域中且名称为spinnaker，就可以使用下述命令设置全称域名：

```
hostname spinnaker.yacht.com
```

如果用户没有全称域名，可以设置自己的域名，前提是主机未连入 Internet。(此时域名与互联网中的域名没有任何关联)。用户没有必要为自己的主机设置域名，用户可以采用下述命令简单地设置主机名：

```
hostname spinnaker
```

此时，在/etc/hosts文件中将自动添加一项以表示本地主机名。用户需要确认 /etc/hosts文件是否已正确修改；同时，还需要知道分配给本机的 IP地址。用户需要一个独一无二的 IP地址以保证正确配置TCP/IP。

在配置TCP/IP时，如果用户需要将信息传送到多个网络，则需要修改 /etc/hosts文件是否已正确修改；同时，还需要知道分配给本机的 IP地址。用户需要一个独一无二的 IP地址以保证正确的配置TCP/IP。

在配置TCP/IP时，如果用户需要将信息传送到多个网络，则需要修改 /etc/networks文件。/etc/networks文件中包含大量网络名及对应的 IP地址。应用程序通过此文件确定目标网络的网络名。/etc/networks文件由两列组成，一列是网络名，一列是对应的 IP地址。大多数 /etc/networks文件至少包含一项，即 loopback(loopback是大多数Linux应用的缺省IP地址)。以下是/etc/networks文件示例：

```

loopback      127.0.0.0
merlin-net    147.154.12.0
BNR           47.0.0.0

```

本例中包含两个网络名及其 IP 地址。注意，只列出网络的 IP 地址，IP 地址的主机部分用 0 代替。

## B.2 网络接口访问

用户需要使网络接口可被操作系统及其应用正确使用。可使用 `ifconfig` 命令。`ifconfig` 命令使内核的网络层与网络接口协议工作，它分配 IP 地址并激活网络接口层。当接口层被激活后，内核通过网络接口收发数据。

用户需要为主机设置多个接口，其中包括 `loopback` 驱动及以太网接口（本章假设用户使用以太网，但也可使用其他接口类型）。`ifconfig` 命令依次作用于各个接口。其语法格式如下：

```
ifconfig 接口类型 IP地址
```

其中接口类型为主机接口设备驱动的名称（如 `lo` 代表 `loopback`，`ppp` 代表 `PPP`，`eth` 代表 `Ethernet`）。IP 地址接口使用的 IP 地址。

在运行 `ifconfig` 激活接口后，用户可以使用 `route` 命令增加或删除核心路由表。只有通过这一步才可使本地机查找到其他主机。`route` 命令的语法格式如下：

```
route add|del IP地址
```

其中 `add` 用于添加路由，`del` 用于删除路由，IP 地址为远程路由的地址。

用户可以仅使用 `route` 命令，不带任何参数显示核心路由表的信息。例如，如果用户仅安装了 `loopback` 驱动，`route` 命令显示如下：

```

$ route
Kernel Routing Table
Destination Gateway Genmask Flags MSS Window Use Iface
loopback    *      255.0.0.0 U    1936 0      16 lo

```

其中包含以下部分，目标名称显示配置目标（本例为 `loopback`）的名称、掩码（`Genmask`）及接口（`Iface`，本例为 `/dev/lo`）。用户可以使用 `-n` 属性显示 IP 地址：

```

$ route -n
Kernel Routing Table
Destination Gateway Genmask Flags MSS Window Use Iface
127.0.0.1   *      255.0.0.0 U    1936 0      16 lo

```

如本书前面提到的，一个典型的 Linux 网络配置包括 `loopback` 接口（存在于任何主机）及一个网络接口如以太网接口。下面我们依次讨论这两种接口。

### B.2.1 配置 `loopback` 接口

`loopback` 接口存在于任意一台主机。它用于需要 IP 地址的应用中以保证其正确运行，如果 Linux 系统不配置网络，这些应用将不存在。`loopback` 应用也被用于某些 TCP/IP 应用的诊断，它一直使用 IP 地址 127.0.0.1，因此，在 `/etc/hosts` 文件中应包含这一接口的对应项。`loopback` 驱动在软件安装时由内核创建，检查 `/etc/hosts` 文件用户可找到以下行（或类似行）：

```
localhost 127.0.0.1
```

如果存在上述行，说明 loopback 已经存在，用户可以直接配置以太网接口。如果用户不能确定，可以使用 ifconfig 命令显示 loopback 驱动的所有信息。命令如下：

```
ifconfig lo
```

