Linux INN-Newsspool mini-HOWTO

Robert Fendt (fendt@student.physik.uni-dortmund.de) v1.5-2, 14. Januar 1999

Dieses Dokument soll beim Aufbau eines lokalen INN-Newsservers Hilfestellung leisten, um Offline-Reading von News mit beliebiger Software zu ermöglichen.

Inhaltsverzeichnis

1 Einleitung

1.1 Neue Versionen dieses Dokuments

Die jeweils neuste Version dieser HOWTO ist auf dem WWW Server des Deutschen Linux HOWTO Projekts unter folgender Adresse zu finden:

`http://www.tu-harburg.de/dlhp/`

1.2 Feedback

Wenn Sie irgendwelche Fragen oder Kommentare zu diesem Dokument haben, erreichen Sie mich unter der E-Mail Adresse

Internet: `fendt@student.physik.uni-dortmund.de`

Ansonsten bin ich zu erreichen unter

Robert Fendt
Bergstraße 42c
D-44575 Castrop-Rauxel

Ich freue mich über alle Vorschläge und Kritiken zu dieser HOWTO. Wenn Ihnen also irgendein Abschnitt in dieser HOWTO unklar sein sollte, oder wenn Sie Ergänzungsvorschläge haben, schreiben Sie mir bitte.

1.3 Copyright

1.4 Zweck dieses Dokuments

Dieses Dokument soll die Konfiguration eines lokal genutzten INN-Pakets erleichtern, wie es häufig zum Offline-Lesen von Usenet-Groups benötigt wird. Ein lokaler NNTP-Server ermöglicht die Benutzung eines beliebigen News-Readers nach Wahl und verringert gleichzeitig durch das lokale Speichern der Artikel die Telefonkosten erheblich. Die hier angegebenen Konfigurationen sollen lediglich *funktionieren*. Es wird wahrscheinlich für Etliches optimalere Einstellungen geben; ich verwende, soweit sinnvoll, möglichst die Defaults, um Arbeit zu sparen. An solchen Stellen gebe ich allerdings auch entsprechende Hinweise auf die Dokumentation. Als Beispiel wird die SuSE-Linux Distribution herangezogen, da es sich hierbei um eine recht häufige Distribution im deutschsprachigen Raum handelt (und weil ich mit den anderen Distributionen wenig Erfahrung habe). Bei anderen Distributionen können u.a. die Verzeichnisnamen variieren.

2 Vorbereitungen

Um anfangen zu können, wird zunächst einmal ein laufender innd benötigt. Dazu muß also das INN-Paket korrekt installiert sein. Die Installationsanleitung des INN (`Install.ms`) liefert hierzu eine gute und ausführliche Dokumentation. Bestimmte Distributionen bieten auch automatisierte Installationsroutinen an. Im SuSE-Linux 5 z.B. genügt die Installation des INN-rpm's über YaST und das Setzen der Konfigurationsvariablen `START_INN` auf `yes`.

Ferner sollte die Variable `NNTPSERVER=localhost` gesetzt sein. Dies kann z.B. im `profile` erfolgen (oder bei SuSE: `NNTPSERVER=localhost"`

in der `rc.config`).

Neben dem INN- brauchen wir auch noch das suck-Paket. suck ist ein Programm, um von anderen NNTP-Hosts Artikel »herunterzusaugen« (to suck), um diese dann in die INN-Datenbank einzubinden. INN besitzt zwar selbst auch ein Programm dazu, jedoch ist dies anders ausgelegt und für unsere Zwecke mehr als umständlich.

Haben wir alles geschafft – also einen laufenden innd, ein funktionstüchtiges suck und die NNTPSERVER-Variable auf »localhost«, dann kann es fast schon losgehen. Zum Abschluß brauchen wir noch das Kürzel des verwendeten NNTP-Servers (`news.uni-dortmund.de` benutzt z.B. `Uni-Dortmund.DE`; dieser Name steht am Anfang von `path` in der Headerinformation jedes Artikels) und die vollständigen Namen der zu bestellenden Newsgroups (NGs), also z.B. `de.comp.os.linux.misc`. Man sollte auch den fqdn des Rechners parat (und richtig eingestellt!) haben- hat man keinen, muß man sich unbedingt einen besorgen (siehe auch den Abschnitt am Schluß des Dokuments).

