Linux Fido-Point HOWTO

Roland Rosenfeld (`roland@spinnaker.rhein.de, 2:2450/42`) v3.22-2, 17. Dezember 1996

Dieses File soll helfen, einen Fido-Point unter Linux zu installieren, wobei ifcico als Mailer und FidoGate als Gatewaysoftware verwendet werden.

Inhaltsverzeichnis

1 Aktuelle Version dieser HOWTO

Falls Du eine etwas veraltete Version dieser HOWTO haben solltest, kannst Du Dir die immer aktuellste Version bei mir (2:2450/42 ZYX und ISDNC/X75) unter dem Namen `FIDOPNT.GZ` requesten (Keine Mailbox, nur File-Requests!). Weiterhin ist sie per WWW unter `http://www.rhein.de/˜roland/FidoPnt/` und per FTP unter `ftp.rhein.de:/pub/unix/fido/` verfügbar. Außerdem ist sie natürlich auf der Homepage des DLHP unter `http://www.tu-harburg.de/dlhp/` zu finden.

Eine englische Übersetzung von Oliver Much ist im Fido unter dem Namen `FIDOPNTE.GZ` sowie im Internet unter `http://www.rhein.de/˜roland/FidoPnt-en/` bzw. `ftp.rhein.de:/pub/linux/fido/FidoPnt-en.*` verfügbar.

2 Benötigte Programme

2.1 FidoGate

Dies ist die Gateway-Software, welche Fido-Messages in Usenet-News/Mail konvertiert. Die aktuelle Release ist Version 3.9.7.

Diese Aussage stimmt so nicht mehr ganz. Aktuell ist Version 4.1.3, bei der es allerlei neue, aber für den Pointbetrieb nicht unbedingt notwendige, Features gibt. Eine komplett überarbeitete HOWTO, bei der auch der Tosser von Fido-Gate verwendet wird, ist geplant, wird aber leider wegen Zeitmangels noch auf sich warten lassen. Bis dahin empfehle ich 3.9.7 gemäß dieser HOWTO zu verwenden oder 4.1.3 selber gemäß der Orginal-Doku zu verwenden.

Das Programm wird von Martin Junius (`mj@fido.de`, 2:2452/110) entwickelt.

Die aktuellste Version findet man immer in `ftp.fido.de:/pub/fidogate/` und gewöhnlich etwas später auch auf `sunsite.unc.edu:/pub/Linux/system/Fido/`. Weiterhin kann man das Programm auch bei Martin requesten (2:2452/110 HST, 2:2452/111 ISDN). Für weitere Informationen kann auch ein Blick auf die FidoGate-Webpage unter `http://www.fido.de/fidogate/` nicht schaden.

2.2 ifcico

Dies ist ein Mailer ähnlich Binkley-Term, der allerdings mehr Unix-like ist, d.h. er wartet nicht aktive (unter Dos stört das ja nicht) und orientiert sich auch ansonsten an uucico von UUCP. ifcico ist im Paket IFmail enthalten. Die aktuelle Version ist 2.8g. Die alte 2.8d sollte man nicht verwenden, da diese einen häßlichen Bug beinhaltet.

Das Programm wird von Eugene Crosser (`crosser@average.org`, 2:5020/230) entwickelt.

IFmail enthält neben ifcico auch eine eigene Gateway-Software, aber ich favourisiere stattdessen FidoGate und beschreibe hier auch nur die Konfiguration mit FidoGate.

Die aktuellste Version wird regelmäßig von Eugene nach `tsx-11.mit.edu:/pub/linux/sources/usr.bin/` und `sunsite.unc.edu:/pub/Linux/system/Fido/` hochgeladen. Weitere Informationen findet man auf der IFmail-Webpage unter `http://www.average.org/ifmail/`.

2.3 News-Transport-Agent (NTA)

Als News-Transport-Agent kann man sowohl cnews als auch INN einsetzen. Um sich für das passende Programm entscheiden zu können, hier einige Vor-/Nachteile beider Programme:

- cnews besteht zu großen Teilen aus Shell-Skripts, die sehr schlecht zu debuggen sind, und auch recht langsam arbeiten
- cnews kann von Natur aus kein NNTP (Online-News-Protokoll). Will man trotzdem NNTP verwenden (z.B. für diverse Newsreader, die nur per NNTP mit dem Newssystem kommunizieren können), braucht man einen nntpd, welcher recht ungeschickt zu konfigurieren ist und auch nicht alle Möglichkeiten von NNTP unterstützt.
- INN ist für NNTP ausgelegt. Daher läuft auch ständig ein Dämon (innd), der auch dafür sorgt, daß die lokal geschriebenen News-Artikel sofort für alle lesbar sind. Bei cnews werden die neuen Artikel nur zu bestimmten Zeiten (z.B. alle 10 Minuten) ins Newssystem übernommen
- cnews macht Probleme, sobald auf `/var/spool/news` der Platz eng wird (weniger als 10MB frei).

Ich persönlich habe früher mit cnews gearbeitet, aber inzwischen auf INN umgestellt, da mir INN insgesamt rr-underërscheint. Außerdem ließ sich INN bei mir ohne Probleme auf Anhieb installieren, was ich von cnews nicht behaupten kann.

2.3.1 cnews

Da bei älteren Slackware-Distributionen das cnews extrem verkrüppelt war (die Manpages fehlten beispielsweise ganz), sollte man mindestens das cnews aus Slackware 2.1 verwenden. Alternativ kann man sich, eine komplette Source-Distribution zu besorgen und selber compilieren. Ich habe hier die Performance-Release vom 20.2.93 auf `ftp.uu.net:/news/` verwendet. Inzwischen gibt es auch eine neuere Version, aber die habe ich noch nicht ausprobiert. Ich werde mich also hier immer auf die alte Version beziehen und dieser Teil meiner Dokumentation wird auch in Zukunft nicht weiterentwickelt.

2.3.2 INN

In der aktuellen Slackware-Distribution sind sowohl cnews als auch INN enthalten, so daß man die freie Auswahl hat. Dieser INN sollte funktionieren, allerdings bevorzuge ich, meine Software selber zu compilieren und zu konfigurieren, daher habe ich Sourcen von INN-1.4sec verwendet, die man unter `sunsite.unc.edu:/pub/Linux/system/News/inn1.4sec-linux-src.tgz` findet.

2.4 Mail-Transport-Agent (MTA)

Hier hat man im wesentlichen die Wahl zwischen smail und Sendmail V8. smail ist das kleinere simplere Paket, welches leider einige kleinere Bugs hat, während Sendmail V8 auf den ersten Blick sehr unangenehm zu konfigurieren ist. Allerdings gibt es inzwischen eine Konfiguration mit dem Macro-Prozessor M4, welche die Konfiguration enorm vereinfacht und wenn man seine anfängliche Scheu abgelegt hat, stellt man fest, daß die Konfiguration nicht komplizierter als die von smail ist. Allerdings ist Sendmail um einiges mächtiger als smail. Ich persönlich bin daher von smail auf Sendmail umgestiegen.

2.4.1 smail

Ich habe zuletzt die Version 3.1.29.1 (`sunsite.unc.edu:/pub/Linux/system/Mail/delivery/smail-linuxbin-3.1.29.1.tar.gz`) verwendet, da das alte 3.1.28.1 einen Bug in uuname-Driver hat (das ist allerdings nur für den UUCP-Betrieb von Belang). Die in dieser HOWTO erklärte Konfiguration setze ich allerdings nicht mehr ein, da ich auf Sendmail umgestellt habe.

2.4.2 Sendmail V8

Die aktuelle Version von Sendmail V8 ist 8.8.4. Da die meisten älteren Versionen aufgrund von bekannt gewordenen Sicherheislücken ersetzt wurden, empfiehlt es sich, immer die aktuellste Sendmail-Version einzusetzen. Die neuste Version von Sendmail findet man immer unter `FTP.CS.Berkeley.EDU:/ucb/src/sendmail/` oder `ftp.cert.dfn.de:/pub/tools/net/sendmail/`. Wichtig ist bei der Installation von Binaries, daß man auch die kompletten Konfigurationsfiles installiert, die sich beispielsweise bei Slackware unter dem verwirrenden Paketnamen `smailcfg.tgz` verbergen.

2.5 Newsreader

Hier kann man einen beliebigen Newsreader verwenden, z.B. tin, nn, trn, xvnews, xrn, slrn, knews, Emacs-GNUS,... Falls man cnews einsetzt, sollte der Newsreader allerdings einen UUCP-Modus besitzen, d.h. nicht ausschließlich mit NNTP arbeiten, da man ansonsten den nntpd zusätzlich installieren muß, was mangels Dokumentation sehr unhandlich ist. Ich empfehle für den Anfang tin, da tin einfach zu bedienen ist und auch wenig Probleme mit der Konfiguration auftreten.

2.6 Mailreader

Auch hier kann jeder verwenden was er will: elm, pine, mail, xmail, xmailtool, xfmail, Emacs-VM, Emacs-GNUS,...

2.7 Packer

Manche Packer finden sich schon in den Distributionen. Falls spezielle Packer fehlen, kann man sich diese auf diversen Servern besorgen, die wie `sunsite.unc.edu:/pub/Linux/utils/compress/` ein spezielles Packer-Verzeichnis besitzen.

ZIP

unzip51 entpackt auch die das neue ZIP-Format. Es gibt inzwischen auch ein zip, das wieder nach dem neuen Format packt.

ARJ

unarj241 entpackt ARJ-Archive, allerdings gibt es dazu unter Linux keinen Packer.

LZH

lha exisitert auch als Linux-Port.

ARC

Alter, aber wichtiger Packer, denn die meisten Nodelisten sind mit arc gepackt. Es existiert mindestens ein Linux-Port.

RAR

unrar101 entpackt RAR-Archive, einen passenden Packer habe ich noch nicht gesehen. Ich habe die Sourcen zu unrar unter `ftp.kiae.su:/.2/unix/arcers/unrar101.tgz` gefunden.

2.8 TIC-Prozessor

Es gibt einige kleinere TIC-Prozessoren, die in PERL geschrieben wurden. Insbesonere ist hier `tic010b.tgz` (teilweise auch unter dem Namen `lt010b.tgz` zu finden) von Cees de Groot (`cg@bofh.lake.de`, 241:10000/1512) zu nennen. Dieses Programm ist seit ifmail Version 2.8a im Verzeichnis `ifmail/misc/contrib/tic` enthalten.

Außerdem ist bei FidoGate ein Tic-Prozessor namens ftntick enthalten.

Abschließend ist hier noch das Paket FidoTools-0.9, zu finden unter `ns.aanet.ru:/vol1/nick/Linux/system/Fido/Fi` `0.9.tar.gz` zu nennen, welches im wesentlichen einen Tic-Prozessor enthält.

3 Weitere Dokumentation

News-HOWTO

zur Konfiguration von cnews

Mail-HOWTO

zur Konfiguration von mail

NET-2-HOWTO

zur Konfiguration des loopback-Netzes

Network-Admin-Guide

für die Konfiguration aller Netzwerk-Komponenten, sowie UUCP.

Weiterhin sollte man sich auf jeden Falle die FidoGate-Dokumentation aus dem Verzeichnis `fidogate/doc` durchlesen.

Für INN ist die inzwischen neuteilige FAQ unverzichtbar, da sie nicht nur praktisch alle auftretenden Fragen beantwortet, sondern außerdem noch wie eine HOWTO die Installation begleitet.

4 Meine Beispielkonfiguration

Ich werde hier als Beispiel meine eigene ehemalige Point-Konfiguration für die drei FTN-Netze Fido (Zone 2), Gernet (Zone 21) und Fido.de (Zone 242) angeben. Ich habe dabei folgende Adressen:

- 2:2450/111.13 (Fido)
- 21:100/64.13 (Gernet)

- 242:5000/4.13 (fido.de)

Ich polle alle drei Netze bei ein und dem selben Boss, aber bei verschiedenen Uplinks würde sich an der Konfiguration auch nicht viel ändern. fido.de bietet dabei einen Internet-Anschluß über das Gate von Martin Junius in Aachen. Ich bin aus dem Internet also als `roland@p13.flokiste.fido.de` erreichbar. Meine Internet-Mails schicke ich an `UUCP @ 242:4900/99`.

Bitte schickt mir an keine der obgengenannten Adressen Mails, da ich inzwischen ganz anders erreichbar bin (siehe am Anfang dieser HOWTO). Ich beschreibe in dieser HOWTO trotzdem die alte Konfiguration, da sie einige Details besser klärt.

Falls Du noch kein Internet-Gate kennst, solltest Du Dich dringend danach umsehen, denn FidoGate unterstützt selbiges ausgezeichnet und geht davon aus, daß Du entweder einen UUCP-Uplink oder wenigstens Connection zu einem Gate hast. (`schiele-ct.de`, `fido.sub.de` und `sesom.nbg.de` sind meines Wissens kostenlos nutzbar, siehe dazu aber auch die diversen Gateway-Listen in der Echomail-Area `GATEWAYS.GER` oder der Newsgroup `de.comm.gateways`).

Inzwischen habe ich für meinen Rechner auch einen direkten Internet-Zugang bei rhein.de gefunden, so daß mein Rechner auch als spinnaker.rhein.de (kurz: spi.rhein.de) im Internet erreichbar ist. Der Rechner rhein (er ist in der UUCP-Worldmap eingetragen, daher braucht er keine Domain, aber das ist inzwischen die Ausnahme) versorgt mich dabei mit Mail und News. Ich werde versuchen in dieser HOWTO die Konfiguration mit Internet via Gate (fido.de) und via UUCP (rhein.de) parallel zu beschreiben, so daß jeder seine Konfiguration wiederfinden sollte.

