SelfLinux-0.10.0

INN

Autor: Steffen Dettmer (*steffen@dett.de*)
Formatierung: Matthias Hagedorn (*matthias.hagedorn@selflinux.org*)

Inhaltsverzeichnis

1 Einleitung

Dieses Dokument beschreibt den Betrieb eines **echten** Newsservers mit der Software ***INN***. Es richtet sich an angehende News-Administratoren, nicht jedoch an Heimanwendungen. Für Heimanwendungen verwendet man am besten einen bereits bestehenden Newsserver oder das Programm ***leafnode***. Der Betrieb eines **echten** Newsservers ist sehr aufwendig, kompliziert und erfordert bei größeren, sinnvollen Installationen, dass mindestens ein Newsserver angepaßt wird, damit der eigene Server News zugeschickt bekommt (das ist ein grundlegender Unterschied zum eMail-Versand, wo ja jeder einfach einen eigenen Server betreiben kann).

Der Betrieb eines **echten** Newsserver lohnt sich, wenn man viele Clienten bedienen möchte (das heißt, weit mehr als 10), oder viele Artikel verwalten muß (das heißt, weit mehr als 10.000). Für kleinere Installationen sollte entweder ein fremder Newsserver verwendet werden, oder das Programm ***leafnode*** verwendet werden. Nur, wenn wirklich Bedarf besteht, sollte man sich einen **echten** Newsserver leisten, denn Einrichtung und richtiger Betrieb sind aufwendig.

1.1 News

Mit News bezeichnet man kurze Textdateien, von denen ständig welche geschrieben werden (in einigen Fällen sind das auch binäre Dateien). Diese werden an viele Benutzer verteilt. Diese können dann auch darauf antworten. Da es sehr viele solcher Artikel gibt, hat man diese sogenannten Gruppen zugeordnet. So gibt es eine Gruppe **comp.os.linux**, die sich mit Linux beschäftigt und viele andere, bis hin zu **de.alt.freieliebe**.

Um der Flut von Artikeln Herr zu werden, werden diese nach einiger Zeit, zum Beispiel nach einem Monat, gelöscht. Es handelt sich schließlich um **News** - also Neuigkeiten.

1.2 Genereller Überblick

Ein Benutzer schreibt einen solchen Artikel mit einem Programm, zum Beispiel mit *Netscape*. Das Programm sendet (postet) den Artikel dann zu einem bestimmten Server. Dieser verteilt den Artikel dann an weitere Server und diese verteilen den Artikel dann auch wieder weiter. Über irgendwelche Wege **kennt** jeder Server indirekt jeden anderen (sonst würde es **Inseln** geben). Es kann somit auch einige Zeit dauern, bis der Artikel überall zur Verfügung steht, von allen gelesen und beantwortet werden kann. Heutzutage sind jedoch die Verbindungen so schnell, dass ein Artikel in Sekunden fast alle Teile der Welt erreicht hat. Jedoch gibt es auch Server, die zum Beispiel aus Kostengründen Artikel nur nachts übertragen.

1.3 Usenet

Mit Usenet bezeichnet man alle Server, die News-Artikel untereinander austauschen. Das Usenet ist eine dezentrale Anordnung von Newsservern. Das Usenet enthält Artikel, die einer Gruppe zugehören. Die Gruppen sind dabei in (mehrere) Hierarchien gegliedert, eine davon heißt z.B. **de** und beinhaltet u.a. **de.answers**, **de.test** und (Sub-) Hierarchien wie z.B. **de.comp** (mit **de.comp.os** usw.). Ein Newsserver hat eine bestimmte Menge von Newsgroups, die er verwaltet/kennt.

Ein Client, auch Newsreader genannt (z.B. *tin*, *Netscape*), kann nun über den NNTP Port auf den Newsserver connecten, sich eine Liste der Newsgroups geben lassen, sich einzelne Artikel herunterladen oder senden (**posten**). Diese Clienten werden hier als Newsreader (oder kurz Reader) bezeichnet, mit Server wird im folgenden ein Newsserver bezeichnet.