该命令显示多行信息。如果显示错误信息，就表明 loopback 驱动不存在。

如果 loopback 接口没有在 /etc/hosts 文件中有相应的项，用户需要使用 ifconfig 命令创建它，命令如下：

```
ifconfig lo 127.0.0.1
```

命令在 /etc/hosts 文件中创建相应的行，可以查看由 ifconfig 命令新创建的行。例如：下述命令显示 loopback 驱动的典型配置：

```
$ ifconfig lo
lo                Link encap: Local Loopback
inet addr 127.0.0.1 Bcast {NONE SET} Mask 255.0.0.0
UP BROADCAST LOOPBACK RUNNING MTU 2000 Metric 1
RX packets:0 errors:0 dropped:0 overruns:0
TX packets:0 errors:0 dropped:0 overruns:0
```

如果可以通过 ifconfig 命令显示 loopback 驱动的细节信息，则接口工作正常。在使用 ifconfig 命令检查之后，用户需要在核心表中添加 loopback 驱动。命令如下：

```
route add 127.0.0.1或
```

```
route add localhost
```

以上两个命令的功能相同。用户可以使用 ping 命令迅速查看 loopback 驱动工作是否正确，且路由工作是否正常。命令如下：

```
ping localhost
```

其输出如下所示：

```
PING localhost: 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0.  ttl=255 time=1 ms
64 bytes from 127.0.0.1: icmp_seq=1.  ttl=255 time=1 ms
64 bytes from 127.0.0.1: icmp_seq=2.  ttl=255 time=1 ms
64 bytes from 127.0.0.1: icmp_seq=3.  ttl=255 time=1 ms
64 bytes from 127.0.0.1: icmp_seq=4.  ttl=255 time=1 ms
64 bytes from 127.0.0.1: icmp_seq=5.  ttl=255 time=1 ms
64 bytes from 127.0.0.1: icmp_seq=6.  ttl=255 time=1 ms
64 bytes from 127.0.0.1: icmp_seq=7.  ttl=255 time=1 ms
^C
localhost PING Statistics
7 packets transmitted, 7 packets received, 0% packet loss
round-trip (ms) min/avg/max = 1/1/1
```

使用 Ctrl+C 可以中断 ping 命令的执行。如果 ping 命令没有正常输出，localhost 不可识别，则用户需要检查文件的配置及路由表配置。

## B.2.2 配置以太网接口

配置以太网接口的过程与 loopback 驱动的配置类似。首先使用 ifconfig 命令添加接口至内核，然后修改路由表信息。如果用户上网，可以直接用 ping 命令检测连接是否正确。

使用ifconfig命令配置以太网接口并激活接口时,其参数为以太网设备名(eth0)及其IP地址。例如,使用如下命令:

```
ifconfig eth0 147.123.20.1
```

命令执行后,IP地址为147.123.20.1。用户不需要使用ifconfig命令设置子网掩码,系统可根据IP地址推算出正确的网络掩码值。如果用户需要使用特殊的子网掩码值,在ifconfig命令中添加netmask关键字。

```
ifconfig eth0 143.123.20.1 netmask 255.255.255.0
```

用户可以使用ifconfig命令加上以太网接口名称查看接口是否正常:

```
$ ifconfig eth0
eth0                        Link encap: Ethernet HWaddr
inet addr 147.123.20.1 Bcast 147.123.1.255 Mask 255.255.255.0
UP BROADCAST RUNNING MTU 1500 Metric 1
RX packets:0 errors:0 dropped:0 overruns:0
TX packets:0 errors:0 dropped:0 overruns:0
```

用户可以看到输出中包含广播地址,它基于本地机的IP地址。TCP/IP使用广播地址访问局域网的所有计算机,消息传输单元(MTU)通常设为最大值1500(以太网)。

下一步需要在核心路由表中添加相应的项使内核知道本地机的网络地址。route命令使用的IP地址是网络地址,不包含本地机标识部分。为了一次设置整个本地网,在route命令中使用-net属性。在上例中,其命令如下:

```
route add -net 147.123.20.0
```

该命令将所有网络地址为147.123.20.0的主机加入到内核可访问主机列表中。如果用户不采用此方法,则需要手工敲入网络中每台主机的IP地址。另外一种方法是使用/etc/networks文件指定IP地址的网络部分。/etc/networks文件中包含一系列网络名及IP地址的对应表。如果用户在/etc/networks文件中包含foobar\_net项,可以使用下述命令将整个网络加入到路由表中:

```
route add foobar_net
```