Und noch eine Bitte: Auch wenn ich überall in diesem File meine Adressen als Beispiel verwende, solltest Du auch für die ersten Versuche die Adressen und den Rechnernamen ändern, denn sonst bekomme ich die Antworten auf die Fragen, die Du unter meinem Namen stellst...

5 Zugriffsrechte

Die Zugriffsrechte der einzelnen Dateien sind ein recht heikles Thema bei der Kombination von FidoGate, ifcico und News. Daher solltest Du zunächst einmal nachsehen, ob die User news und uucp bei Dir existieren. Bei mir sieht das in `/etc/passwd` wie folgt aus:

```
uucp:sadfasdfasdf:5:5::/home/uucp:/bin/tcsh
news:aasdfasdfsda:24:24::/home/news:/bin/tcsh
```

Weiterhin müssen noch die Groups uucp und news existieren. Es hat sich als hilfreich erwiesen, den User news zusätzlich in die Group uucp und den User uucp zusätzlich in die Group news einzutragen. Aus meiner `/etc/group`:

```
uucp::5:uucp,news
news::24:news,uucp
```

Weiterhin sollten möglichst alle weiteren Spool-Dateien auch für die Group schreibbar sein, aber dazu später mehr.

6 Installation von FidoGate

Ich beziehe mich hier auf Version 3.9.7, bei älteren Versionen sind einige der Konfigurationsmöglichkeiten noch nicht vorhanden.

Zunächst ist jetzt `fidogate/config.h` zu editieren. Ich habe folgende Änderungen an der Orginal-Datei vorgenommen: (das ist nicht die komplette Datei, sondern nur die Zeilen die ich geändert habe!)

```
/* #define HOSTS_RESTRICTED */
#define SECURE
/* #define PASSTHRU_NETMAIL */
/* #define PASSTHRU_ECHOMAIL */
#define OUTDIR          "outbound"      /* Output of rfc2fido */

#ifdef SECURE           /* Secure permissions */
# define PACKET_MODE    0600            /* Mode for outbound packets */
# define BSY_MODE       0664            /* Mode for BSY files */
# define FLO_MODE       0664            /* Mode for FLO files */
# define DATA_MODE      0660            /* Mode for ffx data files */
# define DIR_MODE       0775            /* Mode for directories */
# define CONF_MODE      0664            /* Mode for written config files */
#else                   /* Open permissions */
# define PACKET_MODE    0666            /* Mode for outbound packets */
# define BSY_MODE       0666            /* Mode for BSY files */
# define FLO_MODE       0666            /* Mode for FLO files */
# define DATA_MODE      0666            /* Mode for ffx data files */
# define DIR_MODE       0777            /* Mode for directories */
# define CONF_MODE      0666            /* Mode for written config files */
#endif
```

Man sollte aus den Kommentaren von `config.h` erkennen, was das bedeuten soll. Es soll keine Beschränkung auf eingetragenen Hosts vorgenommen werden. Das Outbound-Directory heißt bei mir aus alter Gewohnheit `outbound` und nicht nur `out`. Man kann das problemlos umbenennen, muß nur insgesamt konsistent bleiben. Weiterhin habe ich die Permissions noch auf meinen Geschmack abgeändert.

Als nächstes sind noch einige Änderungen in `fidogate/config.make` vorzunehmen:

```
LIBDIR          = /usr/local/lib/fidogate
SPOOLDIR        = /var/spool/fnet
LOGDIR          = /var/log/fnet
OUTBOUND        = /var/spool/fnet/outbound
INBOUND         = /var/spool/fnet/inbound
PINBOUND        = /var/spool/fnet/pinbound
UUINBOUND       = /var/spool/fnet/uuin

OWNER           = uucp
GROUP           = uucp
DEBUG           = -O2 -s -fomit-frame-pointer
```

Im `fidogate/src/Makefile` unter install-dirs solltest Du noch `$(SPOOLDIR)/out` in `$(SPOOLDIR)/outbound` ändern.

Damit werden die Verzeichnisse an meine Verzeichnis-Struktur angepaßt und `uucp.uucp` wird User von FidoGate. Später wird auch `ifcico` unter dem User `uucp.uucp` arbeiten.

Jetzt mußt Du noch einige Änderungen an `fidogate/src/ftninpost.pl` vornehmen:

```
$RELAY    = "p13.flokiste.fido.de";
$PROTO    = "FIDOGATE";
$SENDMAIL = "/usr/lib/sendmail -oi -odq -oee -f%s -oMs$RELAY -oMr$PROTO -t";
$RNEWS    = "/usr/bin/rnews";
```

```
&do_cmd("$LIBDIR/ftninrecomb $options");
```

Hier wird der Name des Gates angepaßt (das ist eigentlich unwesentlich, der Name tritt allerdings später im Received:-Header der Mails auf). Wenn Du Sendmail/smail nicht als alle 15 Minuten aufrufen willst und auch nicht als Daemon startest, solltest Du die Queue-Option (`-odq`) von sendmail löschen, da die Mails ansonsten ewig in `/var/spool/smail/input` (bei smail) bzw. `/var/spool/mqueue` (bei Sendmail) liegen bleiben. Das `-oi` sorgt dafür, daß Mails, die eine Zeile mit einem Punkt und sonst nichts enthalten, nicht an dieser Stelle beendet werden. Seit FidoGate 3.9.5 ist das schon entsprechend eingebaut. Dann habe ich den Pfad von `rnews` noch an mein System angepaßt und letztendlich habe ich `ftninrecomb` eingeschaltet (im Orginal ist diese Zeile durch '##' auskommentiert). Diese Option hat zur Folge, daß Mails, die gemäß FTS-0047 gesplittet wurden, wieder zusammengeklebt werden.

Nun solltest Du als `root` in `fidogate/src` folgende Kommandos ausführen können:

```
make depend
make all
make install-dirs
make install
```

Nun wechselst Du nach `fidogate/lib` und startest dort

```
make install
```

Anschließend stehen alle benötigten Dateien in `/usr/local/lib/fidogate`. Jetzt mußt Du natürlich noch einige Anpassungen an Dein System vornehmen. Hier zunächst mal ein komplettes `/usr/local/lib/fidogate/config.common`, welches die globalen Konfigurationsdaten enthält:

```
#:ts=8
#
# /usr/local/lib/fidogate/config.common
#
# FIDOGATE config file common stuff,
# included by config.gate, config.main, config.ffx
#
# spinnaker.rhein.de
#
# Format:  keyword arg ...
#          keyword and args may be put in double quotes "..."
#

# Directories: lib, spool, BinkleyTerm-style outbound base dir (without
# the .../out.xxx), BinkleyTerm-style inbound dir
#
# lib, spool defaults are defined in config.h
#
LibDir          /usr/local/lib/fidogate
SpoolDir        /var/spool/fnet
LogDir          /var/log/fnet
Outbound        /var/spool/fnet
Inbound         /var/spool/fnet/inbound

#
```

```
# Internet address
#
#Hostname       p13
#Domain         .flokiste.fido.de
Hostname        spinnaker
Domain          .rhein.de
# Optional domain name for entries in HOSTS file
HostsDomain     .fido.de



#
# Zones and domains, the outbound directory is relative to the one
# specified with 'Outbound'.
#
#               zone    Inet domain     FTN domain      Outbound
#               ----    -----------     ----------      --------
Zone            default .fidonet.org    fidonet         -
Zone            1       .fido.sub.de    fidonet         outbound.001
Zone            2       .fido.sub.de    fidonet         outbound
Zone            3       .fido.sub.de    fidonet         outbound.003
Zone            4       .fido.sub.de    fidonet         outbound.004
Zone            5       .fido.sub.de    fidonet         outbound.005
Zone            6       .fido.sub.de    fidonet         outbound.006
Zone            21      .ger.sub.de     gernet          gernet
Zone            242     .fido.de        fidode          fidode
#
# OPTIONAL:
#
# MSDOS client drive to UNIX server directory translation. This allows
# FIDOGATE running on a UNIX system and BinkleyTerm on an MSDOS PC.
#
#DosDrive       H:      /home
#DosDrive       I:      /var/spool
#DosDrive       I:      /usr/spool
#DosDrive       P:      /u1
#DosDrive       Q:      /u2
```

Besonderes Augenmerk möchte ich dabei auf folgende Zeilen legen: Dies ist mein Rechnername in Internet:

```
Hostname        spinnaker
```

Dies ist meine Domain im Internet:

```
Domain          .rhein.de
```

Als ich nur über fido.de Internet-Zugang hatte, habe ich dort `p13` als Hostname und `.flokiste.fido.de` als Domain verwendet. Wenn Du keine Internet-Adresse hast, kannst Du notfalls auch einen beliebigen Namen als Hostname einsetzen und dann als Domain `.uucp` einsetzen (nicht den Punkt vergessen). Besonders wichtig ist, daß Du dann unbedingt darauf achtst, daß Du in `config.gate` `NoFromLine` definiert hast, da sonst Internet-Adressen generiert werden, auf die man nicht antworten kann. Das ist auch dann dringend anzuraten, wenn Du über ein Gate eine Internet-Adresse hast. Bei fido.de werden beispielsweiese News-Artikel gebounced, wenn diese eine FromLine enthalten, da es sich dabei um potentielle Dupes handelt.

`config.main` ist wie folgt anzupassen:

```
#:ts=8
#
# /usr/local/lib/fidogate/config.main
#
# FIDOGATE config file main AKA
#
# spinnaker.rhein.de
#
# Format:  keyword arg ...
#          keyword and args may be put in double quotes "..."
#

#
# Include common stuff (%L = LIBDIR)
#
include         %L/config.common

#
# FTN addresses
#
Address         2:2450/111.13
Address         21:100/64.13
Address         242:5000/4.13


#
# AREAS.BBS EchoMail distribution list
#
AreasBBS        /usr/local/lib/fidogate/areas.bbs

#
# FAREAS.BBS file distribution list
#
FAreasBBS       /usr/local/lib/fidogate/fareas.bbs

#
# ftnaf: CC of reply mails
#
CCMail          postmaster@spinnaker.rhein.de

#
# OPTIONAL for tosser
#
# Kill empty NetMails addressed to one of our AKAs
#
KillEmpty
```

```
#
# Kill EchoMail for unknown areas
#
#KillUnknown

#
# Kill routed EchoMail
#
#KillRouted

#
# Log sender/recipient of all NetMail messages
#
LogNetMail

#
# Check ^APATH for our own address
#
CheckPath

#
# Zonegate configuration for EchoMail
#
#               Address to      Add to SEEN-BYs
#               ----------      ---------------
#ZoneGate       2:123/456       123/456 2452/110
#ZoneGate       2:789/999       789/999 2452/110
#ZoneGate       1:105/42        105/42 2452/110

#
# Add extra nodes to SEEN-BY
#
#               Area            Nodes
#               ----            -----
#AddToSeenBy    ABLED           2:24/24
#AddToSeenBy    ABLENEWS        2:24/24
```

Letztendlich muß noch das File `config.gate` angepaßt werden:

```
#:ts=8
#
# /usr/local/lib/fidogate/config.gate
#
# FIDOGATE config file gateway programs
#
# spinnaker.rhein.de
#
# Format:  keyword arg ...
#          keyword and args may be put in double quotes "..."
```

```
#
# Include common stuff (%L = LIBDIR)
#
include         %L/config.common

#
# FTN addresses - there must be a corresponding `uplink' statement
# for each `address' statement
#
#               real
#               ----
Address         2:2450/111.13
Address         21:100/64.13
Address         242:5000/4.13

Uplink          2:2450/111
Uplink          21:100/64
Uplink          242:5000/4

#
# Default origin line for EchoMail messages
#
Origin          " Spinnaker - Linux-Point der FloKiste Koeln "

#
# Organization header for News
#
Organization    "Spinnaker-FTN-UseNet-Gate"

#
# Don't generate From: line, FSC-0035 kludges
# MUST be set for FIDOGATE points (no real gateway)
#
NoFromLine

#
# FTN - Internet gateway. If set, Internet mail will be routed via FTN
# and this gateway.
#
#Gateway                242:4900/99

#
# Generate `User.Name@do.main' instead of `User_Name@do.main'
#
#DotNames

#
# X-FTN header:
#
```

```
#     f    X-FTN-From
#     t    X-FTN-To
#     T    X-FTN-Tearline
#     O    X-FTN-Origin
#     V    X-FTN-Via
#     D    X-FTN-Domain
#     S    X-FTN-Seen-By
#     P    X-FTN-Path
#
X-FTN           ftTOVDP

#
# Maximum size of FTN messages (default value)
#
MaxMsgSize              15000
#MaxMsgSize             32000

#
# MAUS gateway support:
#
# If set, convert addresses for `User_Name@XY.MAUSDomain' to
# `User Name % XY @ MAUSGate' for the Fido-MAUS gateway.
#
MAUSDomain              .maus.de
MAUSGate                2:2452/101.6

#
# CC of bounced messages
#
BounceCCMail            postmaster@spinnaker.rhein.de

#
# Send mail from message trackers to
#
TrackerMail             postmaster@spinnaker.rhein.de
```

