

## 第11章 泰勒式UUCP

UUCP是二十世纪70年代后期，由AT&T公司贝尔实验室的Mike Lesk设计出来，用于通过公用电话线路，提供一个简单的拨号网络。由于许多人都想在自己的家中，通过 modem收发电子邮件和Usenet新闻，所以UUCP至今仍然非常流行。尽管不同的硬件平台和操作系统上运行着大量不同的实施方案，但总的说来，都不会产生冲突。

但是，若干年来，有的软件越来越“标准”，已经没有能够称为 UUCP的UUCP了。自1976年发布第一个版本以来，UUCP一直处于一个非常稳定的发展过程。目前，UUCP主要分为两大类，其区别表现在硬件支持和它们的配置上。还有其他许多实施方案，大都与这两类大同小异。

其中一类就是所谓的“版本 2 UUCP”，它起源于1977年，由Mike Lesk、David A. Novitz和Greg Chesson合作编写的第一版UUCP。尽管它相当老了，但仍然广为使用。新近发布的版本2，使最近的UUCP更为好用。

第二类是在1983开发的，常被称为BNU（基本连网程序）、HoneyDanBer UUCP或HDB。这个奇怪的名字源于其作者：P.Honeyman、D.A. Novitz和B.E.Redman。HDB被认为是消灭了版本2 UUCP之不足的功臣。例如，增加了新的传输协议，拆分了假脱机目录，能让你与之进行UUCP通信的每个站点上只有一个目录。

当前，软件厂商在其产品中捆绑的UUCP实施是Taylor UUCP 1.04（该程序是Ian Taylor于1993年编写并申请版本），它是本章所讲的版本的基础。以下将其统称为泰勒式 UUCP，其1.04版本是1993年2月发布的。除了识别传统的配置文件外，对泰勒式 UUCP稍作编译，也能够识别新式（比如a.k.a. “Taylor”）的配置文件。

本书编写过程中，最新的1.05版本也发布了，很快将进入分销渠道。这两个版本的不同之处对你而言，没什么影响，因此，你大可利用本书中的信息，配置泰勒式 UUCP 1.05。

本章的目的不是彻底论述UUCP命令的命令行选项有哪些，其作用是什么，而是向大家介绍如何设置一个正在运行的UUCP节点。第一小节简要介绍UUCP是怎样实施远程命令执行和文件传输的。

但是，我们将假设这个前提：大家对UUCP套件的用户程序有所了解。它们是uucp和uux。有关说明可参考在线手册。

除了公开供用户访问的程序、uucp和uux程序外，UUCP套件中还包含大量的只用于管理的命令。这些命令用于监控节点间的UUCP通信，删除旧的日志文件和编译特性等。详细情况未作讨论，因为我们这里的重点是UUCP。除此以外，这些命令已经编入文档，非常容易理解。但是，UUCP套件中还有第三种类型，这类由真正的UUCP“工程兵”组成，它们叫作uucico（cico表示copy-in copy-out）和uuxqt，后者执行从远程系统发来的作业。

### 11.1 关于UUCP

对于那些在本章内找不到自己需要的资料的用户来说，应该仔细阅读软件包自带的文档。

它是一个 texinfo 文件集，对利用泰勒式配置方案进行的设置进行了详细说明。利用 `tex` 和 `makeinfo`，可将 texinfo 分别转换为 DVI 和 GNU info 文件集。

如果想采用 BNU 甚至版 2 配置文件，则可参考另一本好书《Managing UUCP 和 Usenet》。我觉得这本书讲得非常透彻。关于 Linux 系统上可用的 UUCP，其详情可参考 Guyllhem Aznar（邮件地址 `guyllhem@oeil.qc.ca`）编写的 UUCP-HOWTO 文档，该文档定期在 `comp.os.linux.announce` 发布。

另外，还有一个 UUCP 新闻组，名为 `comp.mail.uucp`。如果有和泰勒式 UUCP 相关的问题，最好提交到该新闻组，让其中的专家为你解决疑难，而不是访问 `comp.os.linux`。

### 11.1.1 UUCP 传输和远程作业的执行

理解 UUCP 的关键在于搞清“作业”（jobs）一词的概念。用户利用 `uucp` 或 `uux` 发起一次传输就叫作一个作业。一次作业由一个命令和一个文件集组成，这个命令将在远程系统上执行，而文件集则将在不同站点间进行传输。

UUCP 一般不会立即调用远程系统来执行作业（或你可利用 `kermit` 执行）。相反地，它会临时性地将作业说明保存起来。这就是所谓的“假脱机”（spooling）。下面保存作业说明的目录树就被称作假脱机目录（spool directory），一般位于 `/var/spool/uucp` 内。我们的示例中，作业说明中包含与即将执行的远程命令（`lpr`）相关的信息、请求执行作业的用户和两个其他的项目。除了作业说明外，UUCP 还必须保存输入文件 `netguide.ps`。

`spool` 文件的确切位置和命名可能会依据某些编译时选项而发生变化。兼容于 HDB 的 UUCP 一般将 `spool` 文件保存在名为 `/var/spool/uucp/site` 的目录内，而“site”代表的是站点名。在针对泰勒式配置方案进行编译时，UUCP 将在站点专有的 `spool` 目录下，为不同类型的 `spool` 文件创建子目录。

UUCP 定期拨入远程系统。与远程主机建立连接之后，UUCP 将说明作业的文件和所有的输入文件传递给它。进入远程主机的作业不会立即执行，而是在连接中断之后才进行。这是由 `uuxqt` 来完成的。`uuxqt` 还会将目的地是另一个站点的作业进行转发。

为了将重要和次要作业区分开来，UUCP 为各个作业制定了级别（grade）。从 0 到 9 的级别用字母来表示，即从 A 到 Z。级别越高，越优先。邮件对应的级别一般是 B 或 C，而新闻对应的则是 N。级别越高的作业，其传输次序也优先。级别的分配是在调用 `uucp` 或 `uux` 时，利用 `-g` 标记来分配的。

某些时候，还可禁止传输低于指定级别的作业。这就是所谓的对话期间的最高 `spool` 级别，其默认设置是 `-z`。注意：只有其级别等于或高于最大 `spool` 级别的文件，才能得以传输。

### 11.1.2 uucico 的内部运行

为了解 `uucico` 需要知道某些情况的原因，先简要谈谈它在事实上是如何与远程系统建立连接的。

当你从命令行执行 `uucico -s` 系统时，它首先必须有一条物理上的连接。根据连接类型，开始采取行动，比如，在使用电话线时，它必须先找到一个 `modem`。在 TCP 链接上，它必须调用 `gethostbyname(3)`，将主机名转换为网络地址，再找出应该打开的端口，并将网络地址绑定到相应的套接字上。