Dann haben wir aber wirklich alles zusammen, was nötig ist. Alle jetzt folgenden Befehle *müssen* übrigens als User `news` eingegeben werden, also am besten

```
su news
```

von einer root-shell aus eingeben. Einige Befehle sind aus Platzgründen mittels »\« auf mehrere Zeilen »gebrochen«; dies hat keine tiefere Bedeutung.

3 Die Konfiguration

3.1 Der innd

Die Konfiguration des innd beschränkt sich auf das Anlegen der nötigen Newsgroups und das Anlegen eines Newsfeeds. Die NGs werden mit dem Befehl

```
/usr/lib/news/bin/ctlinnd newgroup <Gruppenname>
```

angelegt, gelöscht werden sie mit:

```
/usr/lib/news/bin/ctlinnd rmgroup <Gruppenname>
```

Dann muß in der Datei `/var/lib/news/newsfeeds` eine Zeile mit dem Inhalt

```
<Serverkürzel>:*,!junk,!control::
```

hinzugefügt werden, also z.B

```
Uni-Dortmund.DE:*,!junk,!control::
```

Diese Zeile weist inn an, jede neue Nachricht, die nicht von `Uni-Dortmund.DE` kommt, zum Versand an diesen Server zu markieren. Das ist nötig, um selber Artikel posten zu können. Man kann hier auch noch vielfältige Filtermöglichkeiten eingeben; `man newsfeeds` gibt hierüber Aufschluß. Kurze Erklärung: der Zusatz `,!junk,!control` sorgt dafür, daß Nachrichten in diesen beiden internen Gruppen *nicht* für den Versand nach draußen markiert werden. In `junk` landet jede ungültige Nachricht (der ganze Müll halt) und `control` ist hauptsächlich für ISP's interessant (dient zur automatischen Administration von Newsservern).

Jetzt muß man den Newsfeed noch mit

```
/usr/lib/news/bin/ctlinnd begin <Serverkürzel>
```

aktivieren, also z.B. mit

```
/usr/lib/news/bin/ctlinnd begin Uni-Dortmund.DE .
```

Naja, *ganz* fertig sind wir jetzt noch nicht. Eine Datei muß noch geändert werden: `/var/lib/news/inn.conf`. Diese sollte nachher in etwa so aussehen:

```
pathhost:       <Newshost>
organization:   private
server:         localhost
```

Die `organisation` und `pathhost` Parameter sind im Prinzip beliebig. Man sollte `pathhost` allerdings auf die fqdn des Provider-Newshosts einstellen, um Verwirrungen zu vermeiden.

3.2 suck! oder: Wie zum Teufel kriege ich News?

Normalerweise sind NNTP-Server so eingestellt, daß sie schön brav warten, bis ihnen ein anderer NNTP-Server neue Artikel schickt. Andererseits schicken sie jedem aktiven Newsfeed jede neue Nachricht, die nicht von ihm stammt. Hier können wir leider lange darauf warten, daß uns irgendein NNTP-Host neue Nachrichten schickt; abgesehen davon wollen wir ja nur bestimmte NGs, und ständig im Netz zu bleiben und zu warten, ist auch nicht gerade der Hammer. Also muß suck her! Mit suck können wir ganz spezielle NGs *abbonnieren* und auf Befehl aktualisieren.

Zunächst einmal brauchen wir ein temporäres Verzeichnis für die gepollten News. Hierzu eignet sich z.B. `/tmp`, aber auch `/var/spool/news`. In diesem Verzeichnis muß das Verzeichnis `Msgs` existieren, und der User `news` muß Schreibrechte auf alles haben. Gepollt wird dann z.B. mit dem Befehl

```
suck <Newshost> -c -bi <Verz.>/batch -dt <Verz.> -dm \
<Verz.>/Msgs -dd <Verz.>
```

<Newshost> ist hier der fqdn des hosts, also z.B. `news.uni-dortmund.de`. Bei Verwendung eines anderen Verzeichnisses als `/var/spool/news` sollte man beachten, daß suck seine Konfigurationsdateien im durch `-dd` angegebenen Verzeichnis sucht; hier sollte man also auf keinen Fall `/tmp` benutzen, sondern trotzdem `/var/spool/news` einsetzen. Der Parameter `-dm` weist suck an, im entsprechenden Verzeichnis die geholten Artikel abzulegen, während `-dt` das Verzeichnis für die Temporärdateien angibt, welche allerdings durch `-c` sowieso automatisch gelöscht werden.