2 Verteilung von News

2.1 Verteilung vieler Artikel

Um die Verteilung zu ermöglichen, stehen die Newsserver untereinander in Verbindung. Da in der Regel viele Artikel übertragen werden und aus Effizienzgründen keine doppelt übertragen werden sollen, wird ein spezielles Verfahren verwendet. Die Server pflegen Listen, in denen die zu übertragenden Artikel stehen, wird eine Übertragung durchgeführt, wird die Liste **abgearbeitet**, so dass mit einer Verbindung alle Artikel übertragen werden können. Diese Listen nennt man **Feedlisten**. Hat ein Newsserver einen neuen Artikel empfangen, so trägt er die Nummer des Artikels in die Feedlisten ein. Ein Artikel trägt in seinem Header Steuerinformationen, wie z.B. eine Pfadliste, in der die Newsserver stehen, die diesen Artikel übertragen haben (jeder Server fügt seinen Namen am Anfang der Liste ein, das Verfahren ist analog zur Pfadadresse einer eMail aus UUCP-Zeiten).

2.2 Feeden und Posten

Artikel, die in der Feedliste stehen, sollen also später gefeeded werden. Dies darf man nicht mit dem posten verwechseln. Clients posten, Newsserver feeden. Feeden ist auf den ersten Blick zwar sehr ähnlich, jedoch auf Protokollebene anders. Feeden ist für große Mengen an Artikeln optimiert. Posten funktioniert einfacher. Beim Posten geht der Newsserver davon aus, dass der Artikel neu ist. Gepostete Artikel werden gegebenenfalls um eine ID ergänzt. Gefeedete Artikel müssen immer bereits eine ID besitzen.

2.3 Feedlisten

Soll ein Artikel in die Feedliste für einen Zielserver eingetragen werden, ist das natürlich nur nötig, wenn der Name des Zielservers noch nicht in der Pfadliste steht (denn anderenfalls hat er den Artikel ja bereits gesehen und gespeichert). Der Artikel wird also nur in die Feedlisten der Server eingetragen, die den Artikel noch nicht weitergeleitet haben. Nun kann aber ein Zielserver diesen Artikel von einem anderen Server erhalten haben. Um sinnlosen Datenverkehr zu unterbinden, hat deshalb jeder Artikel eine weltweit eindeutige ID (genauer gesagt, muß die ID eindeutig im gesammten Usenet sein, also eindeutig in allen bekannten Universen). Connected ein Server einen anderen, um ihm Artikel zu schicken, sendet er erst einmal eine Liste von IDs, die er hat. Diese Liste wird aus den Feedlisten erstellt. Der andere Server vergleicht diese mit seinem Datenbestand. Artikel mit unbekannten IDs werden nun übertragen. Dieser Vorgang läuft ebenfalls oft über den NNTP Port ab (deswegen kann es Schwierigkeiten geben, wenn vom Feeder aus versucht wird, eine Client-Verbindung aufzubauen, da der Server häufig hier nur einen Server erwartet, und sich entsprechend **[falsch]** verhält).

2.4 Die eigentliche Übertragung

In regelmäßigen Zeitabständen wird ein `cron`-job gestartet, der die Feedlisten überträgt. Alternativ dazu kann man das auch über ein Programm erledigen, was sich wie ein Client verhält und diese Artikel postet (also nicht feeded!), dabei müssen die Artikel allerdings etwas modifiziert werden, da ein Client einige Headerfelder nicht setzen darf (z.B. NNTP-Posting-Host, Xref, X-Trace, X-Complaints-To, NNTP-Posting-Date), da diese vom Newsserver gesetzt werden (dazu kann man ein Filterscript verwenden).