在链接建立之后，必须传递一个身份验证进程。这个进程一般由要求登录名的远程系统和一个密码组成。这就是常说的“登录对话”(login chat)。身份验证进程要么由常用的getty/login套件执行，要么由uucico自己在TCP套接字上执行。如果验证成功，远程主机就会发起uucico。用于初始化链接的uucico之本地备份常被称为主管，远程备份常称为从属。

接下来是握手阶段：现在，主管主机发送其主机名和一些标记。从属便查看这个主机名是否能够登录，收发文件等等。标记说明了准备传输的spool文件的最高级别。如果一切正常，就将产生对话计数或呼叫下个号码检查。有了这一特性，两个站点便可维持成功的链接。如果文件的级别低于指定的级别，握手就会失败。这个特性特别适合于对付网络骗子。

最后，两端的uucico试图与常用的传输协议达成一致。这个协议掌管数据传输、数据一致性检查和出现错误时的重新传送数据的方式。为什么需要不同的协议呢？因为需要支持不同的连接类型。比如，由于电话线路担心出错，因此要求一个“安全”协议，而TCP传输又是天生的可靠，可以使用一个更为高效的协议，这类协议将摒弃特殊错误查找。

握手结束之后，真正的文件传输阶段开始了。连接双方都打开选定的协议驱动程序。这些驱动程序将可能执行协议特定的初始化序列。

其一，主管将排队等候远程系统的所有文件发送出去，这些文件的spool级别非常之高。发完之后，它便通知从属发送完毕，而从属此刻可能已经挂断。现在，从属要么同意挂断，要么接管登录对话。这时，角色发生了变化：此刻，远程主机成了主管，本地主机则成为从属。现在，新任的主管开始发送自己的文件。发送完毕之后，两个uucico的交换终止消息并关闭连接。

我们打算继续深入讨论。详情参考UUCP源代码或相关书籍。网上有一篇见解独到的好文章，是David A. Novitz写的，其中详细介绍了UUCP协议。泰勒式UUCP常见问题集也对UUCP的实施进行了细致的探讨。这个FAQ集定期出版于comp.mail.uucp。

### 11.1.3 uucico命令行选项

本小节只说明几个最重要的命令行选项，要想获得一份关于所有选项的完整列表，请参考uucico手册。

## 11.2 UUCP配置文件

与较为简单的文件传输程序相比，设计UUCP的目的是为了能够自动处理所有的文件传输。一旦正确设置了UUCP，管理员就不必在每日不变的一些基本操作上花太多的时间。设置UUCP所需的信息保存在两个配置文件内，这两个文件驻留在/usr/lib/uucp目录下。它们只用于拨号。

注意 本章中，大家还将看到若干个实例，其中引入的示例目前还不能用。虽然留有相关的文本说明，但最好参考它们的LDP。

### 11.2.1 泰勒式UUCP简介

说UUCP配置很难，并非言过其实。它的确涵括了许许多多的细节问题，即便配置文件的简洁格式也无法使事情变得简单（尽管与老式的HDB和版本2中的格式相比，泰勒式UUCP的格式更易于理解）。

为了让大家感觉一下这些文件是如何相互影响的，我们将介绍几个最重要的文件，看看这些文件内的条目。但现在不打算一一细讲；将在随后的几个小节内，重点介绍其中的一个。如果将自己的机器设置为 UUCP，最好从某些示范文件着手，并逐步地改编它们。示范文件可从下面的示例中选出，也可从你自己喜欢的软件产品中选出。

本小节提到的所有文件都包含在 `/usr/lib/uucp` 或子目录 `thereof` 内。有的软件产品中还有 UUCP 二进制文件（这些文件为 HDB 和泰勒式配置方案的启用提供了支持），并分别为每个配置文件采用不同的子目录。`/usr/lib/uucp` 内，一般都有一个 README 文件。

要想 UUCP 运行无误，这些文件必须只能归 `uucp` 用户所用。其中一些文件中包含密码和电话号码，因此，它们应该有 600 模式许可。

注意 尽管多数 UUCP 命令都必须 `setuid` 为 `uucp`，但你同样必须确定 `uuchk` 不是 `uucp`。否则，用户将能显示密码，即使他们拥有模式 600。

核心的 UUCP 配置文件是 `/usr/lib/uucp/config`，用于设置常规参数。它们之中，最重要的是（迄今为止，也是唯一的）你的主机之 UUCP 名。

下个重要配置文件是 `sys` 文件。该文件中包含的信息和与你链接的站点之系统有关。其中包括站点名和关于链接本身的信息（比如使用 `modem` 连接时采用的电话号码）。

`port` 命名准备使用的端口。`time` 指定何时拨号。`chat` 说明登录对话脚本——字符串顺序必须进行交换，以便允许 `uucico` 登录进入 Pablo。稍后再回头讨论对话脚本。`port` 命令不会命名 `/dev/cua1` 之类的设备专有文件名，而是命名端口文件内的条目。只要这些名字引用的是端口文件内的有效条目，你就可以按照自己的喜好，对它们进行分配。

端口文件中保存有与链接本身相关的信息，它说明了即将使用的设备专有文件名、支持的速率范围和连接到该端口的拨号设备的类型。随后的条目说明了 `/dev/cua1` (a.k.a COM2)，连接到这个设备的 NakWell modem 的速率可高达 38 400bps。选择这个条目的名称时，应该使它和 `sys` 文件指定的端口名相符。

关于拨号者本身的信息，则保存在另一个文件内，名为 `dial`。对每类拨号者，它一般都包含拨叫远程站点时执行命令的序列和具体的电话号码。再次提醒大家注意，它也可作为一个对话脚本。

对话中的第一行指定 `modem` 对话，表示命令的执行序列（这些命令或发给 `modem`，或是从 `modem` 发回），用于初始化 `modem` 并令其拨叫指定的电话号码。T 序列将由 `uucico` 用电话号码来替换。

`uucico` 的第一步是在 `sys` 文件内查找 Pablo。从 Pablo 的 `sys` 文件条目中，得知它应该采用 `serial1` 端口来建立连接。端口文件告诉它这是一个 `modem` 端口，而且已经绑定了一个 NakWell `modem`。

接着，`uucico` 便开始搜索描述 NakWell `modem` 的拨号条目，找到之后，便打开 `/dev/cua1`，执行拨号者对话。也就是说，它发出一个“ATZ”，并等待“OK”应答等。如果碰到 T 这个字符串，它就将其替换为电话号码 123-4567，这个号码取自 `sys` 文件。

`modem` 返回 CONNECT 之后，表示已经建立连接，`modem` 对话已经完成。现在，`uucico` 返回 `sys` 文件，并执行登录对话。在我们的示例中，它将等待 `login:` 提示，然后，再发自己的用户名（Neruda），等到出现 `password:` 提示时，在发自己的密码 `lorca`。

