

第18章 新闻阅读机的配置

新闻阅读机的目的是提供用户功能，允许他 / 她更轻松地访问新闻系统功能，比如以一种更为舒适的方式投递文章，或浏览新闻组内的内容等。这个界面的才能是无止境的。

目前，已经有两个新闻阅读机被移植到了 Linux。下面将描述三个常见新闻阅读机的基本设置，它们是：tin、trn和nn。

最有效的新闻阅读机之一是

```
$ find /var/spool/news -name '[0-9]*' -exec  
cat {} \; | more
```

这是顽固分子阅读新闻的方式。

但是，大多数新闻阅读机都是相当复杂的。它们通常提供全屏界面，这个界面分为若干个单独的层。一层用于显示用户已经订阅的所有新闻组，一层用于显示一个新闻组内的所有文章概述，一层用于显示每篇单独的文章。

在新闻组层，许多新闻阅读机都显示出一个文章列表，展示它们的主题行和作者。在大型的新闻组内，对用户来说，要了解所有相关文章是不可能的，虽然对早期文章的答复进行鉴定是可能做到的。

答复通常重复原文的主题，做法是在原文章前加一个“ Re: ”。另外，对一个直接重复出现的文章来说，其消息 ID可能要在Reference:header这一行指定。按照这两点对文章进行分类，将生成较小的文章簇（事实上，是树），这些簇称为“主线”或“连载”（thread）。编写新闻阅读机的任务之一是设计一种更有效的连载方案。

这里，我们打算深入讨论如何建立用户界面。目前，Linux系统上可用的所有新闻阅读机都有相关的帮助文档，供大家参考。

下面，我们只讨论一些管理任务。大部分与建立连载数据库和创建账户有关。

18.1 tin配置

与连载有关的、功能最齐全的新闻阅读机是 tin。它是Iain Lea编写的，利用一个名为 tass 的老式新闻阅读机作为原型。后者是 Rich Skrenta编写的。用户进入新闻组之后，tin才开始执行连载，它的速度相当快，但通过 NNTP连接时，速度不太理想。

在一台486DX50机器上，直接从磁盘上读取时，它只花30秒，就可连载1 000篇文章。但通过与一个加载新闻服务器的 NNTP连接时，则要花5分钟。

注意 如果由NNTP服务器自己执行连载处理，让客户机获取连载数据库，能够显著改进速度；比如INN-1.4版本，就是这样的。

也可以利用 -u选项，或随 -U选项一起，调用 tin，定期更新自己的索引文件，从而对此进行改进。

通常，tin把自己的连载数据库放在 .tin/index 下面的用户根目录中。但是，这样可能会占

用系统资源，所以应该考虑把每个数据库的副本集中放在某个地方。具体作法是：令 `tin` 的 `setuid` 为 `news`，或某个完全没有特权的账户（但是，千万不要将其设为 `anybody`。通常，无论什么命令和文件都不应该和这个用户有任何关联）。

然后，`tin` 将所有的连载数据库保存在 `/var/spool/news/.index` 下。对任何文件访问和脚本的换码，它都会把自己的有效 `uid` 重新设为调用它的用户之实际 `uid`（这便是你作为超级用户调用它时会出现不详错误消息的原因。之后，无论如何，你都不能以 `root` 的身份执行任何操作了）。

最好的解决之道是安装 `tind` 索引后台程序，它作为后台程序运行，并定期更新索引文件。但是，这个后台程序没有包含在 `tin` 内，所以你必须自行编辑它。如果你运行的局域网带有一个中心新闻服务器，就可以在该服务器上运行 `tind`，让所有的客户机通过 `NNTP`，获取索引文件。当然，这需要对 `NNTP` 进行扩展。实施这一扩展的 `nntpd` 补丁程序包含在 `tin` 源代码内。

有些已发布的程序内包含的 `tin` 版本没有编入对 `NNTP` 的支持，但现在，大多数程序内都有了。在作为 `rtin` 或随 `-r` 选项调用时，`tin` 试着与 `NNTP` 服务器建立连接，这个服务器是 `/etc/nntpserver` 或 `NNTPSERVER` 环境参数中指定的。`nntpserver` 文件中用单独的一行来包含该服务器的名称。

18.2 trn配置

与早期的新闻阅读机（即 `rn`，意为阅读新闻）相比，`trn` 是个当然的赢家。其名字中的“`t`”代表“连载”（`threaded`）。它是 Wayne Davidson 编写的。

与 `tin` 不一样，`trn` 不能在运行时生成其连载式数据库。相反地，它采用的是由一个名为 `mthreads` 的程序提供的文件，该程序必须通过 `cron` 定期调用，以便更新索引文件。

但是，不运行 `mthreads`，并不意味着你不能访问新闻组文章；它只说明你的文章选择菜单中，将散布着所有“`Novell buys out Linux!!`”文章，而不是你可轻易跳过的单一主题。

要为特定新闻组打开连载播出，就要在命令行上随一个新闻组列表调用 `mthreads`。这个列表的结构和 `sys` 文件内的完全一样：

```
mthreads comp.rec.!rec.games.go
```