Um nicht in der Datenflut zu ersticken, akzeptiert ein Newsserver nur einige Gruppen (das können auch einige hundert sein). Das sollte bereits beim Erstellen der Feedlisten berücksichtigt werden, man muß schließlich keine Artikel in die Listen eintragen, die sowieso nicht akzeptiert werden. Auch möchte man evtl. einige **private** Gruppen führen, die ebenfalls nicht gefeeded werden sollen.

3 Neue Gruppe einrichten

3.1 Überblick

Eine neue Gruppe wird in drei Schritten eingerichtet. Zum Einen muß der Feeder (i.d.R. der ISP) die neue Gruppe in seine Feedliste aufnehmen, damit der Feeder neue Artikel in die Feedliste einträgt (das macht er nur, wenn der Empfänger die Gruppe überhaupt haben will). Dazu ist es natürlich notwendig, dass der Feeder der Gruppe selbst gefeeded bekommt und neue Artikel entsprechend **zurückfeeden** kann, sonst erhält man natürlich keine Artikel. Einfacher gesagt, man kann keine Gruppen bekommen, die er selbst nicht hat.

Zum Anderen muß der lokale Newsserver diese Gruppe bedienen, d.h. diese Gruppe muß **active** sein, ansonsten würde der lokale Newsserver die gefeedeten Artikel als **unwanted** - **ungewollt** - ablehnen und in der **Pseudo**-Gruppe **junk** - **Müll** ablegen (diese Gruppe hat nur einen geringe Lebensdauer, oft werden die Artikel nach einem Tag bereits gelöscht).

Dann muß als drittes der lokale Newsserver diese Gruppe auch an den **Feeder feeden**, damit eventuelle Antworten ins Usenet gelangen. Ansonsten würde man ja keine Antwort auf die Artikel bekommen.

3.2 Vorgehen

Man beginnt sicherheitshalber immer mit dem zweiten Schritt (damit keinesfalls Artikel in **junk** landen und **verloren** gehen, denn ein manuelles Übertragen von **junk** in eine Gruppe ist mindestens mühsam!). Dazu richtet man mit dem Dienstprogramm **`ctlinnd(1m)`** eine neue Gruppe ein. Dabei sorgt `ctlinnd` hier **nur** für eine ordnungsgemäße Eintragung in **active** (und macht diese Änderung dem innd bekannt). Der Aufruf lautet dabei z.B.:

```
root@linux ~/ # ctlinnd newgroup de.talk.jokes
```

Dabei können noch weitere Argumente angegeben werden, dass ist hier ein minimalistisches Beispiel. Dann richtet man am besten erstmal gleich den eigenen Feed ein, damit Postings dann auch zurückgehen können (bzw. an weitere Server verteilt werden). Dazu muß man dann die Datei `newsfeeds` (meist `/etc/news/newsfeeds`) anpassen, wenn nötig (es kann z.B. sein, das alle Gruppen, evtl. bis auf Ausnahmen, gefeeded werden).

Nun muß der Feeder den newsfeed für unseren Newsserver anpassen.

Normalerweise bekommt man vom Feeder keine alten Artikel. Das heißt, es kann eine Weile dauern, bis neue Gruppen auch wirklich Artikel beinhalten!

4 Gruppen entfernen

Eine Gruppe loszuwerden, läuft entsprechend rückwärts. Um die Benutzer nicht zu ärgern, sollte sowas immer angekündigt und abgesprochen werden. Zu diesem Zwecke kann man z.B. eine Gruppe `local.users` oder so verwenden. Zuerst sollte dann der Feeder die Gruppe nicht mehr feeden. Es kommen damit keine neuen Artikel, aber es kann noch gepostet werden. Dann kann man die Gruppe einfach **expiren** lassen - irgendwann ist sie leer. Spätestens dann sollte man dann die Gruppe aus dem feed nehmen, und dann auch entfernen.

5 Newsserver ohne Feeder

Wenn man keinen Newsserver zur Verfügung hat, der den eigenen Newsserver feedet, also man keinen Newsadministrator überreden kann, für seinen eigenen Server Feedlisten zu pflegen, kann man auch einen Newsserver betreiben, der sich wie ein Client verhält.

