

**NAME**

leafnode – ein NNTP server für kleine Sites (mit Wählverbindung)

**ÜBERSICHT**

**leafnode**

**BESCHREIBUNG**

**Leafnode** ist ein USENET-Software-Paket, das für kleine Sites gedacht ist, wo es wenige User und nur geringe Festplatten-Kapazität gibt, aber trotzdem eine große Zahl von Gruppen gewünscht wird.

**Leafnode** ist so entworfen, dass es Fehler selbständig reparieren und keinerlei manuelle Pflege benötigen soll.

Das **leafnode** -Programm selbst ist der NNTP-Server. Es wird von **inetd(8)**, **xinetd(8)** oder **tcpserver** gestartet, wenn jemand News lesen möchte. Die anderen Teile des Paketes, **fetchnews** und **texpire**, sind dafür zuständig, News von anderen Servern zu holen und nach einer gewissen Zeit zu löschen.

**ZUGANGSKONTROLLE**

**Leafnode** bietet keinerlei Authentifizierungs- oder Zugangskontrolle an. Eine solche zu implementieren, ist eine Aufgabe, die nicht für jeden einzelnen Dienst neu vorgenommen werden sollte.

**Es ist daher dringendst anzuraten, externe Zugangskontrolle per tcpd, inetd/xinetd mit libwrap-Unterstützung, tcpserver mit -x oder vergleichbarer Funktionalität einzurichten.** tcpd und libwrap sind Bestandteil des **tcp\_wrappers** Paketes von Wietse Venema.

Als grobe sichernde Maßnahme gegen Mißbrauch verweigert **leafnode** Verbindungen, die nicht von einem lokalen Netz (LAN) stammen. Dies kann in der Konfiguration ausgestellt werden, ist aber nicht anzuraten (siehe Option allowstrangers), da die Gefahr von Mißbrauch ohne weitere Sicherungsmaßnahmen sehr groß ist.

**DATEIEN UND VERZEICHNISSE**

Für alle folgenden Dateien und Verzeichnisse muß der User "news" Leserechte besitzen. Es wird angeraten, falls im weiteren nicht anders angegeben, daß "news" der einzige Benutzer in der Gruppe "news" sein sollte und die Dateien "root:news" (Benutzer root, Gruppe news) gehören sollten, so daß die Konfiguration- und Filterdateien von leafnode nicht geändert werden können.

*/opt/ln1/etc* sollte für den Benutzer "news" nicht beschreibbar, für die Gruppe "news" aber les- und ausführbar sein (g+ux).

*/opt/ln1/etc/config* enthält die Konfigurationseinstellungen für **leafnode**. Diese Datei **darf nicht** schreibbar sein für den Benutzer "news". Setzen Sie diese Datei auf mode 640, Besitzer root:news. Siehe auch KONFIGURATION unten.

*/opt/ln1/var/spool/news* muß für den Benutzer "news" les- und schreibbar sein. Es enthält die News-Artikel, so enthält z. B. das Verzeichnis */opt/ln1/var/spool/news/alt/fan/agulbra* die Artikel der Newsgruppe alt.fan.agulbra. Jedes Verzeichnis enthält die Artikel in durchnummerierten Dateien (Dezimalzahlen, streng monoton steigend) und eine spezielle Datei *.overview*, die die "Subject"-, "From"-, "Date"-, "Message-ID"-, "References"-, "Bytes"- und "Lines"-Header für jeden einzelnen Artikel der Gruppe enthält.

Mehrere Unterverzeichnisse haben spezielle Aufgaben:

*/opt/ln1/var/spool/news/leaf.node* enthält diejenigen Dateien, die **leafnode** während des Betriebs anlegt, zum Beispiel die **groupinfo-Datei**, die Informationen über jede einzelne USENET-Gruppe enthält. Diese Datei wird von **fetchnews** (8) erstellt, Sie können diese Datei jederzeit neu erstellen lassen, indem Sie **fetchnews** mit dem Parameter -f aufrufen (s. **fetchnews** (8)).

*/opt/ln1/var/spool/news/out.going* enthält lokale Artikel, die von **fetchnews** (8) an den Upstream-Server weitergereicht werden sollen. Während ein Posting von einem Newsreader empfangen wird, ist das Leserecht für den Benutzer "News" nicht gesetzt (u-r). Nachdem ein Posting erfolgreich in dieses

Verzeichnis geschrieben wurde, wird das Leserecht für die Datei gesetzt (u+r). **fetchnews(8)** interpretiert dieses Recht als "dieses Posting darf veröffentlicht werden", um zu verhindern, dass Postings verschickt werden, die noch nicht vollständig empfangen worden sind. Hinweis: Versionen von 1.9.23 bis 1.9.32 einschließlich haben statt des Leserechts das Recht zur Ausführung (u+x) benutzt. Wegen Unverträglichkeiten (Artikel werden von Leafnode nicht weitergeleitet) nach Updates von Versionen vor 1.9.23 wurde mit Version 1.9.33 auf Leserecht umgestellt, damit sind alle von älteren leafnode-1.9-Versionen verträglich.