身份验证完成之后，远程端主机便假定发起自己的 `uucico`。然后，连接双方开始进入握手

阶段（参见前一小节）。

### 11.2.2 UUCP需要知道些什么

在开始编写自己的UUCP配置文件之前，必须收集一些它需要知道的信息。

首先，必须搞清楚你的 modem 所绑定的串行设备是什么。一般说来，DOS 端口 COM1 到 COM4 对应的设备特有文件是 /dev/cua1 到 /dev/cua3。许多软件（比如 Slackware），都创建了一个 /dev/modem 链接，链接到其相应的 cua\* 设备文件，配置 kermit、seyon 等，从而能够使用这些文件。这种情况下，也应该在你的 UUCP 配置文件内采用 /dev/modem 链接。

这样做的原因是所有的拨号器都采用所谓的锁文件来标识串行端口正在使用中。这些锁文件名是一连串的 LCK.. 字串和设备文件名。比如 LCK.cua1。如果拨号器对同一个设备采用了不同的设备名，它们就不能对彼此的锁文件进行识别。其结果是，在开始的那一刻，同时中断彼此的会话。这在你利用 crontab 条目，安排自己的 UUCP 调用时，是不太可能的。关于如何设置自己的串行端口，请参考第 4 章。

接下来，必须找出 modem 的传输速率。必须将其值设为自己希望得到的最大有效传输速率。有效传输速率可以稍高于 modem 规定的原始物理传输速率。比如，许多 modem 收发数据的速率都是 2 400bps。而利用 V.42bis 之类的压缩协议时，真正的传输速率可高达 9 600bps。

当然，如果希望自己的 UUCP 十项全能，肯定还需准备要拨叫的那个系统之电话号码。而且，还需要一个有效的登录 ID 和密码。

提示 如果你正准备拿 UUCP 做试验，最好能找到一个离你较近的归档站点。记下其登录名和密码——是可随意下载的。多数情况下，它们是类似于 uucp/uucp 和 nuucp/uucp 的。

另外，还必须知道如何才能真正地登录到远程系统。举个例子来说，在登录提示出现之前，有必要按 Break 键吗？登录提示显示的是“login:”，还是“user:”？这一点对编写对话脚本来说，是非常必要的，对话脚本是告诉 uucico 如何登录的“剧本”。如果不知道这一点，或平常所用的对话脚本失效了，就试着用 kermit 或 minicom 之类的终端程序来拨叫远程系统，并准确地记下你必须做的事。

### 11.2.3 站点的命名

由于有基于 TCP/IP 的连网，所以你的主机必须有一个用于 UUCP 连网的主机名。如果只想将 UUCP 用于站点间或本地网络上的文件传输，这个名就不需要符合任何标准。唯一的限制是不得超过 7 个字符，而且不得与文件系统的主机名有冲突。

但是，如果打算把 UUCP 用于邮件或新闻链接，就应该考虑使用 UUCP Mapping Project（UUCP 映射工程）注册的名字。UUCP 映射工程的详情，参见第 12 章。即使你参与了一个域，也有必要考虑为自己的站点选择一个正式的 UUCP 名。

一般说来，人们爱在自己的完整资格域名中，选择第一个组件来作为自己的 UUCP 名。比如，如果你的站点域名地址是 swim.twobirds.com，那么，你的 UUCP 主机名就是 Swim。许多 UUCP 站点都采用这一命名原则。当然，也可采用一个和自己的完整资格域名完全不相干的 UUCP 名。

但是，务必保证不要在邮件地址中使用未取得资格的站点名，除非你已经将其注册为自

己的正式UUCP名（UUCP映射工程负责对全球的所有UUCP主机名进行注册，并保证它们的唯一性。要注册自己的UUCP主机名，可询问负责处理你的邮件之站点维护人员，他们能够帮助你）。因为，如果向尚未注册的UUCP主机发邮件，最终会犹如泥牛入海。如果你使用的名称与别的站点雷同，邮件也会被路由到那个站点，对其邮发主管来说，是件非常头疼的事。

默认情况下，UUCP套件采用的主机名是站点的UUCP名。这个名字通常是在`/etc/rc.local`脚本内设置的。如果你的UUCP名和你自己设置的主机名不同，就必须利用`config`文件内的主机名选项，将你的UUCP名告知`uucico`。这是下文讨论的主题。

## 11.3 泰勒式配置文件

现在回过头来看看配置文件。

泰勒式配置文件一般由若干行组成，这些行中包含一些关键字-值对。#号代表一个批注，表示这一行是备注。如果程序本身要采用#号，就可以在#前加一个斜杠。

利用这些配置文件，可对相当多的选项进行调整。我们不打算对所有的参数进行深入讲解，只讲一些最重要的。有了它们，才可能配置基于modem的UUCP链接。如果你想在TCP/IP或直接串行线路上使用UUCP，则可参见其他小节，它们对一些必要的修正进行了说明。其全面而完整的参考资料包含在泰勒式UUCP源代码附带的`Texinfo`文档内。

当你认为自己已经完成UUCP系统的配置时，可利用`uuchk`工具（位于`/usr/lib/uucp`），对自己的配置进行校验。`uuchk`读取你的配置文件，再打印一份详细报告，报告中对每个系统所用的配置值进行了详细说明。

### 11.3.1 常规配置选项：config文件

一般情况下，除了UUCP主机名外，是不用这个文件的。默认情况下，UUCP采用的是`hostname`命令设置的主机名，但最好显式设置UUCP名。

当然，还有许多参数是可以设置的，比如假脱机目录名和匿名UUCP的访问权限等。后者将在稍后进行讨论。

### 11.3.2 如何将其他的系统告知UUCP：sys文件

`sys`文件描述的是你的机器所了解的系统。条目是系统关键字引入的；后面几行则是系统为站点指定的参数。一般说来，系统条目会对电话号码和登录对话之类的参数进行定义。

第一个系统行之前的参数设置的是用于所有系统的默认值。通常情况下，应该在默认值这部分设置协议参数等等。

下面，便为大家详细介绍最典型的几个字段。

#### 1. 系统名

`system`命令用于命名远程系统。必须指定规范的远程系统名，而不是你自行发明的别名，因为`uucico`将在你登录时，再次检查远程系统广播的系统名（版本2 UUCP在调用时，不会广播系统名，但新版本的UUCP实施通常都会这样做，泰勒式UUCP也不例外）。

每个系统名可能只出现一次。如果你想针对同一个系统，采用多个配置文件集（比如多个电话号码，让`uucico`依次拨打），就可以指定备用配置文件。下面将具体介绍备用配置。

#### 2. 电话号码

如果远程系统能够通过电话线路抵达，`phone`字段指定的就是modem应该拨叫的电话号码。

该字段内可能包含若干个记号，这些记号的解释由 uucico 过程负责。等号 = 表示等待第二次拨号音调，“-”号则产生 1 秒钟的暂停。例如，在拨打区号和电话号码之间，如果没有暂停的话，有些电话线路会中断（不知道应该用哪个术语来表达这种情况，说简单点，有点类似于公司的内部电话，必须先拨打 0 或 9 才能打出去）。