Das bedeutet, er holt die Artikel wie ein Client ab und verteilt Artikel nicht über Feeds, sondern postet diese selbst auch wieder. Mischformen sind natürlich auch möglich.

Man kann hier `suck` und `rpost` verwenden.

5.1 Artikel einspeisen

`Suck` verwendet man wie folgt:

Suck

```
#!/bin/sh

suck <newsserver> -c -bi /var/spool/news/batch -dt /var/spool/news \
        -dm /var/spool/news/Msgs -dd /var/spool/news

/usr/lib/news/bin/innxmit localhost /var/spool/news/batch
```

Die Pfade müssen natürlich den Gegebenheiten angepaßt werden.

5.2 Artikel weiterschicken

Beim Posten müssen noch einige Infos aus dem Header gefiltert werden, da der andere Server keine Newsserver Einträge in den Artikeln sehen möchte (viele Newsserver werfen solche Artikel gleich in **junk**).

Hier ein Beispiel für die Verwendung von Post:

Beispiel für die Verwendung von Post

```
#!/bin/bash
mv /var/spool/news/out.going/newssrv3 \
        /var/spool/news/out.going/newssrv3.new
/usr/lib/news/bin/ctlinnd flush newssrv3
/usr/bin/rpost newssrv3.bedi.net -d -b \
        /var/spool/news/out.going/newssrv3.new \
        -p /var/spool/news \
        -f /etc/news/post.filter  \$\$o=/tmp/filtered_msg \
        \$\$i /tmp/filtered_msg
```

Das ist hier `quick-and-dirty` (für Testumgebungen z.B.). Hier wird `out.going` sicherheitshalber für die Dauer des `rpost` Aufrufes umbenannt, damit keine Artikel übergangen oder doppelt gepostet werden.

5.2.1 Artikel zum Versand aufbereiten

Ein Beispiel-Filter für das Entfernen der Header und Newsservereinträge (`/etc/news/post.filter`) könnte so aussehen:

Beispiel-Filter für das Entfernen der Header und Newsservereinträge

```
#!/bin/sh

PATH=/usr/local/bin:/usr/bin:/bin:/usr/lib/news/bin

PERLCMD='/^(NNTP-Posting-Host|Xref|X-Trace|\
X-Complaints-To|NNTP-Posting-Date)/ or print'

INFILE=$1
OUTFILE=$2

if [ -f ${INFILE} ]; then
        cat ${INFILE} | perl -ne "${PERLCMD}" > ${OUTFILE}

        if [ $? -ne 0 ]; then
                echo "Error"
                exit -1
        fi
else
        echo "$1 does not exist"
        exit -1
fi
```

5.3 Automatisierung

Beide Scripte kann man über einen `Wrapper` (News eXchanger) starten:

Wrapper (News eXchanger)

```
#!/usr/bin/bash

/etc/news/post.sh >> /var/log/news/post_log 2>&1
/etc/news/suck.sh >> /var/log/news/suck_log 2>&1
```

Hier werden noch Logdateien geschrieben.

Das kann man dann per Cron alle 5 Minuten ausführen lassen:

per Cron alle 5 Minuten ausführen lassen

```
*/5 * * * * /etc/news/NX.sh
```

5.4 Erweiterungen

Das ist ein kleines Grundgerüst, auf dem man aufbauen kann. Es ist damit zu rechnen, dass weitere Dinge angepaßt werden müssen; leider verhalten sich Newsserver nicht einheitlich. Das ist meistens neben dem Verwalten der Datenmengen die größte Schwierigkeit beim Administrieren von **richtigen** Newsservern.