Damit kommen wir gleich zur Datei `hosts`. Das gegebene Beispiel ist speziell auf Martin Junius' Konfiguration angepaßt und läßt sich sonst leider sonst von niemanden sinnvoll verwenden. Du hast jetzt zunächst mal die Möglichkeit, die Datei zu löschen (am besten anschließend mit `touch hosts` eine neue leere Datei anlegen), so daß keine symbolischen Namen mehr für Rechner verwendet werden. Jedoch sollten mindestens Deine eigenen Adressen darin stehen:

```
# /usr/local/lib/fidogate/hosts
#
# Fields:
#
# node         FTN address
#
# hostname     Host name in the local domain (config: HostsDomain, Domain)
#              or fully qualified domain name (trailing `.') or `-' for
#              listed node without host name.
```

```
#
# options     -p  Generate addresses with point, e.g. `p99.hippo.fido.de'
#             -d  Node is currently down (can't receive mail)
#
# node          hostname                        options
#-----          --------                        -------
#
2:2450/111      flokiste                        -p
242:5000/4      flokiste                        -p
21:100/64       flokiste                        -p
```

Alternativ kannst Du auch alle Zeilen, die nicht mit `2:` beginnen löschen, denn dann stehen nur noch die Fido-Adressen drin und Du hast keine Proble mehr. Wenn Du Lust hast, kannst Du auch Rechner von anderen Domains eintragen, wie das Beispiel `tweety.dfv.rwth-aachen.de` zeigt. Für Namen mit Domain solltest Du nur den abschließenden Punkt nicht vergessen. So könntest Du z.B. auch die Rechner von `schiele-ct.de` eintragen, die ja auch alle eine Fido- und eine Internet-Adresse haben. Wenn Du nur `schiele-ct.de`-Rechner eintragen willst, dann kannst Du im `config`-File `HostsDomain` auch auf `schiele-ct.de` ändern und dann alle `schiele-ct.de`-Rechner ohne Domain in `hosts` eintragen. Ich hoffe, das ist klar geworden, denn für alle Rechner, die nicht in `hosts` drinstehen, wird die Adresse in z.B. `p13.f111.n2450.z2.fido.sub.de` (o.ä.) gewandelt, bei den Rechnern in `hosts`, z.B. in `p13.flokiste.fido.de`. Du solltest übrigens möglichst bei allen Rechnern in `hosts` das Flag `-p` setzen, damit Adressen mit Point generiert werden. Ansonsten wird die Point-Nummer nämlich gelöscht und dann muß der Message-Tracker des Nodes wieder unnötige Arbeit leisten.

Des weiteren kannst Du noch die `bounce.*`-Dateien auf Dein System anpassen, aber ich kann mich nicht entsinnen, daß die bei mir schonmal benötigt wurden. Auch `aliases` kannst Du anpassen, brauchst Du aber nicht.

Es bleibt `areas`, welches mindestens alle Echmail-Areas, die Du verwendest, enthalten sollte. Wenn mehr Areas darin stehen, stört das aber auch nicht. Man beachte, daß die deutschen Fido-Echos (z.B. `LINUX.GER`) in `fido.ger`-Hiearchie (z.B. `fido.ger.linux`) konvertiert werden. Internationale Fido-Echos werden nach `fido.int.*` umbenannt. Die Namen der Usenet-Echos werden nur in Kleinschreibung konvertiert. Weiterhin sollte man sich für jede andere Domain (Zone) eine eigene Hierarchie anlegen. Bei allen Areas, die nicht zur Zone 2 gehören, muß weiterhin noch angegeben werden, zu welcher Zone sie gehören (zur Sicherheit kann man das natürlich auch für die Areas der Zone 2 explizit mit angeben). Es sind im übrigen noch diverse andere Flags möglich. Hier ein Ausschnitt aus meiner `areas`-Datei:

```
# /usr/local/lib/fidogate/areas
#
# Echomail area <-> News newsgroup conversion
#
# Options:
#     -z ZONE             Alternate zone AKA for this area
#     -d DISTRIBUTION     Distribution header for this newsgroups
#     -o ORIGIN           * Origin line for this area
#     -g                  No gated messages
#     -l                  Only local crosspostings
#     -x                  No crosspostings
#     -8                  Messages with 8bit ISO-8859-1 charset
#     -H                  Names match entire hierarchy, names are translated
#     -!                  Don't gate area/newsgroup [hierarchy]
#     -R LVL              ^ARFC header level
#     -m MAXSIZE          MaxMsgSize for this area (0 = infinity)
#
```

```
# All fields may be quoted in "...", order is import, first area/newsgroup
# found matches!
#
# Format:
#
# area                         newsgroup                      [-option]
# ------------------------     --------------------------     ---------
FLOKISTE.INTERN                fido.flokiste.intern           -z 2
FLOKISTE.STAT                  fido.flokiste.stat             -z 2
LINUX.GER                      fido.ger.linux                 -z 2
SAILING                        fido.int.sailing               -z 2
CT.GER                         ger.ct                         -z 21
CT.PROJEKTE                    ger.ct.projekte                -z 21
CT.PRUEFSTAND                  ger.ct.pruefstand              -z 21
CT.SUPPORT                     ger.ct.support                 -z 21
FIDO.DE                        fido.de                        -z 242
5000.INTERNET                  fido.5000.intern               -z 242
ALT.RELIGION.EMACS             alt.religion.emacs             -z 242
COMP.OS.                       comp.os.                       -z 242 -H
REC.MUSIC.                     rec.music.                     -z 242 -H
DE.ALT.BINARIES.               de.alt.binaries.               -H -!
DE.                            de.                            -z 242 -H
```

Wie man sieht, ist es für die Usenet-Areas, die bis auf Groß-/Kleinschreibung unter RFC und FTN den gleichen Namen haben, möglich, auch ganze Hierarchien (`-H`) einzubinden oder auch einzelne Sub-Hierarchien auszuschließen (`-!`).

Letztendlich solltest Du mit `touch /usr/local/lib/fidogate/passwd` noch ein leeres File anlegen, ansonsten versagt `rfc2ftn` nämlich seinen Dienst.

7 Die Installation von ifcico

Ich beziehe mich hier auf die aktuelle Version von ifcico, welche sich im Paket `ifmail-2.8g` findet. Leider hatte Version 2.8d einen Bug, so daß ich von dieser Version nur dringend abraten kann (alle vorherigen Versionen funktionierten ordnungsgemäß).

Zunächst muß man das globale Compiler-Konfigurations-File `ifmail/CONFIG` editieren. Ich führe hier einfach mal alle bei mir nötigen Änderungen auf:

```
CONFIGFILE  = "/usr/local/lib/fnet/config"
LOCKDIR     = "/var/lock"
PUBDIR      = "/var/spool/uucppublic"
BINDIR = /usr/local/lib/fnet
OWNER = uucp
```

Der eine oder andere wird weiterhin ggf. noch einige andere Pfade ändern wollen. So wird dabei das Debug-File in `/tmp` angelegt und das Logfile in `/var/log/fnet`, was allerdings nicht stört, da das Logfile seit ifcico 2.3 sowieso vom `syslogd` gehandelt wird (mehr dazu später).

Bei Slackware 3.0 wird das Lockfile leider immer noch in `/var/spool/uucp` angelegt, was zu Problem mit einem eventuell laufendem getty auf diesem Port fuehrt. Der Eintrag `LOCKDIR` muss also hier auf `/var/spool/uucp` gesetzt werden.

Jetzt sollte man noch das Verzeichnis /usr/local/lib/fnet anlegen (User uucp, Group uucp) und kann dann mit

```
make depend
make all
make install
```

die Programmdateien in /usr/local/lib/fnet installieren.

Nun muß man noch das config-File von ifcico installieren. Ein Beispiel findet man in ifmail/misc/config. Diese kopiert man nun nach /usr/local/lib/fnet und editiert es passend. Hier ist ein Beispiel:

```
# /usr/local/lib/fnet/config
#
# IFmail - Configurations-File
# von Roland Rosenfeld 07.08.94

# Log file name. Overrides compile-time default.
logfile         /var/log/fnet/iflog

# Debug file name. Overrides compile-time default.
debugfile       /var/log/fnet/ifdebug

# Debugging verbosity level (is overidden by -x key). Default is 0.
# Folgende Kombination generiert ein Debug-File von ertraeglicher
# Laenge, das man noch lesen kann:
verbose                 bcdefghijklmnqrt

# Main address:
address         2:2450/111.13@fidonet

# AKAs:
address         21:100/64.13@gernet
address         242:5000/4.13@fidode

# Passwords: Hier koennen die Passwords direkt stehen oder man lagert sie in
# eine externe Datei aus, welche dem User uucp gehoert und die Permissions
# 600 hat, damit niemand die Passwords lesen kann.
#include                /usr/local/lib/fnet/passwords
password        2:2450/111      GEHEIM
password        21:100/64       GEHEIM
password        242:5000/4      GEHEIM

# Directory for incoming packets/files:
inbound         /var/spool/fnet/inbound
# Directories for "listed" and "protected" sessions
listinbound     /var/spool/fnet/inbound
protinbound     /var/spool/fnet/inbound

# Directory for outgoing packets (default domain and zone):
# other zones will be like "/usr/spool/fnet/outb.003",
```

```
# other domains will be like "/usr/spool/fnet/<domain>.<zone>"
outbound        /var/spool/fnet/outbound

# Directory from which the file requests are satisfied
public          /var/spool/uucppublic

# Directory with executables to satisfy "magic" file requests
# if requested a file present in this directory, it will be
# executed and stdout sent to the remote system.  If the file
# is not executable, it is read line by line and the lines are
# processed as if they were received file requests (recusively).
# Execution of commands may compromize security!  You are warned.
magic           /usr/local/lib/fnet/magic

# Erste Nodeliste:
# Es wird nach nl_short.xxx gesucht und die neueste Version verwendet.
# nl_short generiere ich manuell aus der kompletten Nodeliste, da mir das
# Compilieren der kompletten Liste zu lange dauert.
nodelist        /var/spool/fnet/nodelist/nl_short

# Weitere Nodelisten der anderen Zonen:
# (Achtung: ifindex hat noch einen Bug, der zu einem Core fuehrt, falls eine
#  Nodenummer doppelt auftritt, wie z.B. bei Verwengung von nodelist und
#  r24classic)
#               filename        originating address
nodelist        gernet          21:100/64.13@gernet
nodelist        242_list        242:5000/4.13@fidode

# Sequencer file (used to generate unique IDs)
sequencer       /usr/local/lib/fnet/seq

# An dieser Stelle stehen im Beispiel-Config-File (ifmail/misc/config) noch
# diverse Erlaeuterungen und Optionen, die ich groesstenteils nicht verwende
# und daher hier nicht auffuehre. Man sollte sich die Moeglichkeiten jedoch
# dort mal ansehen.

# Das Modem haengt an /dev/ttyS1, der FIFO ist auf 38400 gelockt (ohne FIFO
# sollte man stattdessen auf 19200 locken).
# Ich verwende ausdruecklich ttyS und nicht cua, da bei mir noch mgetty
# laeuft. Eine genau Erklaerung zu dem Thema findet sich in der Doku zu
# mgetty+sendfax von Gert Doering.
ModemPort       ttyS1:L38400

# Nun die Konvertierung der Telefonnummern. Ganz habe ich das noch nicht
# verstanden, aber folgendes sollte (mit der entsprechenden eigenen Vorwahl)
# fuer ganz Deutschland korrekt sein. Fuer Spezialfaelle sollte man sich ggf.
# nochmal ifmail/misc/config ansehen.

PhoneTrans      49-2236-        /
PhoneTrans      49-     /       0
```

```
PhoneTrans      /       00

# In ModemReset baue ich die passenden AT-Befehle fuer mein Modem ein:
# AT Z0   = Initialisieren mit Profile 0
# AT M3   = Lautsprecher nach dem Wählen aus
# Falls Du isdn4linux einsetzt, dann solltest Du mit AT&B512 die Blocksize
# einstellen, sonst bricht die Verbindung oft unvermittelt ab.
ModemReset ATZ0\r\n\dATM3\r\n\d

# Pulswahl:
ModemDial       ATDP\T\r
ModemHangup     ATZ\r
ModemOK         OK

# Statt einem einfachen "CONNECT" gebe ich viele Connect-Strings an, um
# nachher im Logfile sehen zu koennen, mit welcher Geschwindigkeit connected
# wurde.
ModemConnect    CONNECT\s76800
ModemConnect    CONNECT\s64000
ModemConnect    CONNECT\s57600
ModemConnect    CONNECT\s38400
ModemConnect    CONNECT\s19200
ModemConnect    CONNECT\s16800
ModemConnect    CONNECT\s14400
ModemConnect    CONNECT\s1200
ModemConnect    CONNECT\s9600
ModemConnect    CONNECT\s7200
ModemConnect    CONNECT\s4800
ModemConnect    CONNECT\s2400
ModemConnect    CONNECT\s12000
ModemConnect    CONNECT\r
ModemError      BUSY
ModemError      NO\sCARRIER
ModemError      NO\sDIALTONE
ModemError      NO\sANSWER
ModemError      RING\r
ModemError      ERROR

# Timeouts to wait for "OK" and "CONNECT", cannot be prefixed by logical
# expression.
TimeoutReset    3
TimeoutConnect  90

# Hier stehen noch weitere Konfigurationsmoeglichkeiten fuer Inbound-Calls.
# Da ich keine Inbound-Calls zulasse, habe ich den Teil hier auch nicht
# aufgefuehrt, fuer weiter Infos siehe ifmail/misc/config.

# EMSI data for this node
# From this line on values CANNOT be prefixed with logical expression
# For now, escaping of '}' and ']' unimplemented, try to avoid these
```

```
# characters please!
Name            *** Spinnaker ***
Location        Bornheim-Widdig
SysOp           Roland Rosenfeld
Phone           - Unpublished -
Speed           9600
Flags           XA,V32B,V42B
```

Um die Log-Messages von `ifcico` zu bekommen, solltest Du nun noch den `syslogd` vernünftig konfigurieren. Hierzu mußt Du `/etc/syslog.conf` editieren. Die Messages von `ifcico` erscheinen dabei als `local0.*` Ich habe zu diesem Zweck folgende Zeile eingebaut:

```
local0.*                /var/log/fnet/ifmail
```

(Achtung: Keine Leerzeichen, sondern ausschließlich Tabs verwenden!).