Die ganze Sache könnte dann also so aussehen:

```
suck news.uni-dortmund.de -c -bi /var/spool/news/batch -dt \
/var/spool/news -dm /var/spool/news/Msgs -dd /var/spool/news
```

Damit das aber klappt, muß man suck erstmal sagen, welche NGs er pollen soll. Das geschieht mittels eines Eintrags in die Datei <Verz.>/sucknewsrc. Die Zeile `de.comp.os.linux.misc -100` weist suck z.B. an, die letzten 100 Nachrichten aus der NG zu pollen. Danach wird automatisch anstelle der `-100` die Nummer des letzten Artikels eingesetzt, so daß suck immer nur die neuen Artikel pollt. Hier sollte man also möglichst nicht mehr drangehen.

Nach dem Pollen der News muß inn auch mitkriegen, daß da was gekommen ist. Das geschieht mittels

```
/usr/lib/news/bin/innxmit localhost <Verz.>/batch
```

Diese beiden Befehle packt man am besten in ein Skript, z.B. `/var/spool/news/get.news`, da sie beide immer in Folge ausgeführt werden. Anstatt dessen kann man auch das bei suck mitgelieferte Skript `get.news.innd` zusammen mit `put.news` verwenden. Dieses Skript ist recht leistungsfähig und erledigt auch noch das Versenden von Artikeln. Man muß zur Benutzung allerdings die Variablen am Anfang des Skripts ändern. Dann kopiert man es unter dem Namen `get.news` zusammen mit `put.news`, dem Nachrichtenfilter, in <Verz.>, also z.B. ins Verzeichnis `/var/spool/news`.

3.3 rpost oder: Wie zum Geier kriege ich die Artikel raus?

Jeder Artikel, der zum Versenden an andere Newshosts markiert ist, muß von einem getrennten Programm verschickt werden. Das hier verwendete Programm ist `rpost`. Damit sich `rpost` und der `innd` nicht ins Gehege kommen, sollte man vor dem eigentlichen Versenden noch folgenden Aufruf einbauen:

```
mv /var/spool/news/out.going/<Serverkürzel> \
/var/spool/news/out.going/<Serverkürzel>.new
/usr/lib/news/bin/ctlinnd flush <Serverkürzel>
```

Dieser Aufruf benennt zunächst die Queue-Datei um und veranlaßt dann den `innd`, die Datei zu aktualisieren, zu schließen und neu zu öffnen. Das die Datei zwischenzeitlich umbenannt wurde, ist `innd` egal, da sie ja bereits offen ist. Durch das Schließen und Neuöffnen allerdings wird dann natürlich die Queue-Datei neu angelegt. `rpost` kann jetzt ungestört die Artikel posten und danach automatisch

```
/var/spool/news/out.going/<Serverkürzel>.new
```

löschen. Aufgerufen wird `rpost` mit

```
/usr/bin/rpost <Newshost> -d -b \
/var/spool/news/out.going/<Serverkürzel>.new -p /var/spool/news
```

Man kann, falls nötig, auch noch ein Filterprogramm einsetzen (z.B. `put.news` aus dem suck-Paket), um die ausgehenden Nachrichten noch zu bearbeiten. In diesem Fall wäre der Aufruf

```
/usr/bin/rpost <Newshost> -d -b \
/var/spool/news/out.going/<Serverkürzel>.new -p /var/spool/news \
-f /var/spool/news/put.news \$\$o=/tmp/filtered_msg \
\$\$i /tmp/filtered_msg
```

Für eine genaue Anleitung zur Benutzung von `rpost` sollte man einen Blick in die man-page werfen. Auch diesen Aufruf packt man mit den beiden vorigen am besten in ein Skript, da das Versenden der Artikel am besten gleichzeitig mit dem Empfang von neuen Artikeln durchgeführt wird.