6 Echte Feeds

6.1 Feeds für andere Server konfigurieren

Hier nun ein paar Infos, wie man INN verrät, wie die Feedlisten auszusehen haben. Als erstes muß INN natürlich wissen, für welchen Newsserver überhaupt Feedlisten erstellt werden müssen und welche Gruppen enthalten sein sollen. Das erledigt man in der Datei `newsfeeds`. Das Format sieht wie folgt aus:

Mit `#` beginnende Zeilen sind Kommentare. Ein Eintrag ist eine Zeile lang, kann sich über mehrere erstrecken, wenn als letztes Zeichen ein `\` kommt. Die Felder eines Eintrags werden durch `:` getrennt und bedeuten:

Datei newsfeeds

```
Newsserver/Alias                \
        :Gruppenpattern/Distribution \
        :flag,flag                  \
        :param
```

Jeder Eintrag entspricht einer Feedliste und damit einem Server, dem man Artikel feeden möchte.

Ist Newsserver bereits im Pfad, so wird ein Artikel nicht gefeeded. Unter Umständen stimmen Newsserver **FQDN** und dessen Pfadeintrag nicht überein (ein Server kann ja mehrere [Alias-] Namen haben). Diese kann (und sollte) man dann alle bei Alias aufführen.

6.2 Einen Feed konfigurieren

Was genau der Server gefeeded bekommen soll, wird über die Optionen des Feedlisten-Eintragen eingestellt.

Gruppenpattern bestimmt die Gruppen, die gefeeded werden sollen. Dabei ist als Wildcard `*` erlaubt (was eine beliebige Anzahl von ebenfalls beliebigen Zeichen darstellt).
Gruppen, die ausgeschlossen werden sollen, kann man hinter `!` definieren. Zusätzlich kann man eine Distributionsliste angeben (incl. Ausnahmen).

Die beiden wichtigesten Flags sind:

`T<type> mit <type> <f|p|...>`

- `f`: file
- `p`: programm

(hier wird nur `Tf` erklärt: die Feedliste ist eine normale Datei)

`W<items> mit <items> {m,n,H...}`

- `m`: Message-ID
- `n`: Pfadname
- `H`: Alle Header

Zum späteren Senden via NNTP mittels nntpsend wird die Message-ID und der Pfad benötigt (der Rest steht ja im Artikel unter Pfad), also **Wnm**

6.3 Feedlisten zusammenstellen

Aus technischen Gründen muß erst ein spezieller Feed für den Server selbst eingerichtet werden. Dieser enthält in der Regel alles. Ein entsprechende Eintrag sieht so aus:

```
ME\
        :*,!control,!junk\
        ::
```

(`control` und `junk` interessieren nicht)

Ein Beispiel für einen Feed zum ISP könnte so aussehen:

```
newsfeed\
        :*,!control,!junk,!local.*\
        :Tf,Wnm:
```

Das bedeutet: Es soll NNTP zum Feeden verwendet werden [UUCP ist so gut wie tot]. Dabei sollen alle Gruppen, außer `control`, `junk`, und `local.*` übertragen werden. `nntpsend` erwartet ein File (`Tf`) mit dem Format Message-ID und Pfad (`Wnm`).

Overview- und Crosspostsdata kann so erzeugt werden:

Overview- und Crosspostsdata

```
overview!:*:Tc,WO:/usr/bin/overchan
crosspost:*:Tc,Ap,WR:/usr/bin/crosspost
```

Diese Datei `newsfeed` gehört ins Verzeichnis `/etc/news`.

6.4 Feeds durchführen

Um diese Feeds auch durchzuführen, muß ein entsprechender `cronjob` laufen. Man kann ihn nachts laufen lassen, nur dann erhält man frühestens einen Tag später eine Antwort, laufen beide Feeds (also Server zum ISP und ISP zum Server) einmalig nachts, kann eine Antwort frühestens zwei Tage später erscheinen. Für ein schnelleres Verhalten sollte man nntpsend z.B. alle 10 Minuten starten. Dazu dient unter Linux z.B. folgender Eintrag:

alle 10 Minuten starten

```
*/10 * * * * /usr/lib/news/bin/nntpsend
```

6.5 Feeds testen

Zu Testzwecken kann man `nntpsend` natürlich auch manuell starten. Ein (leider selten nützliches) `logfile` liegt unter `/var/log/news/nntpsend.log`.
Das sieht z.B. so aus:

/var/log/news/nntpsend.log

```
nntpsend: [5694] start
nntpsend: [5694] stop
nntpsend: [5694:5715] begin newssrv2 Wed Dec 15 20:06:03 MET 1999
nntpsend: [5694:5715] innxmit -a newssrv2 ...
nntpsend: [5694:5715] end newssrv2 Wed Dec 15 20:06:04 MET 1999
```

Im Falle, dass Artikel gesendet wurden (`nntpsend` verwendet `innxmit` nur in diesem Fall). Ansonsten besteht es nur aus start/stop Zeilen (keine Artikel übertragen).

Der `innd` auf der **anderen** Seite, also der Empfänger, loggt diese Vorgänge auch (das wird dabei indirekt von `nntpsend` gesteuert), das sieht so aus:

/var/log/news/nntpsend.log

```
Dec 15 20:05:00 newssrv1 innd: newssrv3 flush
Dec 15 20:05:00 newssrv1 innd: newssrv3 opened newssrv3:15:file
Dec 15 20:05:00 newssrv1 innd: newssrv3 closed
Dec 15 20:05:02 newssrv1 innd: localhost connected 18 streaming allowed
Dec 15 20:05:02 newssrv1 innd: localhost:18 NCmode "mode stream" received
Dec 15 20:05:04 newssrv1 innxmit[5692]: localhost stats offered 737 accepted 1 refused 736
rejected 0
Dec 15 20:05:04 newssrv1 innxmit[5692]: localhost times user 0.070 system 0.110 elapsed
2.337
Dec 15 20:05:04 newssrv1 innd: localhost:18 closed seconds 2 accepted 1 refused 736 rejected
0
```

Hier sieht man eine nette Fehlkonfiguration: der Newsserver hat von 737 Artikeln 736 abgelehnt (vermutlich **hat** er diese Gruppe nicht, bzw. möchte sie nicht, weil er sie nicht bedient). Einen hat er jedoch akzeptiert (und keinen rejected. Rejected wird, wenn er ihn eigentlich möchte, aber er z.B. meint, es handle sich um **Spam**, oder der Artikel hat falsche Struktur oder sowas, beispielsweise die oben erwähnten Newsserverheader). Das ganze hat zwei Sekunden gedauert.

6.6 Beliebte Fehlersituationen bei Feeds

Es gibt weitere häufige Fehlermeldungen, die nicht immer klar verständlich sind. Hier einige wichtige Beispiele.

6.6.1 Der eigene Server läuft nicht

Situation: Der eigene Server ist down, `nntpsend` bekommt keine Daten zum versenden. Das sieht dann so aus:

/var/log/news/nntpsend.log

```
nntpsend: [2681] start
Can't send "flush" command (dead server failure) No such process.
nntpsend: file /var/news/storage/out.going/newsfeed for newsfeed not found
nntpsend: skipping newsfeed via newsfeed
```

6.6.2 Der fremde Server läuft nicht

Situation: Der fremde Server ist down, innxmit kann ihn nicht erreichen:

/var/log/news/nntpsend.log

```
nntpsend: [3888:3909] innxmit -a newsfeed ...
Can't connect to newsfeed, Connection refused
```

6.6.3 Probleme mit den Gruppen

In der Praxis kommt es immer wieder mal vor, dass der Server plötzlich hunderte von Artikel ablehnt.

Ungewollte newsgroups werden in `unwanted.log` erfaßt:

unwanted.log

```
130 newsgroup de.comp.lang.delphi.datenbanken
3 newsgroup de.comp.datenbanken.misc
```

Dies kann kommen, wenn uns Datenbanken nicht mehr interessieren und die Gruppe nicht mehr geführt wird. Der Feeder sollte seine Feedlisten anpassen; in der Praxis kann sowas leider oft lange dauern, da die Administratoren selten Zeit haben.