Damit landen alle Log-Meldungen von `ifcico` in `/var/log/fnet/ifmail`. Bei neuen Versionen von `syslogd` reicht nun ein

```
killall -1 syslogd
```

um, das dem Dämon das neue Config-File nahe zu bringen. Bei älteren Versionen muß man das Logfile zunächst anlegen und dann den Dämon killen und neustarten.

Jetzt sollte man allerdings noch die Outbound-Directories in `/var/spool/fnet/` anlegen. Neben dem Standard `outbound` für Fido fehlen noch `fidode` und `gernet` für die anderen Netze. Die Verzeichnisse sollte alle `uucp.uucp` gehören und die Permissions `775` besitzen. `ifcico` legt im übrigen für weitere Netze fehlende Verzeichnisse selber an, allerdings sollte man dann manuell die Permissions entsprechend überarbeiten.

8 Nodelisten-Compiler

Alle im Config-File angegebenen Nodelisten sollten im Verzeichnis `/var/spool/fnet/nodelist` vorhanden sein. Dann ruft man einfach `ifindex` auf, welcher dann `index.dir` und `index.pag` erstellt, welche die compilierte Nodeliste enthalten.

Leider ist `ifindex` nicht besonders schnell, daher kürze ich die `nodelist` immer vor der Compilierung auf das, was ich (in Deutschland) benötige. Hierfür verwende ich folgendes Skript:

```
#!/bin/sh
#
# /var/spool/fnet/nodelist/strip.nodelist
#
if [ ! $1 ]
then
    echo Fallscher Aufruf!
    echo Start mit $0 xxx, wobei xxx die Nummer der aktuellen Nodelist ist.
    exit
fi

NODELIST=nodelist.$1
```

```
if [ ! -s $NODELIST ]
then
    echo $NODELIST ist keine korrekte Nodelist
    exit
fi

NLSHORT=nl_short.$1

if [ -s $NLSHORT ]
then
    echo $NLSHORT existiert schon und wird nach $NLSHORT.bak geschoben
    mv -f $NLSHORT $NLSHORT.bak
fi

head -n 2 $NODELIST >> $NLSHORT
grep ',49-\|^Zone,' $NODELIST >> $NLSHORT

echo Fertig.
```

Dieses Skript startet man mit `strip.nodelist` *xxx*, wobei *xxx* die aktuelle Nummer der Nodeliste ist. Aus `nodelist.`*xxx* wird damit `nl_short.`*xxx* generiert.

Nun muß man natürlich auch Diffs einarbeiten können. Hierzu gibt es seit ifmail-2.6 das Programm `nlpatch`, welches man mit der alten Nodelist sowie dem aktuellen Diff als Parameter aufruft, z.B.:

```
nlpatch nodelist.123 NODEDIFF.130
```

auf (natürlich mit den entsprechenden Tag-Nummern). Es wird dann automatisch nodelist.130 generiert.

9 Poll-Skript und Entpacken

Zum Pollen verwende ich das Skript `ifmail/misc/contrib/ifpoll` in leicht abgewandelter Form:

```
#!/bin/sh
# ver 0.7r
# ifpoll, poll my boss node or the node given as argument 1
#
# i start this shell script every day by crond, but you can
# start it also by hand :) start it as the owner of ifcico.
# rasca, berlin 1993 (Rasca Gmelch, 2:2410/305.4)
#
# Erweiterte Fassung von Roland Rosenfeld

# where "ifcico" and "ifpack" reside
FIDOPATH=/usr/local/lib/fnet

# logfile of ifcico
IFLOG=/var/log/fnet/ifmail
IFDEBUG=/var/log/fnet/ifdebug
```

```
# owner of "ifcico"
IFCICO_OWNER=uucp

# sysop of fido stuff
IFCICO_SYSOP=postmaster

# my boss node (default address to poll)
NODE="f111.n2450.z2"

# how often should i try to call NODE?
MaxTry=20

# delay between outgoing calls in seconds
DELAY=60

# where to log processing - file or tty/console
INFO_TTY=/dev/tty11

function GetConnectMessage () {
   grep 'chat got "CONNECT.*", continue' $IFLOG \
      | tail -n1 \
      | sed 's/.*chat got //;s/, continue//'
}

function GetNoConnectMessage () {
   grep 'chat got .*, aborting' $IFLOG \
      | tail -n1 \
      | sed 's/.*chat got //;s/, aborting//'
}

mv -f $IFDEBUG $IFDEBUG.old
touch $IFDEBUG

echo "`date \"+%b %d %T\"` ifpoll[$$]: starting" >> $INFO_TTY

# remember me, not to run as root..
#
if [ `whoami` != "$IFCICO_OWNER" ]; then
        echo "*** run $0 as the owner of ifcico ***"
        echo "`date \"+%b %d %T\"` ifpoll[$$]: wrong uid (rc 2)" >> $INFO_TTY
        exit 2
fi

# argv[1] is the optional node to call
#
if [ "$1" != "" ]; then
        if [ "$1" = "-?" ] || [ "$1" = "-h" ]; then
                echo "usage: ifpoll [node]"
                exit 3
        else
```

```
                NODE=$1
        fi
fi

# let's pack the fido stuff..
#

$FIDOPATH/fgpack

# loop until ifcico could connect the node or MaxTry is encountered
#
i=1; errlv=1
while let 'i <= MaxTry' && let 'errlv != 0'
do
        echo -n "`date \"+%b %d %T\"` ifpoll[$$]: $i. try ($NODE) " \
                >> $INFO_TTY
        #
        # start ifcico in master mode ..
        #
        $FIDOPATH/ifcico -r 1 $NODE
        errlv=$?
        if [ $errlv != "0" ]; then
                GetNoConnectMessage >> $INFO_TTY
                if [ $i != $MaxTry ]; then
                        sleep $DELAY
                fi
                let i=i+1
        else
                GetConnectMessage >> $INFO_TTY
        fi
done

# if the poll was fine, unpacking..
#
if [ $errlv = "0" ]; then
        echo "`date \"+%b %d %T\"` ifpoll[$$]: unpacking.. " >> $INFO_TTY
        $FIDOPATH/fgunpack $INFO_TTY
        # add here some additional lines for processing tic files or
        # incoming file-lists or simular..
        grep 'chat got .*, aborting' $IFLOG | \
                tail -n20 | \
                elm -s "ifpoll: failed" $IFCICO_SYSOP >/dev/null
fi

echo "`date \"+%b %d %T\"` ifpoll[$$]: finished (rc $errlv)" >> $INFO_TTY

# return the errorlevel of ifcico
exit $errlv
```

Dieses Skript kann ohne Parameter gestartet werden und ruft dann automatisch meinen Boß an, oder aber man gibt den gewünschten Node an (in *pfnz*-Notation, also z.B. `ifpoll f111.n2450.z2`). Natürlich muß das Skript an den eigenen Bedarf angepaßt werden (so hat nicht jeder 12 Consolen etc.).

Dieses Skript ruft zunächst das Pack-Skript auf:

```
#!/bin/sh
# /usr/local/lib/fnet/fgpack
# pack-script for FidoGate (together with ifcico)
#
# written by Roland Rosenfeld   19.08.94
#       roland@p13.flokiste.fido.de     (2:2450/111.13)
#

FNET=/var/spool/fnet
FTNPACK=/usr/local/lib/fidogate/ftnpack

PATH=/bin:/usr/bin:/usr/local/bin

topack=

for pkt in $FNET/*/????????.out $FNET/*/????????.pnt/????????.out
do
        if [ -s $pkt ]
        then
                topack="$topack $pkt"
        fi
done
$FTNPACK $topack
```

Damit werden die `.out`-Files in ARCmail-Files zusammengepackt, damit sie effizienter verschickt werden können.

Damit `ftnpack` korrekt arbeitet, muß man in `/usr/local/lib/fidogate/packing` noch den Packer definieren:

```
# /usr/local/lib/fidogate/packing
#
# FIDOGATE ftnpack config file
#
# Commands:
#
#     arc    NAME  "/PATH/PROG %s %s"
#     prog   NAME  "/PATH/PROG %s"
#
#     pack   NAME  NODES
#     rpack  NAME  TARGET NODES
#     fpack  NAME  TARGET NODES
#

arc     zip     "/usr/bin/zip -gkjq %s %s"
arc     arc     "/usr/local/bin/arc an %s %s"

prog    gate    "/usr/local/lib/fidogate/ftn2rfc %s"
```

```
######## P A C K I N G ##########################################################

pack    zip     *
```

Natürlich sollte man die Pfade der Packer auf die eigenen Bedürfnisse anpassen. Nach erfolgreichem Anruf von `ifcico` wird dann das folgende Entpack-Skript aufgerufen:

```
#!/bin/bash
# usr/local/lib/fnet/fgunpack
#
# unpack-script for FidoGate (together with ifcico)
#
# written by Roland Rosenfeld   12.06.94
#       roland@p13.flokiste.fido.de     (2:2450/111.13)
#
# corrected paths, usage of syslog, more packer, clean up, restructured
#       martin@erde.GUN.de (Martin Seine) <2:2448/413.100>
#

FNET=/usr/local/lib/fnet
FGATE=/usr/local/lib/fidogate
IFCFG=$FNET/config
IN=/var/spool/fnet/in

# Facility is the log-file facility, where syslog stores the messages
# if you're not using syslog, no need to change it
FACILITY=local0

# System-manager who receives notices, if there are unpacking errors
MANAGER=postmaster

PATH=/bin:/usr/bin:/usr/local/bin:$FGATE

if [ $1 ]
then
        INFO_TTY=$1
else
        INFO_TTY=/dev/console
fi

WEARE=`basename $0`

#
# get directory-names from $IFCFG
#
INBOUND=`grep -i "^[    ]*inbound" $IFCFG | awk '{ print $2 }'`
LOGFILE=`grep -i "^[    ]*logfile" $IFCFG | awk '{ print $2 }'`

if [ ! -d $INBOUND/bad ] ; then
   if [ -e $INBOUND/bad ] ; then
```

```
		rm -Rf $INBOUND/bad
	fi
	mkdir $INBOUND/bad
fi

CORRECT=true

function iflog() {
	if [ -S /dev/log ] ; then
		logger -i -p $FACILITY.info -t $WEARE $@
	else
		echo "`date \"+%y/%m/%d %T\"` $$ $WEARE:" $@ >> $LOGFILE
	fi
	echo "`date \"+%b %d %T\"` $WEARE [$$]:" $@ >> $INFO_TTY
}


function unpack_mail() {
	pushd $INBOUND >/dev/null
	EMPTY=true
	for f in *.mo? *.tu? *.we? *.th? *.fr? *.sa? *.su? *.pkt \
		 *.MO? *.TU? *.WE? *.TH? *.FR? *.SA? *.SU? *.PKT
	do
		if [ -f $f ] ; then
			if [ $EMPTY = true ] ; then
				rm -rf bak.oo
				mv -f  bak.o  bak.oo >/dev/null
				mv -f  bak    bak.o  >/dev/null
				mkdir  bak
			fi
			mv -f  $f bak/
			ln -s -f $INBOUND/bak/$f $IN/$f
			EMPTY=false
		fi
	done

	# now all new packets lay in $INBOUND/bak

	if [ $EMPTY = true ]
	then
		iflog No new mail found in $INBOUND
		exit
	fi

	popd >/dev/null

	pushd $IN >/dev/null
	#
	# Analyze packer with file(1) and unpack them if possible
	#
```

```
   for f in *.mo? *.tu? *.we? *.th? *.fr? *.sa? *.su?
   do
       if [ -f $f ] ; then
          arc=`file -L $f | awk '{ print $2 }'`
          case $arc in
            ARJ) unarj e $f ;;
            ARC) arc e $f ;;
            ZIP) unzip -x $f < /dev/null ;;
            LHA) lha e $f ;;
            ZOO) zoo eq $f ;;
            RAR) unrar e $f ;;
              *) iflog unknown packer \'$arc\' for $f
                 false ;;
          esac
          if [ $? -gt 0 ] ; then
              iflog couldn\'t unpack $f \(moved to $INBOUND/bad\)
              cp -f $f $INBOUND/bad
              CORRECT=false
          else
              iflog unpacked $f \($arc\)-archive
              rm -f $f
          fi
       fi
   done
   popd >/dev/null
}

#
#
# main unpack-program
#
#

export FNET
cd $FNET
unpack_mail

iflog starting ftn2rfc
ftn2rfc -x ftninpost -l

if [ $CORRECT = false ] ; then
/usr/lib/sendmail $MANAGER <<END
Subject: Fido-packet errors
There occured errors while processing Fido-Packets.

Please check the logfiles

Your Gateway
END
fi
```

```
#iflog starting Linux-TIC-Processor
#/usr/local/lib/tic/process_tics.pl < /dev/null
#/usr/local/lib/tic/poster.pl Daily < /dev/null
#/usr/local/lib/tic/lister.pl < /dev/null
```

Dieses Skript schiebt alle in `/var/spool/fnet/inbound` angekommenen Mails nach `/var/spool/fnet/inbound/bak`, wobei dieses Verzeichnis vorher in `bak.o` und `bak.oo` verschoben wird, so daß man immer die letzten drei Poll-Resultate vorliegen hat um mögliche Probleme auch nachträglich noch beheben zu können. Anschließend werden alle `.pkt`-Files nach `/var/spool/fnet/in` kopiert und die Archive werden dorthin entpackt (das Skript erkennt automatisch ZIP, ARJ, ARC und LHA). Hierzu müssen folgende Zeilen in `/etc/magic` enthalten sein (möglichst ganz oben, denn die Datei wird von oben nach unten durchsucht):

```
# Einige Eintraege fuer das ifcico-Shell-Script
#
0       byte            0x1a            ARC Archive (maybe)
0       string          PK              ZIP Archive
2       string          -lh             LHA Archive
0       string          ZOO             ZOO Archive
0       short           0xea60          ARJ Archive
```

Ist das geschehen, wird `ftn2rfc` aufgerufen, welches die Pakete nach Mail/News konvertiert und dann seinerseits mittels `ftninpost` `sendmail` und `rnews` startet.