使用/etc/networks文件存在安全隐患,它使网络中所有主机都可获得访问权。这可能并非用户希望的情况。

route命令修改了核心路由表后,用户可以使用以太网接口。在使用ping命令检测时(假设用户已经连接入网),可以使用IP地址,也可使用主机名(它由/etc/hosts文件解析或由类似于DNS的服务解析)。命令及输出如下:

```
tpci_sco1-45> ping 142.12.130.12
PING 142.12.130.12: 64 data bytes
64 bytes from 142.12.130.12: icmp_seq=0. time=20. ms
64 bytes from 142.12.130.12: icmp_seq=1. time=10. ms
64 bytes from 142.12.130.12: icmp_seq=2. time=10. ms
64 bytes from 142.12.130.12: icmp_seq=3. time=20. ms
64 bytes from 142.12.130.12: icmp_seq=4. time=10. ms
64 bytes from 142.12.130.12: icmp_seq=5. time=10. ms
64 bytes from 142.12.130.12: icmp_seq=6. time=10. ms
^C
142.12.130.12 PING Statistics
7 packets transmitted, 7 packets received, 0% packet loss
round-trip (ms) min/avg/max = 10/12/20
```

如果未得到正确输出, 检查网络连接是否正确, IP地址是否正确。如果一切正常, 检查配置及路由器, 然后再使用 ping命令。

经过上述步骤, 用户的 Linux系统可以通过 TCP/IP访问局域网的其他主机。如果用户所在的网络较小, 只需作上述工作即可。若用户处在较大的网络中, 或使用特殊的协议或使用网关时, 就还需要采取更多的配置步骤。下两节将分别讲述这些步骤。

如果用户需要使 TCP/IP网络中的其他主机也可访问 Linux主机, 可将这些主机名及其 IP地址放入 /etc/hosts文件中。图 B-3显示了 /etc/hosts文件的示例, 其中包括名称及名称的变化 (如 godzilla和godzilla.tpci)及其 IP地址。这些主机 (可以是任何运行 TCP/IP的操作系统) 可以使用 telnet、ftp或类似的应用连入 Linux系统。当然, 除非它有正确的帐号, 否则无法从远程主机登录到 Linux系统。如果远程主机名存在于 /etc/hosts文件中, 则用户也可使用主机名或 IP地址 telnet或ftp到该主机。

```
merlin:/etc# cat hosts
#
# hosts          This file describes a number of hostname-to-address
#                mappings for the TCP/IP subsystem.  It is mostly
#                used at boot time, when no name servers are running.
#                On small systems, this file can be used instead of a
#                "named" name server.  Just add the names, addresses
#                and any aliases to this file...
#
# By the way, Arnt Gulbrandsen <agulbra@nvg.unit.no> says that 127.0.0.1
# should NEVER be named with the name of the machine.  It causes problems
# for some (stupid) programs, iro and reputedly talk. ;)
#
# For loopbacking.
127.0.0.1        localhost
147.120.0.1      merlin.tpci.com merlin
147.120.0.2      pepper pepper.tpci.com
147.120.0.3      megan megan.tpci.com
147.120.0.4      godzilla godzilla.tpci.com
# End of hosts.
merlin:/etc#
```

图B-3 /etc/hosts文件使远程主机连入Linux服务器

## B.3 名字服务及名字解析

TCP/IP使用/etc/hosts文件将主机名解析为 IP地址。例如, 当目标主机名为 darkstar时, TCP/IP检查/etc/hosts文件是否包含该主机名。若在文件中, 则返回 IP地址。如果名字不在文件中, 则不能向该主机发送数据。

假设用户与多个不同的主机相连, 在 /etc/hosts文件为每台主机添加相应的项既枯燥也困难。同时还需维护文件以确保主机发生变化时, 文件也跟着修改。为了解决这一问题, 开发了以下服务。

BIND(伯克利互联网名字域服务)用于将主机名解析为 IP地址。BIND最终发展成为今天的 DNS(域名服务系统), 它的功能更强。大多数 Linux系统均使用 BIND, 但也有少数系统采用 DNS。不论是 BIND还是 DNS, 均是复杂的系统, 包含许多细节。对于大多数 Linux用户, 这些细节无需了解。因此, 本节讨论如何使用户的 Linux系统使用 BIND或 DNS。