*/opt/ln1/var/spool/news/failed.postings* enthält lokale Postings, die vom Upstream-Server zurückgewiesen wurden. **Fetchnews** erstellt Dateien in diesem Verzeichnis, allerdings löscht keines der **leafnode** -Programme dort Dateien.

*/opt/ln1/var/spool/news/message.id* enthält harte Links zu jedem Artikel; dies ersetzt die **dbz** Datenbank, die größere Server einsetzen. (Ein solches Verzeichnis ist auf kleinen Servern, für die **leafnode** gedacht ist, effizienter, skaliert dafür aber sehr schlecht.)

*/opt/ln1/var/spool/news/interesting.groups* enthält pro Gruppe, die ein NNTP-Klient angefordert hat, eine Datei. **Leafnode** aktualisiert die **ctime** (ls -l zeigt normalerweise die **mtime**, versuchen Sie ls -lc) der zugehörigen Datei, wenn ein LISTGROUP-, XOVER-, XHDR-, STAT-, HEAD-, BODY- oder ARTICLE-Befehl nach einem GROUP-Befehl ausgeführt wird, oder wenn ein GROUP- oder LIST ACTIVE-Befehl für eine bereits "interessante" (bestellte) Gruppe ausgeführt wird. LIST ACTIVE mit Muster frischt bestehende Bestellungen nicht auf. **fetchnews** (8) lädt die Artikel derjenigen Gruppen, auf deren Dateien entweder

- innerhalb der letzten zwei Tage, oder
- mehr als einmal und mindestens einmal in der vergangenen Woche zugegriffen wurde.

Diese Zeitspannen können mittels **timeout\_short** und **timeout\_long** in der *config*-Datei angepaßt werden. Siehe auch die **-n**-Option in **fetchnews(8)**.

*/etc/inetd.conf* oder */etc/xinetd.conf* enthält die Konfiguration, die **leafnode** startet. Es wird dringend angeraten, **leafnode** vom User "news" starten zu lassen.

## UMGEBUNGSVARIABLEN

### LN\_REJECT\_POST\_PRE

Wenn diese Variable existiert, wird jeder POST-Befehl mit einem 400-Fehlercode verweigert. Nur zum Debugging von Newsreadern benutzen!

### LN\_REJECT\_POST\_POST

Wenn diese Variable existiert, wird jeder POST-Befehl mit einem 400-Fehlercode verweigert, nach dem Artikel und CRLF.CRLF empfangen worden sind. Nur zum Debugging von Newsreadern benutzen!

## KONFIGURATION

Die komplette Konfiguration wird in der Datei */opt/ln1/etc/config* vorgenommen, die eine Filterbeschreibungsdatei, *filterfile*, aufrufen kann, wie unten gezeigt.

Im Rahmen dieses Abschnitts sei "Whitespace" als eine beliebige Folge von einem oder mehr SPACE (Leerzeichen) und HTAB (horizontaler Tabulator) definiert, ASCII-Positionen 32 bzw. 9.

Die Konfigurationsdatei ist streng zeilenorientiert, sie kann als Zeilenabschluss entweder LF (Zeilenvorschub) oder CRLF (Wagenrücklauf + Zeilenvorschub) verwenden.

Leere Zeilen und solche, die ausschließlich aus Whitespace bestehen, optional von einem Kommentar gefolgt werden (der von einem #-Zeichen eingeleitet wird und sich bis zum Zeilenende erstreckt), werden übersprungen.

Alle anderen Zeilen bestehen aus genau drei Pflichtfeldern, einem Parameter im Klartext, einem Gleichheitszeichen (=), der von Whitespace umgeben sein darf und einem Wert. Der Wert kann als Klartext angegeben sein oder - neu seit Leafnode v1.11 - als Zeichenkette in Anführungszeichen mit trivialen

Fluchtzeichen (siehe unten).

Klartext fängt beim ersten nicht-Whitespace-Zeichen an und erstreckt sich bis zum letzten nicht-Whitespace-Zeichen, das nicht zu einem Kommentar gehört. Ein Kommentar am Ende einer Zeile wird ignoriert.