Im Anschluß daran können noch weitere Tools, wie z.B. ein TIC-Prozessor oder ein Programm, das News für User des eigenen Systems sucht, aufgerufen werden.

10 Konfiguration von smail

Wenn Du Dich für Sendmail entschieden hast, kannst Du dieses Kapitel überspringen.

Bei mir liegt smail im Verzeichnis `/usr/local/lib/smail`, wobei `/usr/lib/smail` ein Link darauf ist. Die Konfigurationsdateien liegen dabei gemäß Filesystem-Standard in `/etc/smail`, bei einigen Distributionen auch in `/var/lib/smail`, dann muß man das entsprechend anpassen. Alle von mir hier beschriebenen Dateien sollten also in diesem Verzeichnis liegen. Die systemweite Alias-Datei liegt gemäß FSS in `/etc/aliases`, während sie früher in `/usr/lib/aliases` zu finden war.

Zunächst die allgemeine Konfiguration für die Benutzung ohne UUCP-Connection zum Internet:

```
# /etc/smail/config
#
#       smail configuration for p13.flokiste.fido.de
# (see smail(5) man page for details and other options)
#
-smtp_debug
hostname=p13.flokiste.fido.de
# more_hostnames gibt weitere Adressen an, die hier ausgewertet werden sollen
#   flokiste.fido.de muss hier auch aufgenommen werden, da ggf. Mails an
#   Roland Rosenfeld 2:2450/111 gehen, die dann vom ITrack meines Uplink an
#   mich umgelenkt werden.
#   Hostnamen mit pxxx.fxxx.nxxx.zxxx-Namen koennen hier entfallen, da die
```

```
#   alle in /usr/local/lib/fidogate/hosts auf p13.flokiste.fido.de oder
#   flokiste.fido.de gemappt werden sollen.
more_hostnames=flokiste.fido.de:spinnaker:spi:p13
-visible_name
-smart_path
-uucp_name
error_copy_postmaster
postmaster=postmaster
```

Wenn Du noch einen direkten UUCP-Zugang zum Internet hast, mußt Du folgende Zeilen hinzufügen:

```
smart_path=rhein
smart_transport=uux
uucp_name=spinnaker.rhein.de
```

und die Zeile `-smart_path` löschen. rhein ist dabei der UUCP-Name meines Uplinks.

Nun muß angegeben werden, was wie geroutet werden soll:

```
# /etc/smail/routers
#
# smail routers for p13.flokiste.fido.de

# See smail(5) for a complete description of the contents of this
# file.

fido:
        driver = pathalias,
        transport = fido;
        file = /etc/smail/paths.ftn,
        proto = lsearch

smart_host:
        transport=fido,
        driver=smarthost;
        path=p0.f4.n5000.z242.fido.de
```

Hier wird also alles über fido geroutet, wobei `paths.ftn` (s.u.) angibt, wo welche FTN-Adressen eingesetzt werden sollen. Alles, was nicht gemäß `paths.ftn` verarbeitet werden kann, wird dabei an das fido.de-Internet-Gate geschickt.

Für die Konfiguration mit UUCP wird ein anderer Smarthost (dahin wird alles geschickt, was nicht vorher definiert wurde) verwendet:

```
smart_host:
        driver=smarthost,
        transport=uux
```

`paths.ftn` sieht nun wie folgt aus:

```
# /etc/smail/paths.ftn
#
.fido.sub.de   p0.f111.n2450.z2.fido.sub.de!%s
.z2            p0.f111.n2450.z2.fido.sub.de!%s
.fido.de       p0.f111.n2450.z2.fido.sub.de!%s
.maus.de       p0.f111.n2450.z2.fido.sub.de!%s
#
.ger.sub.de    f64.n100.z21.ger.sub.de!%s
#
.z242.fido.de  p0.f4.n5000.z242.fido.de!%s
```

Letztendlich muß man noch angeben, wie die einzelnen Mails transportiert werden sollen:

```
# /etc/smail/transports
#
fido:   driver = pipe;
        group = uucp,
        cmd = "/usr/local/lib/fidogate/rfc2ftn -w Normal ${strip:user}",
        pipe_as_sender
```

Da Messages aus dem Fido an den Realnamen adressiert werden, sollte selbiger (mit Unterstrich zwischen Vor- und Nachname) auch auf dem eigenen System vorhanden sein. Hierzu trägt man ihn in `/etc/aliases` ein, so daß die Mail an den User selbst weiter geschickt wird. Hier ein Ausschnitt aus meinem Alias-File, wo die häufigsten aliases aufgeführt sind, so daß Mails mich praktisch immer erreichen, wenn sie an mich adressiert sind:

```
# /etc/aliases
#
# This is used by smail for sendmail compatibility.
# See also the /usr/lib/smail/lists directory for another way
# to make mail lists.
#
# Note: if your /etc/smail/directors says that the aliases
# director uses proto=dbm, you must use mkaliases to make
# /etc/aliases.pag and /etc/aliases.dir, which are the
# actual files used by smail.
#
# If you use proto=lsearch, smail will read this file.  (That's
# much slower but it's OK if this file is small.)
#
support: roland
admin: roland
Roland_Rosenfeld: roland
Roland.Rosenfeld: roland
postmaster: roland
faxadmin: roland
usenet: roland
sysop: roland
rosenfeld: roland
roro: roland
uucp: roland
news: roland
```

Man sollte die Funktion dieser Datei allerdings zunächst mal mit einer lokalen Mail testen, denn einige Versionen von smail (wenn sie mit ungünstigen Parametern compiliert wurden) scheinen das File nur als dbm zu erkennen, auch wenn man `lsearch` angibt (siehe Kommentar in `aliases`). Daher sollte man nach jeder Änderung an diesem File `mkaliases` aufrufen, welches `aliases.dir` und `aliases.pag` auf den neusten Stand bringt.

Da viele Fehlermeldungen des Newssytems an `usenet`, `news` oder `uucp` geschickt werden, sollte man diese Messages ebenfalls auf sich selber umleiten.

11 Sendmail V8-Konfiguration

Wenn Du Sendmail installierst, solltest Du besonders darauf achten, daß Du den kompletten Satz an Configurations-Files installierst, welcher sich bei Slackware in dem Paket `smailcnf.tgz` versteckt. Diese Configurations-Files finden sich in einem Verzeichnisbaum unter `/usr/src/sendmail/cf`. In diesem Verzeichnis sollte sich auch ein knapp 50KB großes README befinden, daß die neue Konfigurationsmethode beschreibt.

Obwohl Sendmail über ein gewöhnlich gut 20KB großes Configurations-File (`/etc/sendmail.cf`) gesteuert wird, mußt man dieses File nicht selber schreiben (das denken nämlich die meisten Leute, die von Sendmail abgeschreckt werden). Vielmehr benötigt man den Macroprozessor M4, und ein kleines Konfigurationsfile (mc-File) in `/usr/src/sendmail/cf/cf`. M4 erzeugt dann aus dem mc-File und den anderen (vom User normalerweise nicht zu ändernden) Config-Files das `sendmail.cf`-File.

Zunächst einmal muß FidoGate allerdings noch bekannt gemacht werden. Hierzu muß das File `/usr/src/sendmail/cf/mailer/ftn.m4` mit folgendem Inhalt angelegt werden:

```
PUSHDIVERT(-1)
#
# /usr/src/sendmail/cf/mailer/ftn.m4
#
# FIDOGATE FTN mailer for sendmail V8
#
# MAILER(smtp) and MAILER(uucp) must be included!
#

ifdef(`FTN_MAILER_PATH',, `define(`FTN_MAILER_PATH',
                                   /usr/local/lib/fidogate/rfc2ftn)')
ifdef(`FTN_MAILER_ARGS',, `define(`FTN_MAILER_ARGS', `rfc2ftn -w normal $u')')
ifdef(`FTN_MAILER_FLAGS',, `define(`FTN_MAILER_FLAGS', `')')
ifdef(`FTN_MAX_SIZE',, `define(`FTN_MAX_SIZE', 100000)')
POPDIVERT
#####################################
###    FTN Mailer specification   ###
#####################################

VERSIONID(`ftn.m4 V1.5')

ifdef(`_MAILER_smtp_',
`# FIDOGATE mailer
Mftn,   P=FTN_MAILER_PATH, F=CONCAT(mDFMhu, FTN_MAILER_FLAGS),
        S=52/31, R=ifdef(`_ALL_MASQUERADE_', `11/31', `21'),
        A=FTN_MAILER_ARGS
```

```
')
```

Wie man sieht, ist die Konfiguration zur Verwendung von rfc2ftn nicht allzu aufwendig, aber was da genau passiert, braucht uns auch nicht zu kümmern, wir verwenden den eben definierten Mailer ftn"jetzt einfach, wie die vordefinierten Mailer ßmtp", üucpëtc.

Kommen wir also nun zur Konfiguration des mc-Files. Ich habe es `spinnaker.mc` genannt, aber der Name tut nichts zur Sachen, das File sollte nur in `/usr/src/sendmail/cf/cf` liegen. `spinnaker.mc` sieht bei mir so aus:

```
# /usr/src/sendmail/cf/cf/spinnaker.mc
#
# sendmail V8 configuration for spinnaker.rhein.de
# using UUCP and FidoGate
#
include(`../m4/cf.m4')
VERSIONID(`spinnaker.mc V1.16')

OSTYPE(linux)dnl

define(`confUSE_ERRORS_TO', `True')dnl
define(`confCOPY_ERRORS_TO', `postmaster')dnl
define(`confMIME_FORMAT_ERRORS', `False')dnl

FEATURE(notsticky)dnl
FEATURE(mailertable,hash -o /etc/sendmail/mailertable)dnl
FEATURE(always_add_domain)dnl
FEATURE(nodns)dnl
FEATURE(nocanonify)dnl

MAILER(local)dnl
MAILER(ftn)dnl
MAILER(smtp)dnl
MAILER(uucp)dnl

Cwspi spi.rhein.de
Cwp13 p13.flokiste.fido.de
Cwp13.f111.n2450.z2.fido.sub.de
Cwp13.f64.n100.z21.ger.sub.de

# define(`SMART_HOST', uucp-dom:rhein)
define(`SMART_HOST', ftn:f4.n5000.z2.fido.de)

LOCAL_RULE_3
# Route fidonet.org and fido.sub.org via FTN!
R$* < $* z2.fidonet.org > $*            $1 < $2 z2.fido.sub.de > $3
R$* < $* fido.sub.org > $*              $1 < $2 fido.sub.de > $3
```

Hier wird zunächst das Betriebssystem (Linux) definiert, anschließend wird definiert, daß alle MAILERDAEMON-Mails auch an postmaster geschickt werden, damit ich Fehler schnell finden und beheben kann. Anschließend werden einige Optionen angegeben, die meine genaue Konfiguration angeben, die Du aber so übernehmen kannst. Weiterhin werden die Mailer local"(für Mails an User auf dem gleichen Rechner, welche mittels deliver ausgeliefert werden, das bei Slackware im Sendmail-Paket enthalten ist), ftn"(Fido-compatible Netze via FidoGate), ßmtp"(Transport über das Online-Mail-Protokoll SMTP) und üucp"(Transport per UUCP) definiert. Die letzten beiden sind für ein System, das nur per FidoGate am Internet hängt nicht nötig, aber der Mailer ftn"basiert auf ihnen (nur so konnten wir das ftn.m4 so kurz halten). In den mit `Cw` beginnenden Zeilen werden jetzt noch diverse Namen für meinen Rechner definiert.

Mit

```
define('SMART_HOST', ftn:f4.n5000.z2.fido.de)
```

wird alle Mail, die nicht schon anders transportiert wird via FidoGate an meinen fido.de-Uplink geschickt. Wenn Du einen UUCP-Uplink hast, kannst Du diese Zeile durch

```
define('SMART_HOST', uucp-dom:rhein)
```

ersetzen, dabei wird alles zu meinem UUCP-Uplink rhein geschickt.