任何一个内含阿拉伯数字的字串都可用于隐藏站点比如区号之类的站点相关信息。利用 dialcode 文件，便可将此类字串翻译成 dialcode。

有了翻译文件，就可以在 sys 文件内采用 Bogoham 7732 之类的电话号码，它们更容易理解。

### 3. 端口和速率

端口和速率选项用于选定拨打远程系统的设备和设置设备的最大速率（tty 的波特率至少和最大传输速率一样高）。系统条目即可以单独采用一个选项，又可以两者都用。在端口文件内查找合适的设备时，只能选定那些有匹配端口名和 / 或速率范围的端口。

一般情况下，用速率选项就够了。如果端口文件内，只定义了一个串行设备，无论如何，uucico 始终都会选择右边的端口，你只须为它指定合适的速率。如果你的系统上备有多个 modem，通常也不愿意命名一个特殊的端口，因为 uucico 一旦发现有这么多个设备，就会依次地尝试每一个设备，直到找到没有使用的设备为止。

### 4. 登录对话

由上可知，登录对话脚本是告诉 uucico 如何登录到远程系统的脚本。它由一连串记号组成，用于指定本地 uucico 进程发出和希望收到的字串。其目的在于令 uucico 等待远程主机向其发送登录提示，随后返回登录名，等待远程系统向其发送密码提示，最后，再发送密码。expect（希望收到）和 send（发送）字串是在备用文件中指定的。uucico 自动将回车字符（r）添加到 send 字串上。

uucico 还考虑到了某些违例情况，比如，在发送提示之前，远程主机的 getty 需要进行重新设置。所以，你可将一个子对话添加到一个 expect 字串上，偏移一个“-”。只有主要的 expect 字串失败时（比如，发生超时），才执行这个子对话。利用这一特性的方法之一是：远程站点没有显示登录提示时，发送一个 BREAK。下面的例子给出了一个全面的对话脚本，如果你必须在登录提示出现之前按回车的话，这个脚本应该是有效的。Linux 系统中，要求 UUCP 不等待提示，即刻进行下一个 send 字串的发送。

### 5. 备用

有的时候，对于一个单独的系统，我们希望能够有多个配置。比如，可以通过不同的 modem 线路，接通这一系统等。利用泰勒式 UUCP，定义一个所谓的备用，就能立即实现这一想法。

备用条目中保持主要系统条目中的所有设置，并只指定了默认系统条目中那些应该改写或增添的值。备用从系统条目偏移含有关键字“备用”（alternate）的一行。

在拨打 Pablo 时，uucico 首先拨打 123-456，如果失败，再拨打备用条目中的号码。备用条目中保持有主要系统条目中的所有设置，并只改写了电话号码。

### 6. 限制拨打次数

泰勒式 UUCP 提供了许多方法，可以用来限制拨打远程系统的次数。为什么要限制拨打次数呢？其原因不外乎远程主机只在上班时间提供服务，或只是为了避免高频率的拨打次数。

注意，只要为uucico提供-S或-f选项，就能改写拨打次数限制。

默认情况下，泰勒式 UUCP不允许随时连接，所以你必须采用 sys文件内的时间规格。如果不在意拨打次数限制，可利用 sys文件内的Any值，指定time选项。

要限制拨打次数，最简单的方法是利用时间条目，这个条目后面跟一个字串，该字串由两个子字段组成：日期和时间。日期可以是 Mo、Tu、We、Th、Fr、Sa、Su，或Any、Never和代表工作日的Wk。时间则由两个24时的时钟值组成。这些时间记号之间没有空格符或制表符。任何一个日期和时间规格都必须用逗号列为一组。

特殊的时间字串Any和Never分别表示：随时拨打或从不拨打。

time命令采用了一个可选的第二个参数，该参数以分钟计，说明几分钟后重试。在连接尝试失败之后，uucico不允许在特定的时间间隔内，另行尝试拨打远程系统。默认情况下，uucico采用的是一个exponential backoff方案，也就是说，时间间隔随失败次数的增多而增大。例如，你指定的重试时间间隔是5分钟，uucico将在上次尝试失败之后的5分钟内，拒绝拨打远程系统。

timegrade命令允许你在作业安排中增加一个最大假脱机级别。这样一来，只要一建立拨号连接，假脱机级别为C或更高的作业（邮件的队列级别是B或C）就能得以传输，而新闻（其队列级别一般是N）只能在夜间或周末得以传输。

和time命令一样，timegrade命令也采用了一个重试时间间隔（以分钟计）作为其第三个可选参数。

然而，与假脱机级别有关的说明则适合这种情况：首先，timegrade命令只适用于你的系统发出的作业；远程系统仍然一如既往地传输自己的作业。因此，你可利用 call-timegrade选项，显式请求它只发送假脱机级别高于指定级别的作业；但这样不能保证它会接受一个请求（如果远程系统运行的是泰勒式 UUCP，它就会接受这一请求）。

类似地，远程系统拨入本地时，不会对timegrade字段进行检查，所以所有排队等候拨叫系统的作业都能得以发送。但是，远程系统可以显式请求你的uucico限定发送特定级别的假脱机作业。

### 11.3.3 设备：端口文件

端口文件用于告诉uucico哪些端口是可以用的。既可以是modem端口，又可以是其他类型的端口（比如直接串行线路）或已获支持的TCP套接字。

与sys文件一样，端口文件由几个独立的条目组成。首先是关键字 port，然后是端口名。这个名可以供sys文件内的端口语句使用。端口名不必是唯一的；如果几个端口同名，uucico将依次尝试于各个端口建立连接，直到找出一个目前尚未使用的为止。

port命令后面应该紧跟type语句，该语句用以说明端口类型。有效的类型有 modem、用于直接连接的direct和用于TCP套接字的tcp。如果不采用port命令，端口类型就会采用默认的modem类型。

本小节中，我们只介绍modem端口，TCP端口和直接串行线路将留在稍后进行讨论。

对modem和直接端口来说，都必须利用device directive（设备指令）指定用于拨出的设备。通常，这个设备是一个设备专有文件名，位于/dev目录下，比方说/dev/cua1这个例子就是如此。

注意 有人偏爱ttyS\*设备，这类设备只能用于拨入。



如果是modem设备，端口条目还能用于判断连接到这个端口的modem类型是什么。不同类型的modem必须采用不同的配置。即使它声明自己采用的是贺氏标准也是如此。你必须告诉uucico如何初始化modem、如何令其拨打指定的电话号码。泰勒式UUCP将所有拨号器的相关描述保存在一个名为dail的文件内。要想采用这些拨号器，必须利用dialer命令，指定拨号器名称。