3.4 news.daily

Wenn der root-User beim Hochfahren des Systems jedesmal eine Mail bekommt, in der sich inn über ein fehlendes oder veraltetes `.news.daily` beschwert, dann wurde an diesem Tag noch nicht bzw. noch nie das Skript `/usr/lib/news/bin/news.daily` ausgeführt, welches u.a. zum Löschen von alten Artikeln zuständig ist. Dieses Skript sollte am besten per cron einmal täglich ausgeführt werden. Wenn auf Ihrem System nur der root-user eine crontab haben darf, tragen sie dort z.B. ein:

```
00 22 * * * su news -c /usr/lib/news/bin/news.daily
```

In diesem Beispiel wird täglich um 22.00 Uhr das Skript `news.daily` vom User `news` ausgeführt. Man kann natürlich auch andere Zeiten eintragen, allerdings sollte man das `news.daily`-Skript genau einmal pro Tag ausführen - nicht weniger oft und auch nicht öfter. Wenn man es öfter ausführt, werden sonst u.U. bestimmte Log-Dateien zu schnell rotiert. Wenn `news.daily` mehrfach täglich aufgerufen werden soll, müssen die zusätzlichen Aufrufe so aussehen:

```
su news -c /usr/lib/news/bin/news.daily -notdaily
```

Sie können natürlich `news.daily` auch in die crontab des users `news` eintragen, wobei hierbei dann `su news -c` entfallen kann:

```
00 22 * * * /usr/lib/news/bin/news.daily
```

Das Skript `news.daily` steuert u.a. das 'expire'-Programm, welches seine Daten aus der Datei `/var/lib/news/expire.ctl` bezieht. Hier kann man angeben, wie lange eine Nachricht mindestens (Standard: 1 Tag) bzw. höchstens (Standard: 10 Tage) im News-System verbleibt. Man kann diese Angaben auch abhängig von den einzelnen NG's machen; die Datei ist selbsterklärend, und eine man-Page gibt es auch dazu.

3.5 Filtern

Leider sind die Informationen, die im Header von eigenen Postings (und damit von den ausgehenden Artikeln) erscheinen, i. Allg. nicht korrekt. Einige Dinge sind lediglich Schönheitsfehler, wie z.B. der »NNTP-Posting-Host«, »Xref« oder »Message-ID«. Andererseits produzieren viele News-Reader ihre »From«-Angabe aus der lokalen Paßwort-Datei, was normalerweise mehr als unerwünscht ist: so könnte der Absender auf einmal lauten:

```
From: fritz@dummy.local.net (Fritz Mueller)
```

Diese »From«-Zeile ist schlicht und ergreifend falsch; E-Mail-Replys gehen daher ins Leere. Einigen Newsreadern kann man dieses Verhalten abgewöhnen, aber leider nicht allen. Um dennoch mit einem solchen Programm arbeiten zu können, hilft ein (ziemlich schmutziger) Trick. Und zwar wird normalerweise zusammen mit rpost ein Filter-Skript benutzt (Im Beispiel oben schon einmal genannt; bei suck z.B. mitgeliefert: `put.news`), das genau die oben genannten unschönen Headerzeilen aus jeder Nachricht entfernt, bevor sie abgeschickt wird.

Dies geschieht über sed. Der entsprechende Befehl lautet

```
sed -e /^Zeichenkette/d Datei > Zieldatei .
```

Hiermit wird jede Zeile, die mit *Zeichenkette* beginnt (wenn man »^« wegläßt, dann jede Zeile, die *Zeichenkette* enthält), nicht in *Zieldatei* ausgegeben, also quasi gelöscht. Durch piping kann man mehrere solcher Aufrufe auch koppeln.

Das `put.news`-Skript kann man auch dazu verwenden, um die »From«-Zeile zu ändern. Eine entsprechend angepaßte `put.news`-Version, die auf dem Skript von suck basiert, habe ich hier abgedruckt, da das nötige Tool sed etwas kryptisch zu bedienen ist. Beim Abtippen muß man nur noch die voreingestellte »From«-Zeile nach eigenem Gusto verändern.

Wen's interessiert: Das Ersetzen wird über den sed-Befehl »c\« realisiert. Die Befehlszeile lautet dann z.B.

```
sed -e '/^Zeichenkette/c\
Ersatzzeichenkette' Datei > Zieldatei .
```