Zum Ende von `spinnaker.mc` habe ich noch ein besonderes Schmankerl eingebaut:

```
LOCAL_RULE_3
# Route fidonet.org and fido.sub.org via FTN!
R$* < $* z2.fidonet.org > $*            $1 < $2 z2.fido.sub.de > $3
R$* < $* fido.sub.org > $*              $1 < $2 fido.sub.de > $3
```

Hier wird `z2.fidonet.org` in `z2.fido.sub.de` und `fido.sub.org` in `fido.sub.de` umgemappt, so daß alle Fido-Mails an die diversen Fido-Domains nach `fido.sub.de` geschickt werden, was dann per FTN (statt per UUCP o.ä.) verschickt wird. Dabei sollte man darauf achten, daß zwischen `$*` und `$1` mindestens ein TAB stehen sollte (nicht nur Blanks).

Hierzu wird das oben definierte File `/etc/sendmail/mailertable` benötigt. In diesem File werden Äusnahmen"vom Default-Routing angegeben, so z.B. auch das Routing über Fido.

```
 .fido.sub.de     ftn:f111.n2450.z2.fido.sub.de
 .fido.de         ftn:f111.n2450.z2.fido.sub.de
 .ger.sub.de      ftn:f64.n100.z21.ger.sub.de
 .z242.fido.de    ftn:f4.n5000.z242.fido.de
```

Wie man sieht werden alle Fido-Mails per ftn transportiert. Diese Mailertable muß nun noch nach jeder Änderung in eine Datenbank compiliert werden. Hierzu gibt man im Verzeichnis `/etc/sendmail` einfach folgendes Kommando ein:

```
makemap hash mailertable.db < mailertable
```

Sollte das die Fehlermeldung makemap: Type hash not supported in this versionäusgegeben werden, hast Du vermutlich Slackware 3.0 installiert, welche aus mir unerklärlichen Gründen den Typ hash nicht verkraftet. Abilfe schafft das alternative Kommando

```
makemap dbm mailertable < mailertable
```

im selben Verzeichnis. Jetzt mußt Du allerdings auch `spinnaker.mc` ändern, damit `dbm` statt `hash` von sendmail erkannt wird. Du trägst dort jetzt in der mailertable-Zeile folgendes ein:

```
FEATURE(mailertable,dbm -o /etc/sendmail/mailertable)dnl
```

Dann sollte es keine Probleme mehr geben.

Nachdem `spinnaker.mc` jetzt fertig editiert ist, können wir daraus `sendmail.cf` erzeugen. Hierzu wechselt man ins Verzeichnis `/usr/src/sendmail/cf/cf` und gibt dort `pmake spinnaker.cf` ein. Daraufhin wird `/usr/src/sendmail/cf/cf/obj/spinnaker.cf` erstellt, welches man nach nach `/etc/sendmail.cf` schiebt und schon ist die Konfiguration fertig. Falls pmake nicht installiert sein sollte kann man stattdessen auch folgendes eingeben:

```
m4 spinnaker.mc > obj/spinnaker.cf
```

Abschließend mußt Du noch das ein Alias-File anlegen:

```
#
# /etc/aliases
#
# compile with
#       newaliases
#
nobody: /dev/null
support: roland
admin: roland
operator: roland
Roland_Rosenfeld: roland
Roland.Rosenfeld: roland
postmaster: roland
faxadmin: roland
usenet: roland
sysop: roland
rosenfel: roland
rosenfeld: roland
roro: roland
uucp: roland
news: roland
```

Dieses File muß nach jeder Änderung mid dem Kommando `newaliases` in eine Datenbank compiliert werden.

Nun muß Sendmail nur noch als Daemon gestartet werden, was mithilfe des Kommandos

```
/usr/sbin/sendmail -bd -q20m
```

in `/etc/rc.d/rc.local` geschieht. Bei vielen Systemen ist selbiges auch schon in `/etc/rc.d/rc.M` angegeben, was ebenfalls möglich ist. Alternativ kann man sendmail auch per Cron starten und den inetd auf dem Port lauschen lassen. Hierzu trägt man in die Crontab des Users root folgendes zusätzlich ein:

```
*/20  * * * *   /usr/sbin/sendmail -q
```

und fügt in `/etc/inetd.conf` die Zeile

```
smtp  stream  tcp   nowait  root  /usr/sbin/tcpd  /usr/sbin/sendmail -bs
```

Anschließend das `killall -1 inetd` nicht vergessen, um `inetd.conf` neu einzulesen.

Sendmail schreibt seine Logmessages per syslog. Hierzu empfiehlt es sich, folgenden Eintrag in `/etc/syslog.conf` aufzunehmen:

```
mail.*                          /var/log/mail
```

Nicht vergessen, den `syslogd` nach dieser Änderung mit `killall -1` behandeln, wie schon in der 7 beschrieben.

12 cnews-Konfiguration

Wenn Du Dich für INN entschieden hast, kannst Du dieses Kapitel überspringen.

Ich werde mich hier auf die Pfadkonventionen beziehen, die das cnews aus Slackware 2.1 verwendet: `/var/lib/news` für die Konfigurationsfiles, Logfiles und Statusfiles und `/usr/lib/newsbin` für die cnews-Binaries. In anderen Distributionen wird bzw. wurde das alles gemeinsam in `/usr/lib/news` und `/usr/lib/news/bin` oder `/usr/local/lib/news` und `/usr/local/lib/news/bin` plaziert.

Ich liste hier einfach mal alle wichtigen Konfigurations-Files auf:

Zunächst das `sys`-File welches angibt, welche Newsgroups wohin exportiert werden sollen:

```
# /var/lib/news/sys
#
# Achtung: keine ueberfluessigen Leerzeichen einfuegen, das kann zu
#          Problemen fuehren
#
# ME: gibt an, welche Newsgroups auf dem eigenen System gehalten werden
# sollen. Dabei sollte es auch moeglich sein einfach "all" zu schreiben. Das
# ist jedoch riskant, denn falls man auch noch uucp verwendet, dann koennten
# automatisch News-Hierarchien angelegt werden, die man nicht haben will.
ME:alt,comp,news,gnu,de,rec,fido,ger,spinnaker,junk

# FidoGate (alle FTN-Netze, die ueber FidoGate angesprochen werden):
# fido.*, ger.*, de.*, comp.*, rec.*, alt.*, gnu.*
fidogate/flokiste.fido.de:\
fido,de,comp,rec,alt,gnu/all:Lf:
```

Hierfür muß nun noch ein Outbound-Verzeichnis angelegt werden. cnews erwartet dieses Verzeichnis als `/var/spool/news/out.going/fidogate`. Dieses Verzeichnis sollte `news.news` gehören und die Permissions `775` haben.

Nun muß man angeben, wie die News per FidoGate verarbeitet werden sollen:

```
# /var/lib/news/batchparms
#
# site          size    queue   builder muncher sender
# ----          ----    -----   ------- ------- ------
/default/       100000  20      batcher compcun viauux
#
fidogate        250000  200     batcher nocomp  viafido
```

Das bedeutet, daß für fidogate das Skript `viafido` zum Transport verwendet werden soll. Weiterhin sollen maximal 250K in ein Paket gesteckt werden. Falls man wenig Platz auf der Platte hat, kann man auch die "`sizeruntersetzen (für obiges braucht man nämlich mindestens 10MB freien Speicher auf /tmp), nur werden dann kleinere .pkt-Files generiert.`

`Bei neueren Versionen von cnews hat sich das Format dieses Files geändert, so daß da stehen muß:`

```
fidogate    u     250000  200     batcher | nocomp | viafido
```

`Das File /usr/lib/newsbin/batch/nocomp fehlt wohl bei Slackware 3.0, es muß mit folgendem Inhalt neu erstellt werden:`

```
#!/bin/sh
# /usr/lib/newsbin/batch/nocomp
#
exec cat $*
```

`Das File viafido existiert nicht, es muß erst erstellt werden:`

```
#! /bin/sh
# /usr/lib/newsbin/batch/viafido
#
# Submit news batch to FIDOGATE's rfc2ftn -w.
#
# The 'exec' cuts down the number of processes active for this simple case.

exec /usr/local/lib/fidogate/rfc2ftn -w Normal -b -n
```

`Letztendlich muß noch die expire-List configuriert werden, welche angibt, welche Artikel wie schnell gelöscht werden dürfen:`

```
# /var/lib/news/explist
#
# Die History soll 30 Tage gehalten werden. Wenn Artikel ankommen, die aelter
# als 30 Tage sind, werden die gar nicht erst angezeigt, sondern sofort
# geloescht.
/expired/                       x       30      -

# Artikel werden mindestens 3, maximal 90 Tage gehalten
/bounds/                        x       3-5-90  -

# Alles weitere wird sequentiell von oben(!) abgearbeitet.
junk                            x       7       -
fido.junk                       x       7       -
control                         x       7       -
de.newusers.questions           x       8       -
de.newusers                     x       90      -
de                              x       14      -
```

```
comp.os.linux.announce          x       90      -
comp.os.linux                   x       8       -
fido.2450                       x       27      -
fido.r24                        x       10      -
fido.ger.linux                  x       18      -
fido.ger                        x       14      -

# default:  9 Tage (d.h. alles, was durch obiges noch nicht getroffen wurde)
all                             x       9       -
```

Damit sollte das News-System funktionstüchtig sein, allerdings müssen noch einige Dinge regelmäßig ausgeführt werden, damit es funktioniert.

Hierzu verwende ein paar Cron-Jobs. Die Crontab für den User news sieht dafür wie folgt aus:

```
# crontab.news
#
SHELL=/bin/sh
MAILTO=news
#
# Crontab entries for news system
#
15 * * * * /bin/date >  /tmp/news_cron_ok

# Kurz vor meinen beiden Polls werden die News gesammelt und nach Fido
# exportiert. Zusaetzlich werden sie noch stuendlich exportiert.
27      8       * * *   /usr/local/lib/fidogate/run-batch
27      20      * * *   /usr/local/lib/fidogate/run-batch
13      *       * * *   /usr/local/lib/fidogate/run-batch

# Die Artikel sollen regelmaessig fuer alle User des Systems lesbar gemacht
# werden (danach kann man sie auch noch canceln und wenn noch kein run-batch
# ausgefuehrt wurde, wird der gecancelte Artikel auch nicht einmal
# exportiert): [das */5 bedeutet, dass alle 5 Miunten newsrun gestarte
# werden soll. Fuer naehere Informationen siehe man 5 crontab]
*/5     *       * * *   /usr/lib/newsbin/input/newsrun

# Kontrollieren.... um 21:18
18      21      * * *   /usr/lib/newsbin/maint/newsdaily

# Aufraeumen...
24       8      * * *   /usr/lib/newsbin/expire/doexpire
24       20     * * *   /usr/lib/newsbin/expire/doexpire
```

Dabei ist run-batch ein kleines Shell-Script, welches zuerst newsrun und dann sendbatches aufruft:

```
#!/bin/sh
# /usr/local/lib/fidogate/run-batch
#
```

```
# Call newsrun, sendbatches
#

/usr/lib/newsbin/input/newsrun
/usr/lib/newsbin/batch/sendbatches
```

Bei Slackware 3.0 wird bei jedem Aufruf von sendbatches die UUCP-Queue überprüft, wobei eine Fehlermeldung ausgegeben wird, weil das System fidogate nicht als UUCP-System existiert. Um dieses Problem zu beheben, muß das Script queulen wie folgt geändert werden:

```
#!/bin/sh
# Find size of current queue of news outbound to $1.  Taylor version.
# =()<. ${NEWSCONFIG-@<NEWSCONFIG>@}>()=
. ${NEWSCONFIG-/var/lib/news/config}

PATH=$NEWSCTL/bin:$NEWSBIN:$NEWSPATH ; export PATH
umask $NEWSUMASK

if test "$1" = "fidogate"
then
   echo -e "\t0"
   exit 0
else
uustat -s $1 -c rnews | wc -l
fi
```

Nun müssen die Newsgroups noch in das News-System eingetragen werden. Hierzu verwendet man das Programm /usr/lib/newsbin/maint/addgroup, welches man mit dem Namen der Newsgroup sowie dem Parameter y aufruft, z.B.:

```
/usr/lib/newsbin/maint/addgroup fido.ger.linux y
```

Damit wird nun die Newsgroup fido.ger.linux (entspricht bei mir der Fido-Area LINUX.GER) eingerichtet. Dies muß man nun mit allen gewünschten Newsgroups machen. Weiterhin sollte man auch noch die beiden Newsgroups fido.junk und junk anlegen, welche News aufnehmen, die in keine andere Newsgroup einsortiert werden können. Wenn Fido-Echomail in eine Area gehört, die in /usr/local/lib/fidogate/areas nicht aufgeführt ist, so wandert diese in fido.junk. Wenn die Area in fidogate/areas eingetragen ist, aber vergessen wurde, die entsprechende Newsgroup mit addgroup anzulegen, dann wandert der Artikel nach junk. Man sollte diese Newsgroup als Systemverwalter auch subscriben, damit man fehlende Newsgroup-Einträge bemerkt und eintragen kann. Um Artikel canceln zu können, benötigt man übrigens noch die Newsgroup control, die man auch mittels addgroup anlegen sollte.