根据自己拨打的系统，你可能想采用别的方式来利用modem。例如，高速modem试图以14 400bps的速率连接老式的modem时，后者不能识别它们，只是将线路断掉，而不是协商以9 600bps的速率连接。如果你得知自己准备连接的站点采用的正是这类该进博物馆的modem时，就必须另行设置自己的modem。所以，就需要在端口文件内增加一条端口条目，指定另外的拨号器。现在，可以为新端口指定一个新端口名，比如serial-slow，并采用sys文件中drop系统条目内的端口指令。

要想区分各个端口，最好看它们支持的速率。站点drop的系统条目将serial作为其端口名，但只请求以9 600bps的速率进行连接。然后，uucico自动采用第二个端口条目。最后，利用第一个端口条目，拨打系统条目中有38 400bps速率的其他所有站点。

### 11.3.4 如何拨号：拨号文件

拨号(dial)文件说明的是不同的拨号器采用的方法。过去，UUCP提到的是拨号器，而不是modem，因为早期，普遍采用一个(非常昂贵)自动拨号设备来充当modem。今天，许多modem都有内置的拨号支持，所以拨号器和modem之间的界线才日益淡化。有的甚至干脆把拨号器叫作modem。

不管怎么说，不同的拨号器和modem都可能要求不同的配置。你可将各类拨号器和modem的相关描述写入dial文件内。dial文件内的条目以dialer命令开头，该命令会给出拨号器的名称。

除dialer命令之外，另一个最重要的条目就是chat命令指定的modem chat(modem对话)。它和登录对话类似，由一连串字符组成，这些字符是由uucico发给拨号器的字串和依次返回的应答字串组成。它常用于将modem重新设置为已知状态并进行拨号。

uucico modem对话以空的expect字串开始。因此，uucico立即发送第一条命令(ATZ)。ATZ是Hayes兼容modem采用的命令，用于重新设置modem。然后，uucico等待modem发回OK之后，再发送下一条命令，这条命令将关闭本地应答，如此等等。modem再次发回OK之后，uucico便发出拨号命令(ATDT)。这个字串中的避开序号T被系统条目文件内的电话号码所代替。然后，uucico就等待modem返回CONNECT字串，后者标志着与远程modem的连接已经成功建立。

modem不能连接到远程系统，这是常有的事。比如，远程系统正忙于和别人对话，线路正忙等。这时，modem返回的错误消息，能够指出不能连接的原因。modem对话不能侦测此类消息；所以uucico将继续等待自己希望收到的字串，直到超时为止。因此，UUCP日志文件中将只能有这样一行：“timed out in chat script”(对话脚本超时)，而不能记录不能连接的真正原因。

但是，泰勒式UUCP允许你利用前面所讲的chat-fail(对话失败)命令，将错误消息告知uucico。uucico在执行modem对话期间，一侦测到chat-fail字串，就会中止拨号，将错误消息

记录在UUCP日志文件内。

前面的示例中，最后一个命令要求UUCP在开始modem对话之前，一直停留在DTR这一行。侦测到DTR行有变动时，许多modem都可被配置为挂起，并进入命令模式。

注意 也可将有些modem配置为侦测到DTR发生变化时，自行重新设置。但是有的则不能这样，有时还会自行挂断。

### 11.3.5 TCP上的UUCP

利用UUCP，在TCP链接上传输数据，乍一听，有些可笑。事实上，这并不是个坏主意，特别是在传输 Usenet新闻组之类的大型数据时。在基于 TCP的链接上，新闻通常是用 NNTP协议来交换的，在 NNTP协议中，文章的请求和发送是单独进行的，无须压缩和任何优化。尽管这对大型的站点来说，可以同时接收多个新闻传输，但对通过速度较慢的连接（比如 ISDN）接收新闻的小型站点来说，这种方法并不理想。这类站点通常想综合利用 TCP的传输质量和大批量发送新闻的好处，也就是说既能够对新闻进行压缩，又能够降低传输开销。传输批量新闻的标准作法是在TCP链接上使用UUCP。

### 11.3.6 直接连接的使用

假设你采用一条直接线路将自己的系统 vstout连接到tiny。与modem连接的情况极为相似，必须在sys文件内写入一条系统条目。port命令标识串行端口tiny已处于连接状态。

端口文件中，必须对直接连接所用的串行端口进行说明。这里不需要拨号器条目，因为没必要进行拨号。

## 11.4 UUCP的注意事项：调节权限

### 11.4.1 命令执行

UUCP的任务是把文件从一个系统复制到另一个系统，并请求在远程主机上执行特定的命令。当然，作为管理员的你，肯定想对各系统的命令执行权进行控制——允许它们在你的系统上执行所有命令并不是件好事。

默认情况下，泰勒式 UUCP允许其他系统在你的机器上执行的命令只有两条，那就是 rmail和rnews。这两个命令常用于在 UUCP上交换电子邮件和 Usenet新闻。uuxqt采用的默认搜索路径是一个编译时选项，但其中通常应该包含 /bin、/usr/bin和/usr/local/bin。要针对特定的系统改变命令设置，可采用 sys文件内的命令关键字。类似地，也可以用 command-path（命令路径）语句，改变搜索路径。例如，可以让系统 Pablo在执行rmail和rnews命令之外，还执行 rsmtp命令（ rsmtp用于投递带有批量SMTP的邮件；有关详情，将在后续的邮件章节中讨论）。

### 11.4.2 文件传输

泰勒式UUCP还允许更为详细地优化文件传输。甚至还可以取消将文件传输到特定系统或禁止从特定的系统传输文件。只须把请求设置为 No，远程系统就不能向你的系统上传或下载文件。类似地，你也可将传输设置为 No，禁止你的用户向特定的系统上传或下载文件。默认

情况下，本地和远程系统上的用户都可上传和下载文件。

另外，可以配置准备复制文件的目录。通常情况下，作为系统管理员的你，多半打算限制远程系统用户访问你的目录结构，但仍允许你自己的用户发送其根目录中的文件。此时，远程系统用户将只能接收公用 UUCP 目录（`/var/spool/uucppublic`）中的文件。这个目录中常放置一些公用文件，与因特网上的 FTP 服务器非常类似。这个目录常用“~”符号来表示。

为此，泰勒式 UUCP 为收发文件提供了四个用于配置目录的不同命令。它们是：

`local-send`（指定一份目录清单，用户要求 UUCP 发送这一系列目录中的文件）、`local-receive`（给出一份目录清单，用户要求 UUCP 将收到的文件放入这些目录）、`remote-send` 和 `remote-receive`，对远程系统的收发请求进行类似的处理。

`local-send` 命令允许你的用户将 `/home` 目录下和公用 UUCP 目录中的所有文件发给 Pablo。`local-receive` 命令允许他们接收发向人人可写的目录或 `/home` 目录下的人人可写文件，这些人人可写的目录在 `uucppublic` 的 `receeive` 目录中。`remote-send` 指令允许 Pablo 请求 `/var/spool/uucppublic` 内的文件，但 `incoming` 和 `receive` 目录下的文件除外。`uucico` 是怎样得知这一点的呢？原来目录名前面有一个感叹号！最后一行允许 Pablo 将任何文件上传到 `incoming` 目录。