配置 BIND或 DNS的过程比较枯燥, 它仅用于 /etc/hosts不能满足用户需求时。例如, 如果用户连接的主机仅有 10台左右, 维护 /etc/hosts文件远比配置 BIND简单。对于较大的网络, 或用户需要运行 Linux上所有的服务, 则需要配置 BIND。幸运的是, Linux中 BIND仅需配置一



次。用户需要BIND软件，它通常包含在Linux软件中。BIND包中包含所有的文件及可执行文件，并且还有BIND操作人员指南(BIND Operator's Guide, BOG)。

配置BIND或DNS的细节太多，在此不作论述。

## B.4 网关

当两个或更多的局域网互联时，需要使用网关。网关是一台主机，它作为两个网络间的连接器，根据目标主机的IP地址路由数据。当用户主机需要使用网关时，需要修改网络配置文件，在将主机作为网关时，也需要修改配置文件。

当使用其他主机作为网关时，用户需要在路由表中添加网关及其连接网络的项。网关最简单的用法是使其连接其他所有网络。可通过以下route命令实现：

```
route add default gw net_gate
```

其中net\_gate为局域网中作为网关的主机名称。网关主机名放在route命令的gw关键字之后。命令中default关键字表明核心路由表将所有网络的出入报文均通过网关。

如果用户需要配置连入其他网络的网关，其网络名必须在/etc/networks文件中。例如，假设名称为gate\_serv的网关连接本地网络及相邻的称为big\_corp的网络(/etc/networks文件中包含big\_corp及其对应的IP地址)，用户需要配置本地机的路由表，可使用以下命令使本地主机使用网关gate\_serv访问big\_corp网络：

```
route add big_corp gw gate_serv
```

在远程网络的路由表中也应包含本地网络地址的相应项，否则，用户只能发送数据，而不能接收数据。

如果用户想要将本地机配置为网关，则需要在两个连接的网络分别配置连接。这通常需要两个网络适配器，PPP连接或SLIP连接。假设用户主机为small\_net和big\_net两个网络之间的网关，且主机上已经安装了两块网卡。用户已经为两块网卡分别配置了网络IP地址(例如，big\_net网络中的IP地址为163.12.34.36，而在small\_net中的IP地址为147.123.12.1)。

用户需要将两个网络地址添加到/etc/hosts文件中以简化网络名字解析过程，例如，对于上面提到的网络及IP地址，就需要在/etc/hosts文件中添加以下两项：

```
163.12.34.36      merlin.big_net.com merlin-iface1
147.123.12.1     merlin.small_net.com merlin-iface2
```

在本例中，我们在/etc/hosts文件中添加了全称域名(假设主机在两个网络中均称为merlin)。用户也可以采用简短的名称(如merlin、netlin.big\_net等)。最后，还需添加网络接口名(merlin-iface1为merlin主机上的第一个接口，merlin-iface2为第二个接口)。

然后用户需要使用ifconfig命令配置接口与名称间的连接，其中接口名称与/etc/hosts文件中的名称一致。

```
ifconfig eth0 merlin-iface1
ifconfig eth1 merlin-iface2
```

这些命令假设以太网设备/dev/eth0为big\_net的接口，/dev/eth1为small\_net的接口。

最后，内核路由表必须更新以反映两个网络名。在上例中需要使用以下两个命令：

```
route add big_net
route add small_net
```

完成上述步骤后，用户可以使用自己的主机作为两个网络间的网关。在两个网络中的主机可以使用该主机作为连接另一网络的网关。

## B.5 配置SLIP和PPP

在通用TCP/IP配置完成后，用户可以接着配置串行线路互联网协议（SLIP）或点到点协议（PPP）。SLIP和PPP均需使用 modem与远程系统建立 modem连接。用户可在配置通用 TCP/IP文件后紧接着配置SLIP或PPP，也可以在需要时再安装SLIP或PPP访问。并非所有安装均需SLIP或PPP，但大多数互联网服务提供商均选择SLIP或PPP访问供小型系统使用。

### B.5.1 配置dummy接口

dummy接口用于当主机使用SLIP和PPP接口时，为主机提供一个假冒的IP地址。dummy接口解决了孤立的主机仅有loopback驱动(127.0.0.1)所带来的问题。虽然使用SLIP和PPP可以使主机连入到外部网络，但当接口未激活时，用户没有内部IP地址供应用使用。