6.7 Selektion von Gruppen

Bei der Auswahl der erwünschten Newsgroups, die vom ISP gefeeded werden sollen, ist sehr selektiv auszuwählen. Keinesfalls darf der Fehler gemacht werden, z.B. `de.*` oder `alt.*` haben zu wollen (es sei denn, Sie sind glücklicher Besitzer einer gelangweilten E3 Anbindung - das sind 34 Mbit Bandbreite - und haben einen Kleiderschrank voller Geld für die Trafficgebühren). Lektüre entsprechender **Selbstzweck**-Newsgroups (`news.admin.technical` z.B.) ist hilfreich, daher auch das folgende Zitat:

Zitat

```
> curt@kcwc.com (Curt Welch) wrote:
> thebyte@san.rr.com (Daniel Trewella) wrote:
> > Our news server currently hogs about 2mb/s on our 6mb/s backbone.
> > (Can
> > you say "ouch"?) Is there a way to limit the bandwith that our news
> > server is using?
> >
> > The system is running FreeBSD 3.0-RELEASE and INN 2.1.
>
> Are you talking about your incoming feeds?  You're lucky it's only
> 2mb/s.
> A full feed is more like 8mb/s these days.
>
> The only good way to deal with the size of your incomming feed is to
> change what your feeds are sending you.
```

Überschlagen wir mal ganz grob: 8mb/s sind 1Mbyte/s. Der Tag hat 60*60*24 Sekunden, macht etwa 84 GB am Tag, und 2,5 TB (TeraByte) im Monat. Nehmen wir mal an, 1 GB würde EUR 10,- kosten (das ist ein in etwa realistischer Preis), dann kämen wir auf etwa 25.000 Euro im Monat!! Dies setzt natürlich voraus, das die Bandbreite im Tagesdurchschnitt bei 8Mbit liegt...

7 Cron Jobs

7.1 Für Feeds

Hier ein Cronjob zum Durchführen der Feeds:

nntpsend z.B. alle 10 Minuten starten. Dazu dient unter Linux z.B.:

alle 10 Minuten starten

```
*/10 * * * * /usr/lib/news/bin/nntpsend
```

7.2 Nächtliches Aufräumen

Diese Jobs können viel Rechen- und Plattenleistung verbrauchen und sollten daher nachts ausgeführt werden.

Cronjob

```
0  * * * * /usr/lib/news/bin/news.daily expireover lowmark
24 3 * * 0 /usr/lib/news/bin/expireover -a -v
24 1 1 * * /usr/lib/news/bin/makehistory -buv
```

7.3 Löschen alter Artikel

Meist reicht es aus, `news.daily` regelmäßig auszuführen. Beispielsweise könnte man es dreimal täglich starten (bei großen Servern vielleicht auch nur einmal täglich, daher der Name des Scriptes; einmal nachts beim Aufräumen reicht meistens):

news.daily

```
5 9,15,21 * * * /usr/bin/news.daily delayrm \
expireover norenumber nomail nologs >/dev/null 26gt;&1
```

8 Debugging, Inbetriebnahme, Probleme

8.1 1.1 Informationsquellen

Die beiden wichtigsten Quellen für Informationen bzw. Fehlerbeschreibungen sind eMails, die an **news** bzw. **root** gemailt werden, und die Logfiles. Die wichtigesten logfiles werden dabei meistens via `syslog` verwaltet. Der Standard-Pfad ist `/var/log/*` (einschließlích `/var/log/news/*`) unter Linux. Diese lassen sich natürlich anpassen, wenn man möchte. Eine kleine Inkonsistenz tritt hierbei auf: Linux-Syslog legt manchmal alle news Meldungen in `news.notice` ab. Diese beiden Dateien sind immer die erste Anlaufstelle bei

Problemen.