利用 UUCP 传输文件时，最大的问题是它只接收那些发到人人可写的目录中的文件。这将导致有的用户为其他用户设计一些圈套，或捉弄他们。但是，这个问题是无法避免的，除非你不用 UUCP 文件传输。

### 11.4.3 文件转发

UUCP 提供了一种机制，可令其他系统在你这一端执行文件传输。例如，允许你使 `seci` 为自己取得 `uchile` 上的文件，并将它发送到自己的系统中。

这种通过多个系统的传输机制就叫作“转发”。上面的例子中，使用转发机制的原因可能是 `seci` 能够通过 UUCP 访问 `uchile`，你的主机却不能。但是，如果你的主机也运行 UUCP 系统，可能想将转发服务限定在自己信得过的少数几台主机上，以免自己为了下载最新的 X11R6 源代码不得不支付巨额的电话账单。

默认情况下，泰勒式 UUCP 完全不允许转发。要针对特定的系统启用转发机制，可以采用 `forward` 命令。这个命令指定一系列站点，系统要求你将转发作业交给这些站点去完成。

`uchile` 的 `forward-to` 条目是非常必要的，因为它返回的任何一个文件事实上都会传递到 Pablo。如果没有这个条目，UUCP 就会将这些文件丢弃。这个条目采用的是 `forward` 命令的变体，允许 `uchile` 通过 `seci`，只把文件发给 Pablo；不得发向别的系统。

如果想把文件转发到所有的系统，可采用一个特殊的关键字 `ANY`（必须采用大写）。

## 11.5 如何设置拨入

如果你想将自己的站点设置为拨入，必须允许用户登录到你的串行端口，并自定义一些系统文件，提供 UUCP 账号。这就是我们这一小节的主题。

### 11.5.1 设置 `getty`

如果打算将串行线路用作拨入端口，就必须在这个端口上启用一个 `getty` 进程。但是，有

些getty实施的确不适合拨入，因为人们通常想采用同一个串行端口进行拨入和拨出。因此，必须保证你使用的getty进程能够和其他程序（比如uucico和minicom）共享同一条线路。具有此功能的一个程序是getty\_ps包内的uugetty。许多软件中都有这个程序，在/sbin目录下，可找到它。我所知道的另一个程序是mgetty，它是Gert Doering编写的，同样具有这一功能。可从sunsite.unc.edu获得其最新版本。

至于uugetty和mgetty在控制登录方面的不同，不在本书讨论之列。有兴趣的读者，可参考Greg Hankins编写得《Serial HOWTO》，以及getty\_ps和mgetty自带的文档。

### 11.5.2 提供UUCP账号

接下来，必须设置的是用户账号，有了账号，远程站点才能登录到你的系统，建立UUCP连接。一般说来，要为每个拨入系统提供一个独立的登录名。在为Pablo系统设置账号时，可将Upablo作为它的用户名。

对于通过串行端口拨入的系统，通常情况下，你都必须将其账号添加到系统密码文件/etc/passwd中。最好把所有的UUCP登录信息统统归为一个特殊的组，比如uuguest。系统账号的根目录应该设为公用假脱机目录/var/spool/uucppublic；其登录外壳脚本必须是uucico。

如果系统中已经安装了影子密码套件，就可以用useradd命令来设置账号：

```
# useradd -d /var/spool/uucppublic -G uuguest -s /usr/lib/uucp/uucic
```

如果没有安装影子密码套件，则需要手工编辑/etc/passwd。即增加下面这一行，其中5000和150是数字化的uid和gid，分别分配给用户Upablo和组uuguest。

```
Upablo;x:5000:150:UUCP Account:/var/spool/uucppublic:/usr/lib/uucp/u
```

安装账号之后，必须激活它。也就是说利用passwd命令，将其设置为密码。

针对通过TCP连接到你的站点上的UUCP系统，如果要为它们提供服务，必须设置inetd，令其处理进入uucp端口的连接。在/etc/inetd.conf内增加下面这一行即可：

```
uucp stream tcp nowait root /usr/sbin/tcpd /usr/lib/uucp/uuc
```

注意 通常情况下，tcpd有700模式，所以你必须以root用户的身份来调用它，这一点和uucp是不同的。

-l选项令uucico执行其本身的登录身份验证。uucico将像标准的登录程序一样，提示登录名和密码，但依据是自己的私用密码数据库，而不是/etc/passwd。这个私用密码数据库文件名为/usr/lib/uucp/passwd，其中包含成对的登录名和密码，比如：

```
Upablo IslaNegra  
Ulorca co ' rdoaba
```

当然，这个文件的拥有者必须是uucp，其保护模式为600。

如果你认为这个数据库不错的话，还可将其用于普通的串行登录。

首先，你需要泰勒式UUCP 1.05，因为它允许getty利用-u选项，将拨号用户的登录名传递给uucico（1.04版本中也有-u选项，但它只是一个no-op）。然后，必须让你正在使用的getty，调用uucico，而不是常用的/bin/login。有了getty\_ps，就可以通过在配置文件内设置LOGIN选项来达到这一目的。但是，与此同时，也取消了交互式登录对话。另一方面，mgetty有一个讨人喜欢的特性，允许你根据用户提供的登录名，调用不同的登录命令。比如，你可以要求

mgetty针对特定的所有用户（他们提供的登录名统统以大写的 U 开头）使用 uucico，而个别用户则采用标准的登录命令。

要保护自己的UUCP用户，防止恶意呼叫者指定一个假系统名，中途拦截那些用户的所有邮件，你应该在 sys 文件内的每一条系统条目中，增加 called-login 命令。

### 11.5.3 预防措施

关于UUCP，最大的问题之一是呼叫系统可以采用假名；它在登录之后，向被呼叫系统声明自己的名字。因此，攻击者可能登录到他/她自己的 UUCP 账号，假装某某人，中途截取其他站点的邮件。如果你提供通过匿名 UUCP 的登录，这个问题就严重了，因为，这样的话，登录密码几乎是公开的。

除非你知道自己信得过呼叫自己的那些站点，不然，你必须采取适当的措施，来防范那些欺骗者。具体的做法是在 sys 中指定一个 called-login，要求每个系统都使用一个特定的登录名。

这样做的结果就是，一旦某个系统登录进入，并伪称自己是 Pablo，那么 uucico 就会查实它是否曾以 Upablo 的身份登录过。假如没有，那么呼叫的系统就会被拒绝，并取消此次连接。大家应养成为增加到 sys 文件的每个系统条目，增加一个 called-login 命令的习惯——无论它们是否曾经呼叫过你的站点。对那些从未呼叫过你的站点，或许也应将 called-login 设为某些明显是伪造的用户名，比如 neverlogsin 等等。