创建dummy接口非常简单。如果用户主机已经被分配了一个有效的IP地址，且地址存在于/etc/hosts文件中，则只需配置该接口并创建路由即可。命令如下：

```
ifconfig dummy machine_name  
route add machine_name
```

其中machine\_name为本地主机名。它创建了与本地IP地址的连接。如果本地主机没有IP地址，用户需要在创建dummy接口前添加一个，即在/etc/hosts文件中添加主机IP地址、主机名及别名，示例如下：

```
147.120.0.34 merlin merlin.tpci.com
```

### B.5.2 配置SLIP

许多拨号互联网提供者都使用SLIP与其他网络的主机互联。当建立了modem连接后，SLIP接管并维护该连接。SLIP驱动通常为Linux内核的一部分。Linux SLIP驱动也处理CSLIP，一种压缩版本的SLIP，用于某些应用中。绝大多数Linux内核缺省安装SLIP驱动，但也存在某些Linux版本需要用户重新编译内核，并添加SLIP或CSLIP驱动。用户可以在连接的双方都使用CSLIP。大多数互联网提供商同时支持SLIP和CSLIP，但用户需首先检查它们。CSLIP包中的信息比SLIP里的信息量大，因而吞吐率高。

对于使用SLIP的Linux系统，需要占用一个串行端口。该串行端口不能作为它用。内核使用称为SLIPDISC的程序控制SLIP串行端口并阻止其他非SLIP应用程序使用该端口。

预留一个串行端口供SLIP使用的最简单方法是采用slattach程序。它采用串行端口名作为参数。例如，将第二个串行端口(/dev/cua1)供SLIP使用，可采用下述命令：

```
slattach /dev/cua1 &
```

通过命令中的“&”符号使命令在后台执行。如果不在后台执行，则该终端直到命令结束才可执行其他操作。用户可以将slattach命令放在启动文件中。

当绑定成功，系统使第一个SLIP设备/dev/sl0使用该端口。缺省情况下，系统设置SLIP端口使用CSLIP。如果用户需要改变缺省值，可在slattach命令中使用-P属性及SLIP命令：

```
slattach -p slip /dev/cua1 &
```



用户需要确信连接的两端采用相同的 SLIP形式。例如，用户不能将设备设置为 CSLIP并与其他运行SLIP的主机通信。如果SLIP版本不匹配，ping命令将失败。

在为SLIP设置串行端口后，用户可以采用与平常网络连接相同的方法配置网络接口。例如，用户主机名为merlin，远程系统名称为arthur，就可以采用下述命令：

```
ifconfig sl0 merlin-slip pointopoint arthur
route add arthur
```

其中ifconfig命令配置 merlin-slip接口(本地主机的SLIP接口)与arthur建立点到点连接。route命令将名为arthur的远程主机加入到本地路由表中。用户也可使用 route命令将arthur设置为缺省网关。

```
route add default gw arthur
```

如果用户想要使用 SLIP端口访问互联网，则需要一个 IP地址并在/etc/hosts文件中建立相应项。它为SLIP系统设置合法的互联网项。

执行ifconfig命令和route命令后，用户可以测试并使用 SLIP网络。如果用户将来想删除SLIP接口，必须删除路由项，使用 ifconfig命令关闭SLIP接口，并“杀死”slattach进程。前两步可使用下述命令完成：

```
route del arthur
ifconfig sl0 down
```

终止slattach进程必须先找到slattach进程ID(PID)(使用ps命令)，然后使用kill命令。

一些Linux版本包含名为dip (dial-up IP)的实用程序以帮助用户实现上述步骤，并为 SLIP行提供解释语言。大多数dip版本均可使用。

### B.5.3 配置PPP

PPP协议的功能远比SLIP强，除非用户系统不支持PPP，否则大多数用户均选用PPP协议。Linux将PPP功能分为两部分：一部分为高层数据连接控制 (HDLC)协议，它定义在两台主机间发送PPP数据报的规则，另一部分为PPP守护进程，名为PPPD，它在HDLC系统建立连接后处理协议。此外，Linux使用名为chat的程序呼叫远程系统，与 SLIP类似，PPP首先在两台主机间建立modem连接，然后将连接控制交给PPP。