8.2 Startprobleme

Wenn innd überhaupt nicht startet, kann es z.B. an einem Syntaxfehler in der Konfiguration liegen. Das Programm `inncheck` hilft, derartige Fehler zu finden. Im Normalfall sollte es keine Ausgabe machen. Ausgewiesene Fehler sind entsprechend zu korrigieren, klar. Wenn es gar nicht klappt, kann man evtl. durch ein System trace (mit `strace -f /pfad/innd` - Pfade ergänzen!) Hinweise bekommen. Die Interpretation erfordert allerdings recht intensive Systemkenntnisse.

8.3 Zugriffsprobleme

Wenn `innd` läuft, kann man prüfen, ob ein Client Zugriff bekommt:

```
root@linux ~/ # telnet newshost 119
```

Wenn man einen **connect** bekommt, kann man z.B. das Kommando `LIST` probieren (danach dann `QUIT`). Man sollte die Gruppenliste erhalten. Ein Feedtest macht man z.B. mit dem Kommando

```
root@linux ~/ # ihave <xxx@test.de>
```

Einem Clienten sollte dann geantwortet werden **480 Transfer permission denied**, einem Feeder **335** und dem Warten auf Eingaben (Ende mit **<CR>"."<CR>** (also einem Punkt als einziges Zeichen auf einer Zeile, wie auch bei SMTP). Die zu erwartende Fehlermeldung: **437 No colon-space in header**. Je nach Version und Typ werden Abweichungen vorhanden sein.

8.4 Probleme mit Feedern

Wird ein Feed nicht als Feed erkannt, sollte als erstes geprüft werden, ob der **FQDN** (Systemname des Servers) übereinstimmt. Dazu kann man z.B. eine Verbindung aufbauen und dann mit

```
root@linux ~/ # netstat -a|grep nntp
```

schauen, wie der Name ist, bzw.

```
root@linux ~/ # netstat -an|grep 119
```

mit nachfolgendem `nslookup`. Ist der Name bestimmt, können die Einträge in `hosts.nntp` überprüft werden.

8.5 Abgelehnte Artikel

Gründe für rejects oder refuses zu finden, kann aufwendig werden. Es ist zu beachten, dass auch bereits gespoolte Artikel abgelehnt werden können. Unerwünschte Gruppen werden sowieso abgelehnt. Ein Blick ins `active`-File ist jedenfalls immer ratsam.

8.6 Probleme mit Clienten

Falls sich Clienten beschweren, nicht mehr connecten zu können, sollten als erstes die Prozesse des Newsservers neu gestartet werden:

```
root@linux ~/ # /etc/init.d/inn stop
root@linux ~/ # /etc/init.d/inn start
```

Dann kann z.B. mit:

```
root@linux ~/ # telnet <newsserver> 119
```

ein Test gemacht werden.
In jedem Falle sind die Logfiles zu analysieren. Das sollte von Zeit zu Zeit auch gemacht werden, wenn augenscheinlich alles funktioniert, um evtl. Fehler früh zu erkennen.

8.7 Vollaufen eines Dateisystems

Einer der schlimmsten anzunehmenden Fehler ist ein Vollaufen eines Dateisystems, insbesondere des **root**-Dateisystems. In einem solchen Fall wird nicht nur der Newsbetrieb gestört, sondern fast alle Serverfunktionen. Außerdem können dadurch undefinierte Zustände auftreten, die sich nur schwer erkennen und beheben lassen (z.B. existierende, aber leere Dateien). Dateien, die regelmäßig **überarbeitet** und neuangelegt werden, verschwinden dabei unter Umständen. Deshalb ist die freie Kapazität streng zu beobachten. Dabei ist es eine Hilfe, dass bei normalen Konfigurationen per Cronjob eine eMail generiert wird, die auch diese Information liefert. Man kann das Problem entschärfen, in dem man Quotas verwendet oder eine eigene Partition für News bereitstellt. Newsserver neigen im Betrieb dazu, riesige Datenmengen zu produzieren.