### 11.5.4 呼叫序列号检查

要想防范网络骗子，有效地探测那些虚假的登录，另一个办法是利用“呼叫序列号检查”。利用呼叫序列号检查，可防范非法闯入者找出我们登录进入 UUCP 系统时使用的密码。

在使用呼叫序列号检查的情况下，双方的机器都会追踪并维护着迄今为止建立过的连接数量。每建立一次连接，这个序列号都会递增 1。登录进入后，呼叫者需发出它的呼叫序列号。同时，被呼叫者需要将其与自己的号码核对。假如两者不同，连接请求便会被拒绝。因此，假如初始的编号是随机选定的，那么攻击者很难猜出正确的呼叫序列号是多少。

除此以外，呼叫序列号还能做更多事情：即使某些“异常聪明”的人能侦测出你的呼叫序列号，同时找到你使用的密码，我们也能及时地发现这一情况！一旦攻击者调用了你的 UUCP feed，并偷走你的邮件，那么 feeds 调用序列号就会增 1。我们下一次调用自己的 feed，并试图登录时，远程的 uucico 就会拒绝登录，因为数字（编号）已不再相符了！

假如启用了呼叫序列检查功能，便应定期检查自己的日志文件，找出那些可能是由于外来攻击而造成的错误消息。假如你的系统拒绝由呼叫系统提供的一个序列号，uucico 就会在日志文件中记录一条错误消息，类似于“Out of sequence call rejected”（序列调用被拒绝）。假如由于序列号不同步，而造成 feed 拒绝你的系统登录，它就会在日志文件中写入一条错误消息，注明“Handshake failed(RBADSEQ)”（握手失败）。

接下来，我们必须实际地创建文件，文件中包括序列号本身。Taylor UUCP 将序列号保存在一个名为 .Sequence 的文件中。该文件放在远程站点的 spool 目录下。注意该文件必须由 uucp 所有，而且必须采用模式 600（亦即只能由 uucp 读写）。最好用一个任意的值对这个文件进行初始化。否则的话，攻击者就有可能轻松地猜出序列号，并用小于它的所有值进行尝试（比

如小于60)。

当然，远程站点也必须允许呼叫序列检查。而且在最开始的时候，采用和我们指定的完全一样的序列号。

### 11.5.5 匿名UUCP

如果要为系统提供匿名 UUCP 访问权限，首先必须设置一个特殊的账号（原因如前所述）。一种常见的做法是将它的登录名和密码都设为 `uucp`。

除此以外，必须针对那些未知的系统，设定几个安全防护选项。举个例子来说，或许应禁止它们在我们的系统上执行任何命令。但是，我们不能在 `sys` 文件条目中设置这些参数。这是由于 `system` 命令要求用到系统的名字，但这个名字是我们所没有的。Taylor UUCP 解决这个问题的办法是使用 `unknown` 命令。可在 `config` 文件中使用该命令，指定那些通常可在一个系统条目中出现的任何命令。

这样一来，我们就可以限制未知的系统从 `pub` 目录中下载任何文件，并能禁止它们将文件上载到 `/var/spool/uucppublic` 目录下的 `incoming` 目录中。下一行可令 `uucico` 忽略来自远程系统的、从而打开本地调试功能的任何请求。最后两行则允许未知的系统执行 `rmail`；但是，指定的命令路径将令 `uucico` 只在一个名为 `anon-bin` 的私用目录内查找 `rmail` 命令。这样一来，你就可以提供某一特殊的 `rmail` 命令，将所有邮件转发到超级用户那里进行检查。这样便允许匿名用户抵达系统维护人员那里，与此同时，还防止他们将邮件投入其他站点。

要启用匿名 UUCP，必须在 `config` 文件内至少指定一条未知语句。否则，`uucico` 就会拒绝任何一个未知系统。

## 11.6 UUCP低级协议

为了与远程终端协商会话控制和文件传输，`uucico` 采用了一个标准化的消息集。这就是通常所说的“高级协议”。在初始化阶段和挂断阶段，这些消息被作为一系列字串得以传输。但是，在真正的传输阶段，采用的就是低级协议，这些低级协议对高级协议来说，大部分是透明的。这样很方便进行错误检查，特别适用于使用不太可靠的传输线路的情况。

### 11.6.1 协议概述

由于 UUCP 使用在不同类型的连接上，比如串行线路和 TCP，甚至于 X.25，所以需要采用某些特殊的低级协议。另外，几个 UUCP 实施方案已介绍了不同的协议。

协议大致可分为两类：流式传输协议和面向包的传输协议。近来的协议通过计算校验和的方式，将一个文件作为一个整体进行传输。这几乎没有任何开销，但要求有一条非常可靠的连接，因为任何一个错误都可能导致整个文件被重新传输。这些协议常用于 TCP 连接，但不适于电话线连接。尽管如今的 modem 的确能够及时纠错，但仍然不够理想，不能对你的计算机和 modem 之间存在的错误进行侦测。

另一方面，面向包的传输协议则是将文件分为若干个相同大小的数据包。每个包单独进行收发，单独计算校验和，并向发送方返回收到确认。为了使这类协议更为有效，人们还发明了滑动窗口协议，极大地减少了 `uucico` 在文件传输过程中，必须等候的时间。与流式传输协议相比，包传输协议的开销较大，所以它不太适合于 TCP 连接。

数据路径的宽度也有所不同。有时，根本不可能通过串行连接发送 8 位字符。比如，如果连接通过一个反应迟钝的终端服务器，就会出现这种情况。此时，传输过程中，带有 8 位字符集的字符必须用引号表示出来。在通过 7 位连接传输 8 位字符时，这种情况被视为是最糟糕的；因为它增加了一倍的数据传输量，尽管硬件可以对这些数据进行压缩。可传输任意 8 位字符的线路叫作 8-bit clean。所有 TCP 连接和大多数 modem 连接都属于这种情况。

### 11.6.2 传输协议的调节

所有协议都允许对数据包的大小、超时等进行调整。通常情况下，默认设置能够保证数据包得以正常传输，但对具体情况而言，可能不是最好的选择。比如，g 协议采用的窗口大小是 1 到 7 之间，包的大小则是从 64 到 4 096 之间的 2 的乘幂个字节（包含在开发软件中的多数二进制都默认采用窗口大小在 7 到 128 字节之间的包）。如果你使用的电话线路噪音很大以至于丢包率高达 5%，那你就应该减少包的大小，缩小窗口的大小。另一方面，如果电话线路较好，每个 128 字节的包都发送一个 ACK（收到确认），又太浪费开销了，所以可以将包的大小增大至 512，甚至 1 024 个字节。