Wichtig sind hier unbedingt die einfachen Anführungszeichen und der Zeilenwechsel. Dieser gehört zum sed-Befehl (siehe man-Seite).

```
#!/bin/sh

#set -x
# this is just a simple script to use sed to strip off the
# NNTP_Posting_Host and Xref headers that my ISP's newsfeed
# doesn't like.  this could be written as a one liner
# sed -e SEDCMD1 $1 | sed SEDCMD2 > $2
#
# *** modifications:
# *** 4/98 : transferred 'From'-part can be modified
# *** 1/99 : Message-IDs keep unchanged

if [ $# -ne 2 ]; then
        echo
        echo "Syntax `basename $0` Eingabedatei Ausgabedatei <RETURN>"
        echo
        exit -1
fi
```

```
SEDCMD="/^NNTP-Posting-Host/d"
SEDCMD2="/^Xref/d"
OUTFILE=$2
INFILE=$1

if [ -f ${INFILE} ]; then

sed -e ${SEDCMD} ${INFILE} | sed -e ${SEDCMD2} | sed -e '/^From/c\
From: fmueller@provider.de (Fritz Mueller)' > ${OUTFILE}

        if [ $? -ne 0 ]; then
                echo "Fehler"
                exit -1
        fi

else
        echo "$1 existiert nicht"
        exit -1
fi
```

3.6 Bonus 1: Gruppenbeschreibungen

Sie haben sicherlich schon einmal online News gelesen, und Ihnen ist dabei vielleicht auch aufgefallen, daß zu den Newsgroups i.Allg. eine kurze Beschreibung verfügbar ist. Kann der Reader diese Kommentare anzeigen, können sie recht interessant sein (wenn man nicht gerade den Netscape »Collabra« benutzt). Und so wird's gemacht: in die Datei `/var/lib/news/newsgroups` wird einfach eine Liste mit den Gruppen und ihren Beschreibungen geschrieben. Zum besseren Verständnis gibt's hier ein Beispiel:

```
control                 News server internal group
junk                    News server internal group
de.comp.os.unix.linux.hardware  Hardwarefragen zu Linux.
de.comp.os.unix.linux.misc      Alles, was sich nicht anders einordnen laesst.
de.comp.os.unix.linux.newusers  Fragen von Linux-Anfaengern.
de.comp.os.unix.discussion      Nichtigkeiten, Smalltalk und Glaubenskriege.
de.comp.os.unix.x11     No description.
comp.os.linux.misc      Linux-specific topics not covered by other groups.
comp.os.linux.announce  Announcements important to the Linux community. (Moderat
ed)
de.newusers.questions   Neue Benutzer im Netz fragen, Experten antworten.
```

Das sind die wichtigsten Linux-Gruppen mit den (hoffentlich) korrekten Kommentaren, plus die beiden internen Gruppen 'control' und 'junk'.

3.7 Bonus 2: Wie man sich einen eigenen Domainnamen besorgt

Wer heutzutage einen Account beantragt, ist oft mit zwei Problemen geschlagen: dynamischer IP-Vergabe und fehlendem Namensraum. Sprich: man hat keine feste IP-Nummer und keinen Domainnamen. Also kann man keine gültigen Message-IDs erzeugen. Deshalb muß irgendwie zumindest ein Namensraum her, da man die Message-IDs auch nicht einfach löschen sollte. Das funktioniert zwar (ältere Versionen dieses HOWTOs schlugen dies auch vor), ist aber gefährlich. Bei falsch eingestellten Newsfeeds oder Alias-Änderungen seitens des Providers kann man so jede Menge »Dupes« (also doppelte Mails mit unterschiedlicher Message-ID) erzeugen und einige Leute ziemlich ärgern.

Bestimmte Provider (u.a. z.B. T-Online) stellen einen Namensraum speziell zur Message-ID-Generierung zur Verfügung. Da allerdings INN die Message-IDs aus dem intern benutzten Rechnernamen ableitet, muß man dem Rechner u.U. einen etwas komplizierten Namen geben (bei T-Online enthält er z.B. u.a. die Benutzernummer) oder INN neu kompilieren; es ist aber immerhin eine Möglichkeit. Hier sollte man also bei seinem Provider Erkundigungen einholen.

Eine andere Möglichkeit, einen eigenen fqdn zu bekommen, soll hier vorgeführt werden: ddns.org. Über diese Domain kann jeder kostenlos eine eigene Subdomain bekommen, sogar mit dynamischen IPs. Man beantragt hierfür unter »dynamic DNS« einen Account. Hat man diesen bekommen (Benachrichtigung per E-Mail), kann man fortan kurzfristig eine IP unter der gewählten Domain registrieren, und (beim Trennen der Verbindung) wieder löschen. Diesen Vorgang automatisiert man (wenn man will) am besten durch Installation des Linux-Clients und entsprechende Skripten. Danach kann man die Domain »ddns.org« im System eintragen, so daß dann die Message-IDs mit `<rechername>.ddns.org` absolut gültig sind.