Falls noch Permission-Probleme auftreten können die auch in einem Bug von relaynews begründet sein, denn dort wird nach einem su weiterhin die alte User-ID und Group-ID verwandt. Das läßt sich aber durch folgenden Patch beheben:

```
--- org/relaynews.c     Tue Mar 17 08:42:42 1992
```

```
+++ relaynews.c Sun Jan 30 13:48:34 1994
@@ -201,7 +201,7 @@
        else
                newsuid = geteuid(), newsgid = getegid();
        if (setgid(newsgid) < 0 || setuid(newsuid) < 0 ||
-           getgid() != newsgid || getuid() != newsuid) {
+           getegid() != newsgid || geteuid() != newsuid) {
                if (getenv("NEWSPERMS") != 0)
                        error("recursive loop setting ids", "");
                /*
```

13 INN-Konfiguration

Zunächst muß man INN natürlich installieren, wofür man entweder ein vorkonfiguriertes Binary verwenden kann, oder halt sein Binary selber compiliert, was anhand des Beispiel-config.data aus dem newspak recht einfach möglich sein sollte. Hier habe ich gegenüber dem newspak folgende Änderungen vorgenommen (das ist nicht das gesamte config.data sondern nur die Unterschiede zwischen config.data.newspak und meinen Anpassungen):

```
_EXITVAL               void
INEWS_PATH             DONT
INND_NICE_KIDS         DO
INND_NICE_VALUE        10
DEFAULT_CUTOFF         30
HAVE_UUSTAT            DO
_PATH_LOGFILE          /var/log/inn/log
_PATH_MOST_LOGS        /var/log/inn
```

Weiterhin habe ich die Pfade so geändert, daß für NEWSPAK_NEWS_LIB_DIR /usr/local/lib/news und für NEWSPAK_SPOOL_DIR /var/spool/news verwendet wird. Slackware verwendt meines Wissens /usr/lib/news, so daß man hier ggf. einen Link anbringen sollte, oder etwas umdenken muß.

Nach der Installation sind in /usr/local/lib/news und dessen Unterverzeichnissen diverse Änderungen vorzunhemen. Ich werde hier zu konfigurierenden Files einfach nacheinander besprechen:

Zunächst einige kleinere Files: /usr/local/lib/news/passwd.nntp und /usr/local/lib/news/nntpsend.ctl sollten nur Kommentare enthalten und ansonsten leer sein. /usr/local/lib/news/hosts.nntp enthält nur folgende eine Zeile:

```
##  hosts.nntp - names and addresses that feed us news
##  Format
##      <host>:
##      <host>:<password>
##  <host> can be a name or IP address; no wildcards.  Any hosts not
##  listed here are handed off to nnrpd.
spinnaker.rhein.de:
```

```
Damit darf nur mein eigener Rechner auf meinen NNTP-Server zugreifen.
```

/usr/local/lib/news/nnrp.access sollte wie folgt aussehen:

```
##  nnrp.access - access file for on-campus NNTP sites
##  Format:
##      <host>:<perm>:<user>:<pass>:<groups>
##  Connecting host must be found in this file; the last match found is
##  used, so put defaults first.
##      <host>          Wildcard name or IP address
##      <perm>          R to read; P to post
##      <user>          Username for authentication before posting
##      <pass>          Password, for same reason
##      <groups>        Newsgroup patterns that can be read or not read
##  To disable posting put a space in the <user> and <pass> fields, since
##  there is no way for client to enter one.
##
## Default is no access, no way to authentication, and no groups.
*:: -no- : -no- :!*
##  Foo, Incorporated, hosts have no password, can read anything.
*:Read Post:::spinnaker*
localhost:Read Post:::*
spinnaker.rhein.de:Read Post:::*
```

Damit darf jeder per NNRP (das ist das Protokoll, mit dem Newsreader mit dem Newsserver kommunizieren) auf die Newsgroups, deren Name mit spinnaker beginnt lesen und auch dorthinein posten. Alle anderen Newsgroups dürfen sie nicht lesen und auf meinem Rechner darf jeder jede Newsgroup lesen und darin schreiben.

Als nächstes ist /usr/local/lib/news/inn.conf wie folgt zu konfigurieren:

```
##  inn.conf -- inn configuration data
##  Format:
##      <parameter>:<whitespace><value>
##  Used by various programs and libinn.  The following parame-
ters are defined:
##      domain          Local domain, without leading period.
##      fromhost        What to put in the From line; default is FQDN
##                      of the local host.
##      moderatormailer Where to mail moderated postings, if not found
##                      in the moderators file; see moderators(5).
##      pathhost        What to put in the Path and Xref headers; default
##                      is FQDN of the local host.
##      organization    If $ORGANIZATION doesn't exist.  What to put in
##                      the Organization header if blank.
##      server          If $NNTPSERVER doesn't exist.  Local NNTP server
##                      host to connect to.
##
organization:   private site, Widdig, Germany
```

```
#server:        spinnaker.rhein.de
server:         localhost
```

Als nächstes das wichtigste File der INN-Konfiguration /usr/local/lib/news/newsfeeds, in dem festgelegt wird, welche Newsgroups an wen exportiert werden sollen.

```
##  newsfeeds - determine where Usenet articles get sent
##  Format:
##      site[/exclude,exclude...]\
##              :pattern,pattern...[/distrib,distrib...]\
##              :flag,flag...\
##              :param
##  Summary of flags:
##      <size           Article must be less then size bytes.
##      Aitems          Article checks -- d (must have Distribution header)
##                      p (don't check for site in Path header).
##      Bhigh/low       Internal buffer size before writing to output.
##      H[count]        Article must have less then count hops; default is 1.
##      Isize           Internal buffer size (if a file feed)
##      Nm              Only moderated groups that match the patterns.
##      Nu              Only unmoderated groups that match the patterns.
##      Ssize           Start spooling if more than size bytes get queued.
##      Ttype           Feed types -- f (file) m (funnel; param names the
##                      real entry) p (pipe to program) c (send to stdin
##                      channel of param's sub-process); x (like c, but
##                      handles commands on stdin).
##      Witems          What to write -- b (article bytesize) f (full path)
##                      g (first newsgroup) m (Message-ID) n (relative
##                      path) s (site that fed article) t (time received)
##                      * (names of funnel feed-in's or all sites that get
##                      the article) N (Newsgroups header) D (Distribution
##                      header) H (all headers) O (overview data) R
##                      (replication data).
##  Param field depends on T flag.  For Tf, relative paths are from the
##  out.going directory.  For Tp and Tc, it is a shell command to execute.
##  If a Tm refers to this entry (which will have its own T param) then "*"
##  is expanded to all the funnel sites that triggered this one.  Useful
##  for spawning one mail process, e.g.
##
##  This file is complicated -- see newsfeeds.5!

# ME zeigt an, welche Newsgroups hier bestellt werden sollen:
ME\
        :!*\
        ::

# Alles, was per FidoGate verschickt werden soll:
# - fido.* außer fido.junk, denn das ist eine Pseudo-Newsgroup in der
#   nur Artikel landen, die nicht ordentlich in Fidogate eingetragen
#   wurden,
# - ger.*, die Gernet-Newsgroups
```

```
# Wenn auch Internet ueber Fido (z.B. mittels fido.de) reinkommt, dann
# sind hier auch noch die entsprechenden Newsgroups (rec.*, comp.*,
# de.*, alt.*, gnu.*,...) einzutragen.
#
# Alle meine Uplinks haben den gleichen Namen (über die Datei
# /usr/local/lib/fidogate/hosts), naemlich flokiste.fido.de, daher ist
# dieser hier einzutragen, damit Mails, bei denen flokiste.fido.de im
# Pfad steht, nicht mehr dorthin exportiert werden (sonst gäbe es
# Dupes).
# Da ich Fido und Internet bei mir absolut auseinander halten möchte
# (die was über Fido reinkommt darf nicht ins Internet exportiert
# werden und umgekehrt), habe ich rhein hier ebenfalls als Alias
# eingetragen.
fidogate/flokiste.fido.de,rhein\
        :fido.*,!fido.junk,ger.*\
        :Tf,Wnb:

# Dies ist mein Internet (Usenet)-Feed.
# Hier werden die Areas comp.*,de.*,rec.*,alt.*,gnu.* exportiert.
# Damit hier nichts aus dem Fido rausgeht, steht flokiste.fido.de als
# Alias für rhein.
rhein/flokiste.fido.de\
        :comp.*,de.*,rec.*,alt.*,gnu.*\
        :Tf,Wfb:
```

Als nächstes ist das File /usr/local/lib/news/expire.ctl zu konfigurieren, welches angibt, wann welche Newsgroups expiren.

```
##  expire.ctl - expire control file
##  Format:
##      /remember/:<keep>
##      <patterns>:<modflag>:<keep>:<default>:<purge>
##  First line gives history retention; other lines specify expiration
##  for newsgroups.  Must have a "*:A:..." line which is the default.
##      <patterns>      wildmat-style patterns for the newsgroups
##      <modflag>       Pick one of M U A -- modifies pattern to be only
##                      moderated, unmoderated, or all groups
##      <keep>          Mininum number of days to keep article
##      <default>       Default number of days to keep the article
##      <purge>         Flush article after this many days
##  <keep>, <default>, and <purge> can be floating-point numbers or the
##  word "never."  Times are based on when received unless -p is used;
##  see expire.8

##  If article expires before 14 days, we still remember it for 14 days in
##  case we get offered it again.  Depending on what you use for the innd
##  -c flag and how paranoid you are about old news, you might want to
##  make this 28, 30, etc.
/remember/:30

# Alle Newsgroups für mindestens 3, maximal 90 und normalerweise 5
```

```
# Tage behalten (relativ zur Ankunftszeit)
*:A:3:5:90
*:U:3:5:90
# Moderierte Newsgroups länger behalten:
*:M:5:9:90

# Lokale Newsgroups normalerweise 90 Tage behalten:
spinnaker.*:A:3:90:90

# Deutsche Newsgroups 7 Tage behalten:
de.*:A:3:7:90
# de.comp.os.linux.* etwas länger:
de.comp.os.linux.*:A:3:14:90

# Moderierte c.o.l.* länger behalten:
comp.os.linux.*:M:5:90:90

# Fido etwas länger behalten:
fido*:A:3:9:90
ger*:A:3:8:90

# Junk schnell löschen:
fido.junk:A:3:4:90

# Und noch eine Ausnahme:
fido.ger.linux:A:3:24:90
```

Das kann man natürlich alles nach eigenem Geschmack ändern. Allerdings sollte man die Default-Zeiten (insbesondere für Fido) möglichst nicht weiter verkürzen, da die Messages meist lange unterwegs sind und ansonsten gar nicht erst ins Newssystem eingestellt werden, sondern direkt expiren.

Als nächstes ist das File /usr/local/lib/news/send-fidogate zu erstellen, das regelmäßig die zu verschickenden Fido-Echomails zusammenpackt. Einen Prototyp für dieses File findet man unter fidogate/config/inn/send-fidogate, allerdings muß man im Anfang des Files zwei kleinere Änderungen an den Pfaden vornehmen:

```
#! /bin/sh
#:ts=8
##  SH script to send batches to FIDOGATE

##  =()<. @<_PATH_SHELLVARS>@>()=
. /usr/local/lib/news/innshellvars

#
# FIDOGATE rfc2ftn
#
RFC2FTN="/usr/local/lib/fidogate/rfc2ftn -w Normal -b -n "
```

Damit die News exportiert werden können, muß ein Batch-Verzeichnis für sie existieren, das /var/spool/news/out.going heißt, news.news gehört und die Permissions 775 haben sollte. Im Gegensatz zu cnews dürfen in diesem Verzeichnis keine Unterverzeichnisse für die einzelnen Newsfeeds angelegt werden, denn INN verwendet nur ein File je Feed, welches er selber anlegt. Wer aber von cnews auf INN umsteigt, sollte die bestehenden Verzeichnisse löschen.

Da INN mit einem Dämon arbeitet, der immer läuft, muß dieser beim Booten gestartet werden. Hierzu fügt man am Ende von /etc/rc.d/rc.M vor dem Aufruf von rc.local folgendes ein:

```
  /usr/local/lib/news/etc/rc.news &
```

Für den ersten Start von INN kann dieses Skript (als root) auch manuell gestartet werden. Allerdings ist darauf zu achten, daß nicht versehentlich mehrere innd gleichzeitig laufen.

Bei einigen Distributionen ist standardmäßig ein anderer NNTP-Dämon (nntpd) für cnews in /etc/inetd.conf eingetragen. Dann kann der innd nicht gestartet werden, weil der TCP-Port bereits belegt ist. Daher muß die entsprechende Zeile in inetd.conf auskommentiert werden, was man dem inetd durch kill -1 mitteilen kann.

Weiterhin sind für den reibunslosen Betrieb von INN noch einige Einträge in der crontab des Users news notwendig:

```
SHELL=/bin/sh
MAILTO=roland
#
# INN:
#
13  *   * * *   /usr/local/lib/news/send-fidogate
17  *   * * *   /usr/local/lib/news/send-uucp rhein
13  21  * * *   /usr/bin/nice /usr/local/lib/news/bin/news.daily delayrm
48  21  * * *   /usr/local/lib/news/rnews -U
```

Hier werden einmal pro Stunde die Batches für alle Fidosysteme sowie für das UUCP-System rhein gepackt (ohne UUCP-Connection kann der Teil natürlich entfallen). Weiterhin wird einmal pro Tag aufgeräumt, und darüber per Mail informiert. Abschließend werden alle versehentlich liegengebliebenen Artikel verarbeitet.

Es bleibt noch das Erstellen der Newsgroups. Hierzu sowie für fast alle anderen Änderungen am Newssystem wird nur das eine Kommando ctlinnd verwendet. Es sollte immer nur vom User news aufgerufen werden und es ist im Verzeichnis /usr/local/lib/news/bin zu finden. Newsgroups werden dabei mittels ctlinnd newgroup fido.ger.linux angelegt (mit der entsprechenden Newsgroup natürlich). Dabei sind diverse weitere Optionen möglich, die man alle in der Manpage zu ctlinnd(8) nachlesen kann. Insbesondere sollten die Newsgroups control, junk und fido.junk angelegt werden, die Control-Messages (z.B. cancel), Artikel für nicht installierte Newsgroups und Artikel für nicht in /usr/local/lib/fidogate/areas eingetragene Areas auffangen.