泰勒式 UUCP 提供了一种机制，可满足你的不同需求，即利用 `sys` 文件内的 `protocol-parameter` 命令，对传输参数进行调节。

不同协议的可调节参数和参数名是不相同的。要想得到一份相关的完整列表，请参考泰勒式 UUCP 源代码中包含的文档信息。

### 11.6.3 如何选定特殊协议

并不是每个 `uucico` 实施方案都能分辨和识别出每个协议，所以在起初的握手阶段，两方的进程都必须同意采用一个常见的协议。主管 `uucico` 向其从属 `uucico` 发送 `Protlist`，为其提供一份已获支持的协议列表，从属再从其中挑选一个协议。

根据使用端口的类型（`modem`、`TCP` 或直接连接），`uucico` 将组建一个默认的协议列表。对 `modem` 和直接连接来说，这份列表通常包含 `i`、`a`、`g`、`G` 和 `j`。对 `TCP` 连接来说，则包含 `t`、`e`、`i`、`a`、`g`、`G`、`j` 和 `f`。你也可利用 `protocols` 命令，改写默认的协议列表，该协议命令可在系统条目和端口条目中指定。

协议列表要求所以输入或输出的连接都要通过这个端口，采用 `i`、`g` 或 `G` 协议。如果远程终端不支持这些协议，连接对话就会失败。

## 11.7 故障排除

本小节将对 UUCP 连接可能出现的故障进行描述，并建议从哪里着手查错。但是，这方面的问题太多了，无法一一列举，本文将着眼于一些比较常见的重要问题。

任何情况下，利用 `-xall` 启用调试，都可在假脱机目录中的 `Debug` 内查看调试输出。它将有 助于你快速识别问题所在。同时，我还发现在 `modem` 没有连接时，它还可帮助我打开 `modem`。对采用贺氏标准的 `modem` 来说，具体作法是在拨号文件内的 `modem` 对话内，增加 `ATL1M1OK`。

第一步通常是检查所有的文件访问许可是否设置正确。`uucico` 应该被 `setuid` 为 `uucp`，而 `/usr/lib/uucp/`、`/var/spool/uucp` 和 `var/spool/uucppublic` 内的所有文件都应该属 `uucp` 拥有。另外，

假脱机目录内还有一些隐藏文件（也就是说，其文件名以“.”开头的文件；此类文件一般不能通过ls命令显示出来），同样也必须归uucp所有。

uucico总是说“Wrong time to call”(呼叫时间有误)：通常意味着sys文件内的系统条目中，没有指定time命令，该命令详细规定了呼叫远程系统的时间，或目前禁止呼叫你指定的系统。如果不指定呼叫时间表，uucico就会假定从不呼叫该系统。

uucico总是抱怨站点被锁：这通常意味着uucico侦测到远程系统的/var/spool/uucp内有一个锁文件。该锁文件可能源于上一次呼叫的已崩溃或已被杀死的系统。但是，更可能存在另一个uucico进程，该进程准备呼叫远程系统，但又凝在对话脚本中。如果这个uucico进程未能成功连接到远程系统，就应该用挂断信号将其杀死，并删除所有的锁文件。

我能够连接到远程站点，但对话脚本却不能：查看你从远程站点收到的文本。如果它含混不清，可能就是速度方面的问题。否则，就确定它真的与你的对话脚本所希望的行为保持一致。记住，对话脚本的开头是一个expect字串。在你收到登录提示时，就应该发送自己的用户名，但绝不可能在发送密码之前得到密码提示，插入延时。可能是因为你的modem太慢了缘故。

我的modem不能拨号：uucico在拨出时，如果你的modem没有指明DTR线路已被激活，可能是因为你没有为uucico指定一个恰当的设备。如果你的modem识别出了DTR，就会利用你可以写入的终端程序进行验证。如果这样做行得通，就在modem对话开始之即，利用E打开应答。如果在modem对话期间，它没有对你的命令作出应答，就查看自己的modem，线路速率是否设置过高或过低。如果有了应答，就看你是已经取消modem响应，还是将它设置为数字代码。最后再验证对话脚本本身是否正确。记住，你必须写入两个后斜杠，并将其中之一发给modem。

我的modem在拨号时，总不能成功：在电话号码中插入延时。这样做是非常有用的，特别是通过公司内部电话网拨出时。

日志文件说我的包丢失率非常之高：这看来是速度问题。计算机和modem之间的连接是不是太慢（记住将其速率调整为最高）？或你的硬件速度太慢，以至于不能及时提供服务？如果你的串行端口采用的是NSC-16550A芯片，38Kbps这个速率是比较理想的；但是，如果没有FIFO（比如16450芯片），9600bps就是极限了。另外，你还应该保证串行线路上已经启用硬件握手。

另一个可能存在的问题是端口上未启用硬件握手。泰勒式UUCP 1.04没有打开硬件握手这一规定。

我能够登录，但不能握手：产生此类故障的原因很多。日志文件中的输出应该能够给出一大堆原因。看看远程站点提供的协议是什么（在握手期间，它发送了一个Pprotlist字串）。但也许日志文件内什么东西也没有（如果你没有在sys或端口文件内选定协议的话）。

如果远程系统发回RLCK，说明远程系统上有一个很早以前的锁文件。如果因为你已经通过另外的线路连接到远程系统，它没有与你握手的话，就可要求将它删除。

如果它发回RBADSEQ，另一个站点就会为你启用对话计数检查，但数字不符。如果它发回RLOGIN，就说明你不能通过这个ID进行登录。

## 11.8 日志文件

在编辑UUCP套件，使其能够采用泰勒式登录时，只有三个日志文件，这三个日志文件都



驻留在假脱机目录内。主要的日志文件名为 Log，与已建立连接和已传输文件相关的所有信息统统包含在这个文件内。

下一个比较重要的日志文件是 Stats，它列出了文件传输特性。

第三个日志文件便是 Debug。该文件内主要保存一些调试信息。使用调试时，应该保证这个文件的保护模式是 600。根据你所选定的调试模式，该文件中可能包含用于连接远程系统的登录和密码信息。

已发布的 Linux 产品中，有的 UUCP 二进制文件已被编辑为采用 HDB 式记录。HDB UUCP 采用了大量的日志文件，这些文件保存在 /var/spool/uucp/.Log 目录下。这个目录中包含另外三个子目录，名为 uucico、uuxqt 和 uux。这三个子目录中包含每条相应命令生成的记录输出，而且这些输出被保存在各个站点的不同文件内。因此，在拨叫 pablo 站点时，uucico 的输出将记入 .Log/uucico/pablo 文件，而后来 uuxqt 的运行结果将被写入 .Log.uuxqt/pablo。但是，写入不同日志文件的各行输出和泰勒式记录的一样。

在你启用 HDB 式记录的调试输出时，它将被写入 /var/spool/uucp 目录下的 .Admin 目录。在拨出期间，调试信息也会被发送到 .Admin/audit.local 中，在有人拨入时，uucico 的输出结果将被记入 .Admin/audit。