在需要为特殊用户帐号提供安全保护时，最好选择 PPP用户。这并不是必须的，可以在PPP上使用任何帐号，但为了安全起见，用户可以考虑创建 PPP用户。首先，需要在/etc/passwd文件中添加一个新用户。为PPP帐号创建新用户的/etc/passwd文件示例如下(UID为201，GID为51)：

```
ppp:*:201:51:PPP account:/tmp:/etc/ppp/pppscript
```

在本例中，帐号被设为无口令，且 home目录为/tmp(没有文件被创建)。启动程序在/etc/ppp/pppscript文件中设置，文件示例如下：

```
#!/bin/sh
mesg n
stty -echo
exec pppd -detach silent modem crtscts
```

第一行迫使脚本在 Bourne shell中执行。第二条命令关闭所有对 PPP帐号tty的写入。stty命

令防止远程发送回显。最后，exec命令运行pppd守护进程(它处理所有PPP通信)。用户可在本节后面见到pppd守护进程及其属性。

在PPP接管和处理通信之前，用户需要与远程计算机建立 modem连接。许多应用可以处理这一工作，其中最常用的是chat。

使用chat时，用户需要告诉chat如何呼叫modem并与远程系统连接。例如，使用与Hayes兼容的modem呼叫远程系统(使用AT命令设置)，号码为555-1234，可使用下述命令：

```
chat "" ATZ OK ATDT5551234 CONNECT "" ogin: ppp word: secret1
```

所有的项均采用send-except格式，在接收后发送指定信息。chat脚本以某个特定的字符串启动，一般情况下被设为空，因为modem在没有信号的情况下不会通知用户。在空字符串后用户发送ATZ(重置)，modem返回OK后，发送dial命令。modem接收到CONNECT消息时，到远程主机的登录脚本将被执行。用户发送空字符等待ogin(login)提示符，然后发送登录名PPP，等待口令word:提示符，然后发送用户口令。在登录完成后，chat中止但连接仍然存在。

注意 脚本中为何使用“ogin”和“word”而不使用“login”和“password”？因为在远程系统中二者没有任何区别，采用“login”和“Login”也是一样。缩短“password”可使字符丢失时，不会引起上锁或会话失败。

如果连接的另一端不响应login脚本，在modem响应时，用户需要使用break强迫线路中断。

```
chat -v "" ATZ OK ATDT5551234 CONNECT ""  
ogin:-BREAK-ogin: ppp word: secret1
```

在配置PPP连接时，用户需要激活pppd。如果用户已经建立了PPP连接，且本地机已使用PPP帐号登录和远程主机，就可以启动pppd。假设本地主机使用设备/dev/cua1建立PPP连接，传输率为38400bps，用户可使用下述命令启动pppd：

```
pppd /dev/cua1 38400 crtscts defaultroute
```

命令使Linux内核将/dev/cua1上的接口切换到PPP，建立到远程主机的IP链接。其中的crtscts属性用于超过9600bps的PPP连接，切换至硬件握手。

因为用户需要chat首先建立连接，所以可以将chat命令嵌入到pppd命令中。最佳方法是在从文件中(用一个属性)读取chat脚本时嵌入。例如，用户可使用下述pppd命令：

```
pppd connect "chat -f chat_file" /dev/cua1 38400  
-detach crtscts modem defaultroute
```

在chat\_file文件中包含下述字符串：

```
"" ATZ OK ATDT5551234 CONNECT "" ogin: ppp word: secret1
```

除了增加chat命令外，pppd命令还作了相应修改。connect命令指定pppd启动时使用的拨号脚本，-detach命令使pppd不与单独应用分离，并转为后台执行。modem关键字使pppd监听modem端口(本例中线路自动掉线)并在呼叫完成后将线路自动挂起。

pppd开始设置连接参数，与远程主机交换IP地址，并设置通信值。完成以上步骤后，pppd设置Linux内核中的网络层，通过将接口设为/dev/ppp0(假设它为主机上第一个PPP连接)以使用PPP链路。最后，pppd建立内核路由表项指出PPP链接另一端的主机。

PPP有许多属性、相关的配置文件及认证过程，用户在与远程系统建立连接时可能需要了

解这些内容。请阅读相关书籍，了解详细信息。

## B.6 小结

用户采用本章讲述的步骤配置 TCP/IP，Linux 系统的 TCP/IP 应可以正常工作。用户可以使本地主机为其他 TCP/IP 服务的客户端也可以作为局域网的服务器。本章还简单讲述了 PPP 和 SLIP，它们是基于 modem 的协议，用户可以使用它们连入互联网。