Um im Newsreader die Beschreibungen der einzelnen Newsgroups zu finden, kann man weiterhin noch das File /usr/local/lib/news/newsgroups anlegen, welches in jeder Zeile den Namen einer Newgroups und dahinter (durch einen Tab abgetrennt) die Beschreibung der Newsgroup. Dabei stört es nicht, wenn nicht existierende Newsgroups eingetragen sind, Newsgroups doppelt eingetragen sind (es wird der erste Eintrag angezeigt) oder Newsgroups nicht eingetragen sind.

Da auch INN seine Log-Messages über syslog erstellt, sollte man folgende Einträge in /etc/syslog.conf vornehmen:

```
news.crit                                /var/log/inn/news.crit
news.err                                 /var/log/inn/news.err
news.notice                              /var/log/inn/news.notice
```

Weiterhin sollte man noch das Verzeichnis /var/log/inn anlegen.

Für weitere Informationen zu INN, die Umstellung eines cnews-Systems auf INN, Probleme etc. verweise ich auf die sehr gute INN-FAQ und die reichlich vorhandenen Manpages.

14 News/Echomail lesen und schreiben

Ich persönlich bevorzuge als Newsreader tin. Da dieser, wie alle anderen auch nicht für Fido konstruiert wurde, gibt es standardmäßig keine Möglichkeit, bei Echomails einen Adressaten anzugeben, was ja im Fido möglich und üblich ist. FidoGate ermöglicht das jedoch über eine zusätzliche Headerzeile X-Comment-To:. Selbige kann man manuell erstellen, wobei zu beachten ist, daß der Username in runden Klammern stehen muß. Es ist aber auch erlaubt, hier eine komplette Internet-Adresse inclusive Realname anzugeben.

Beispiele:

```
X-Comment-To: (Roland Rosenfeld)
X-Comment-To: rosenfel@uran.informatik.uni-bonn.de (Roland Rosenfeld)
```

Im Verzeichnis fidogate/contrib/news findet man jedoch auch Patches für tin, xrn und nn, die diese Zeile automatisch generieren, wenn man auf einen Artikel antwortet. Falls man tin nicht patchen möchte, kann man auch folgende Zeile in ~/.tin/tinrc eintragen:

```
news_quote_format=X-Comment-To: (%N)\n\n%F wrote in msg %M:
```

dann darf man allerdings nicht vergessen, bei jedem Follow-Up die überflüssige Leerzeile vor dem X-Comment-To: zu löschen.

Wenn Du schon eine der Beta-Versionen von tin einsetzt (ab tin-1.3beta-950824), dann kannst Du einfach folgendes in ~/.tin/attributes eintragen:

```
scope=*
auto_save_msg=OFF

scope=fido.*
x_comment_to=ON

scope=ger.*
x_comment_to=ON
```

und schon weden in den Fido- und Gernet-Echos X-Comment-To:-Header erstellt, ohne daß Du tin patchen mußt.

Als weitere Alternative bietet sich seit FidoGate 3.9.5 an, den Newsreader so zu konfigurieren, daß eine folgender Zeilen vorn in der Message steht, denn dann wird diese zur Ermittlung des Realnames verwendet:

```
user@do.main (User Name) writes:
User Name <user@do.main> writes:
user@do.main.writes:
user@do.main (User Name) wrote:
User Name <user@do.main> wrote:
user@do.main wrote:
In article <id@do.main> user@do.main writes:
In article <id@do.main>, user@do.main writes:
```

15 Mail/Netmail lesen und schreiben

Ich persönlich verwende hierzu elm. Eigentlich muß man sich dabei nicht um allzuviel kümmern, man adressiert nur halt mit *pfnz*-Notation. So wird aus

Roland Rosenfeld 2:2450/111.13

die Adresse

Roland_Rosenfeld@p13.f111.n2450.z2.fido.sub.de

oder gar

Roland_Rosenfeld@p13.f111.n2450.z2.fido.sub.de (Roland Rosenfeld)

und aus

Roland Rosenfeld 21:100/64.13

wird

Roland_Rosenfeld@p13.f64.n100.z21.ger.sub.de

Internet-Adressen kann man hingegen direkt angeben und für Mäuse gibt man die entsprechende Internet-Adresse an, also z.B. Vorname_Name@su.maus.de. Trotzdem wird die Mail über das Fido-Maus-Gate geschickt (es sei denn man hat in fidogate/src/config.h etwas anderes eingestellt).

Für Maus-Messages sollte die Datei /usr/local/lib/fidogate/maus immer auf dem neusten Stand sein, damit die verschiedenen Maus-Domains korrekt erkannt und konvertiert werden (nicht alle Mäuse sind unter .maus.de erreichbar, es gibt diverse Ausnahmen. Bei FidoGate ist aber immer eine aktuelle Version des Files dabei, welches man in fidogate/lib/maus findet.

Wichtig ist nun auch, daß man Mails an Tools (Areafix, FileMgr,...) schicken kann. Diese haben jedoch oft Probleme mit den von FidoGate erstellen Mails. Selbige haben nämlich meist in der ersten Zeile ein "* To: adresseßtehen, was manche der Tools so interpretieren, daß alle Areas ("*") anbestellt werden sollen. Will man also diese Zeile abstellen, so kann man mit elm im Menü **H**)eader mit **U**)ser-Defined-Header (vor dem Verschicken) eine zusätzliche Header-Zeile eingeben:

X-Flags: N

Weiterhin hat FileScan beispielsweise in einigen Versionen den Bug, daß er die Adresse nicht dem Header entnimmt, sondern der Message-ID. Selbige enthält bei FidoGate-Messages aber nicht die FTN-Adresse (Z:N/F.P), sondern die Internet-Adresse, so daß auch hier Schwierigkeiten auftauchen. Selbige lassen sich aber mit

X-Flags: M

beheben, denn dann wird eine Fido-übliche Message-ID generiert (für verschiedene Netze mit der jeweiligen Adresse).

Man kann die X-Flags natürlich auch kombinieren, ich verwende beispielsweise für alle Mails an irgendwelche Robots immer

X-Flags: MN

Will man Crashmails verschicken, so muß man zusätzlich "X-Flags: Cßetzen. Wird dabei an einen Point adressiert, so wird die Mail an dessen Boß gecrasht.

Auch FileAttaches sind möglich. Hierzu gibt man "X-Flags: Fän und gibt den Dateinamen mit Pfad(!) als Subject an.

Will man irgendwelche Flags für alle Mails setzen, dann hat man auch die Möglichkeit eine Datei ˜/.elm/elmheaders anzulegen, welche nur die entsprechende Headerzeile, also beispielsweise

X-Flags: MN

enthält. Dann wird diese Headerzeile in jeder Mail eingetragen.

Sollten die X-Flags wider erwarten bei Dir keine Wirkung zeigen, dann schau mal ins FidoGate-Logfile, denn dort bemerkt FidoGate dann gewöhnlich, daß der in /usr/local/lib/fidogate/config.common eingetragene Rechnername Hostname.Domain nicht mit dem Namen der Maschine, auf der FidoGate läuft, übereinstimmt. Genau das ist jedoch Vorraussetzung, damit die X-Flags akzeptiert werden, denn andernfalls könnte ja jeder per Netmail bei Dir Crashmails, Filerequests etc. absetzen.

16 Files requesten

Für Filerequests kann man das Perl-Skript ifreq verwenden, welches sich im Verzeichnis ifmail/misc/contrib befindet. Mit einigen DOS-Request-Prozessoren macht das noch Probleme, da an den Zeilenenden das Linefeed fehlt. Folgender Patch löst dieses Problem:

```
--- ifreq~      Sat Aug  6 14:41:28 1994
+++ ifreq       Sat Oct  8 12:05:09 1994
@@ -73,7 +73,7 @@
# open the flofile for appending
open(FLOFILE, ">>" . $reqfile) || die "can't open $reqfile";
while (@files) {
-                        print (FLOFILE shift(@files), "\n");
+                        print (FLOFILE shift(@files), "\r\n");
}
close(FLOFILE);
```

17 Fehlersuche

Sollte Das alles trotz meiner Anleitung noch nicht auf Anhieb funktionieren, so gibt es noch diverse Möglichkeiten, einen Fehler aufzuspüren.

Wichtigstes Tool hierzu ist das Programm ifstat, welches anzeigt, welche Pakete für welche Adresse bereit liegen. Hier sollte man sich nicht wundern, wenn die Pakete nach dem Start von ifpoll plötzlich geschrumpft sind, das liegt nämlich daran, daß die Mails dann zu ZIP-Files gepackt wurden.

17.1 Mail wird nicht verarbeitet

Hier ist es sinnvoll, das Debug-Level von rfc2ftn in /etc/smail/transports bzw. im /etc/sendmail.cf hochzusetzen. Hierzu gibt man im entsprechenden Netz den zusätzlichen Parameter -v bei rfc2ftn an (ggf. auch mehrfach, dann wird das Debug-Level entsprechend höher, ich verwende meist -vvvvvvvvvvvvvvvvvvvvv oder so). Um die Debug-Meldungen dann wirklich in die Hand zu bekommen, schicke ich die Mail einfach mit einer Kopie an eine nicht existente Adresse und bekomme dann den Rückläufer mit sämtlichen Debug-Meldungen. Kommt der Rückläufer überhaupt nicht, so kann es durchaus sein, daß irgendwelche Permissions nicht stimmen und das System einfach wartet (man sieht das daran, daß noch sendmail und rfc2ftn als Prozesse laufen). Dann findet sich das aktuelle Debug-File in /var/spool/smail/msglog. Unter /var/spool/smail findet man dann auch die liegengebliebenen Messages. Bei Verwendung von Sendmail werden die Log-Meldungen dagegen per syslog ins Syslogfile geschrieben, während die liegengebliebenen Messages in /var/spool/mqueue landen.

17.2 News werden nicht verarbeitet

Analog zum Vorgehen bei Mail kann man auch bei News verfahren. Hier gehe ich gewöhnlich so vor, daß ich in /usr/lib/newsbin/batch/viafido (cnews) bzw. /usr/local/lib/news/send-fidogate (INN) die entsprechenden -v's einbaue und dann an die entsprechende Zeile ein 2> /tmp/fgate.out anhänge, was dazu führt, daß stderr nach /tmp/fgate.out umgeleitet wird, das ich mir dann anschauen kann.

Ansonsten sollte man bei cnews zwischen dem Aufruf von newsrun und run-batch nachsehen, ob der Artikel in /var/spool/news/out.going/*/togo aufgelistet wurde und dort anschließend auch wieder gelöscht wurde. Manchmal (je nach cnews-Version) werden die Artikel auch in /var/spool/news/in.coming zwischengelagert.

Bei INN sollten die Artikel sofort ins Newssystem aufgenommen werden, und damit auch direkt im File /var/spool/news/out.going/* erscheinen. Nach dem send-fidogate sollten sie dort verschwinden und im Fido-Outbound landen.

Weiterhin sollte man sich alle vorhandenen Logfiles ansehen:

smail

/var/spool/smail/log/logfile

Sendmail

/var/log/mail

cnews

/var/lib/news/log.* /var/lib/news/batchlog.* /var/lib/news/errlog.*

INN

/usr/local/lib/news/log /usr/local/lib/news/errlog /var/log/inn/*

fidogate

/var/log/fnet/log

ifcico

/var/log/fnet/ifmail /var/log/fnet/iflog /var/log/fnet/ifdebug

Um die Korrektheit erstellter Pakete zu erkunden oder Probleme in bestehenden Paketen zu finden, enthält FidoGate das Programm pktdebug. Braucht man mehr Informationen, so sollte man sich das DOS-Programm Inspect besorgen, welches problemlos in der Dosemu läuft und wenn man die Verzeichnisse entsprechend freigibt und die Dosemu dann als User uucp startet, dann kann man die .pkt-Files und Archive sogar verändern.

Hängt CNews komplett, so bleibt einem nur ein Debugging der diversen cnews-Shell-Scripts. Hierzu sollte man in das jeweilige Skript ein "set -xväm Anfang einbauen, welches genau ausgibt, welche Zeilen des Skripts in welcher Reihenfolge und mit welchen Parametern aufgerufen werden. Für weitere Informationen dazu empfehle ich die Manpage der bash.

Falls nur das Posten von News-Artikeln nicht klappt, kann man auch mal versuchen, mit Pnews aus dem Paket trn direkt zu posten. Selbiges ist nämlich im Gegensatz zu tin ein einfaches Shell-Skript, bei dem die Fehlermeldungen von cnews immerhin so lange stehen bleiben, bis man sie gelesen hat (bei tin werden die direkt wieder vom Menü überschrieben).

18 Unterstützung

Bei dieser HOWTO wurde ich durch viele Kommentare und Verbesserungen unterstützt. Besonders sind hier zu nennen:

- andij@andi.tricbbs.fn.sub.org (Andreas Jellinghaus)

- uli@corno.cologne.de (Ulrich Villers) 2:2450/115.3
- martin@erde.GUN.de (Martin Seine) 2:2448/413.100
- piet@smallchange.rhein.de (Peter Theisohn) 2:2450/30
- mg@genyosha.in-chemnitz.de (Mike Gaertner) 2:249/5060.17
- mbudde@sms.antar.com (Marco Budde) 2:240/6298.5

Ergänzungen, Fehler-Reports etc. werde ich gern einarbeiten.