这样，将针对 `comp` 和 `rec` 内的所有文章进行连载处理，`rec.games.go` 除外（玩 `Go` 的人通常都不需要新奇的主题）。之后，根本不用任何选项，调用它，就会令其将所有新近抵达的文章按照主题罗列在一起。`active` 文件内所有新闻组的连载处理也是可以打开的，具体做法是随一个 `all` 新闻组列表调用 `mthreads`。

如果你是在夜间接收新闻，可自行定义在每天早上运行 `mthreads`，但如果需要，还可以更频繁地运行它。对通信量大的站点来说，它们可能希望在后台程序模式下运行 `mthreads`。利用 `-d` 选项，在系统启动时启用它时，它会把自己置入后台运行，每隔 10 分钟，就出来检查是否有新文章到达，如果有，就对它们进行连载处理。要在后台程序模式下运行 `mthreads`，须把下面这一行放入你的 `rc.news` 脚本内：

```
/usr/local/bin/rn/mthreads -deav
```

`-a` 选项令 `mthreads` 在新闻组建立时，自动打开对这些新闻组的连载处理；`-v` 选项为 `mthreads` 文件启用冗长的记录消息，该文件名为 `mt.log`，位于你安装的 `trn` 目录中。

不再有用的旧文章必须定期从索引文件内删除。默认情况下，只有其编号在最低编号以

下的文章才会被删除。

注意 C-News不会自动更新这个最低编号；所以只有运行updatemin。详情参见第16章。

对已经被过期处理的、最低编号以上的文章（因为最旧的文章已经被 Expires:header字段分配了一个很长的过期日期）来说，也是可以删除的，具体做法是为 mthreads加上-e选项，实施一个“增强型”的过期运行。mthreads在后台模式运行时，-e选项会令其在午夜之后，进入增强型过期运行，一天一次。

18.3 nn配置

nn是Kim F.Storm编写的，它被认为是一个新闻阅读机，其最终目的是不阅读新闻。其名字代表的是“没有新闻”（即no news），其座右铭是“没有新闻，便是好事。nn更好。”

为了实现这一抱负，nn为我们带来了名目繁多、功能齐全的维护工具，它们不仅允许你生成新闻主题，还扩展到检查主题数据库的一致性、清算和收集特性以及访问限制等。另外，还有一个管理程序，名为nnadmin，它允许你交互执行这些任务。这个程序非常直观易懂，所以我们不打算对它进行详细讨论，只说一下如何生成索引文件。

nn主题数据库管理程序称为nnmaster。它通常作为后台程序运行，从rc.news或rc.inet2脚本开始。它的调用形式如下：

```
/usr/local/lib/nn/nnmaster -1 -r -c
```

如此这般，就为active文件内的所有新闻组启用了主题归类处理。

同样地，也可周期性地从cron调用nnmaster，具体作法是为其指定一个新闻组列表，作为其调用依据。这个列表和sys文件内的订阅列表极为类似，不同的是它采用的是空格，后者采用的则是逗号。它没有采用假的新闻组名all，而是用一个空白参数来代表所有的新闻组。其示范调用如下：

```
# /usr/local/lib/nn/nnmaster !rec.games.go rec comp
```

注意，这里的顺序有特定的含义：最左面的组定义总是有效的。所以，如果我们把!rec.games.go放在rec之后，这个组内的所有文章都会被忽略。

nn提供了若干种方法，从其数据库内删除过期文章。其一是：查看新闻组目录，丢弃失效文章对应的条目，从而更新数据库。这是一个默认操作，是通过调用带有-E选项的nnmaster来执行的。它的速度相当快，通过NNTP进行此操作时除外。

第2种方法和默认的mthreads过期处理运行极为相似，它只删除其编号低于最低编号的文章之对应条目，最低编号在active文件内。利用-e选项，可启用它。

最后是第3种方法，丢弃整个数据库，重新收集所有文章。具体做法是为nnmaster指定-E3选项。

对即将过期的新闻组列表来说，它们是由-F选项采用上面的方式指定的。但是，如果你把nnmaster作为后台程序运行，必须在它过期之前，利用-k杀死它，随后再利用原来的选项重新启动它。所以在所有新闻组上运行过期处理的正确命令是：

```
## nnmaster -kf ""  
# nnmaster -lrc
```

用于优化nn行为的标记还有许多。如果还想知道如何删除不好的文章或摘取文章提要，

可参考nnmaster手册。

nnmaster依赖于一个名为GROUPS的文件，这个文件位于/usr/local/lib/nn内。如果最初没有这个文件，就会创建一个。对于每个新闻组，它都包含一行，以新闻组名开头，后面选择性地跟一个时间戳和若干个标记。这些标记是可以编辑的，用于针对拿不准的新闻组启用特定行为，但你不能更改新闻组出现的顺序。（这是因为它们的顺序必须和二进制MASTER文件内的条目顺序一致。）允许设置的标记及其作用的有关详情也包含在nnmaster手册内。