Zeichenketten in Anführungszeichen verwenden das doppelte Anführungszeichen oben ("). Das Backslash-(Rückwärtsschrägstrich)-Zeichen (\) ist ein Fluchtzeichen und wird übersprungen, liest aber das unmittelbar folgende Zeichen originalgetreu, sodass das Backslash-Zeichen (durch Verdopplung, als \\) und Anführungszeichen (als \") in der Zeichenkette angegeben werden können; das #-Zeichen verliert zwischen den Anführungszeichen seine besondere Bedeutung als Kommentareinleitung. Text nach dem Ende der Zeichenkette wird stillschweigend ignoriert (das kann sich in zukünftigen Versionen ändern). Kommentare am Ende der Zeile werden ignoriert.

### OBLIGATORISCHE PARAMETER

Diese Parameter müssen angegeben werden, damit Leafnode funktioniert.

`server = news02.example.com`

"server" wird von **fetchnews** (8) benutzt, um festzustellen, von welchem Server es News herunterladen und wohin es neue Artikel posten soll. Es ist möglich, mehrere Server anzugeben, sie werden der Reihe nach kontaktiert. Sie brauchen auf diesen Servern Leserechte, falls Sie selber Artikel veröffentlichen möchten, auf mindestens einem der Server auch Schreibrechte. Im obigen Beispiel ist news02.example.com der primäre News-Server.

Dieser Parameter kann mehrmals angegeben werden. Jeder **Server** erhält dabei zunächst einen Satz Standard-Optionen, die Einstellungen des vorherigen Server-Eintrags werden nicht übernommen. Ebenso können Server-spezifische Einstellungen nicht global getätigt werden.

`expire = 5`

"expire" ist die Anzahl der Tage, die ein Artikel aufbewahrt werden soll. Im Beispiel wird ein Artikel fünf Tage, nachdem er zuletzt gelesen wurde, durch **texpire**(8) gelöscht. Dieser Wert MUSS wenigstens 1 sein.

### SERVERUNABHÄNGIGE OPTIONALE PARAMETER

Diese Optionen können nur einmal in der Konfigurationsdatei angegeben werden und wirken sich auf Leafnode insgesamt aus. Es ist unwichtig, an welcher Stelle diese Optionen - auch relativ zu `server=-`Zeilen - stehen, es ist aber aus Gründen der Übersicht sinnvoll, sie noch vor der ersten `server=-`Zeile anzugeben. Wird eine dieser Optionen mehrfach anzugeben, wird die letzte Angabe verwendet; zukünftig kann die mehrfache Angabe aber ein Fehler sein.

`hostname = host.domain.country`

**Leafnode** versucht zunächst, den Namen des Computers zu ermitteln und dabei ungültige Namen auszulassen. Der host name wird mit `gethostname(3)` erfragt und mit bei Bedarf mit `gethostbyname(3)` aufgelöst. Dieser Name entstammt üblicherweise der Datei `/etc/hosts` oder NIS und DNS (Details in der System-Dokumentation).

Falls Leafnode keinen host-Namen ermitteln kann, deutet das in der Regel auf ein nicht sorgfältig konfiguriertes System oder auf einen nicht eindeutigen Namen hin, der für die Erstellung von Message-IDs nicht verwendbar ist. Namen wie "localhost.localdomain" sind nicht eindeutig und daher zu ändern. Es genügt in der Regel, einen sogenannten fully-qualified host name (FQDN) der Datei `/etc/hosts` hinzuzufügen. In der Datei *README-FQDN* werden weitere Informationen dazu (auf Englisch gegeben), weiterhin finden sich in der Datei *INSTALL\_de* Hinweise auf deutsche Dokumentation zu diesem Thema.

Sie können den Teil der Message-ID nach dem @ auch mit dem Parameter `hostname` konfigurieren. Die Verwendung von "hostname" zur Erzeugung einer Phantasie-Message-ID ist zwar möglich, aber dringend abzuraten.

`create_all_links = 1`

Üblicherweise speichert **fetchnews** Artikel nur in den Newsgruppen, die in `/opt/ln1/var/spool/news/interesting.groups` vorhanden sind, wenn Sie `create_all_links` auf 1 setzen, erstellt **fetchnews** jedoch für jede Newsgruppe, die es im "Newsgroups:"-Header findet, einen harten Link. Dies kann interessant für Sie sein, wenn Sie ein Score- oder Kill-File auf den "Xref:"-Header anwenden möchten.

`maxfetch = 1000`

"maxfetch" begrenzt die Anzahl der Artikel, die **fetchnews** (8) vom Upstream-Server laden darf. Die Benutzung dieses Parameters ist jedoch nicht zu empfehlen, da Sie sonst nicht alle Artikel einer Gruppe bekommen. In der Grundeinstellung gibt es keine Obergrenze.

`initialfetch = 1`

"initialfetch" beschränkt die Anzahl der Artikel, die von einer vorher noch nicht bezogenen Gruppe geladen werden sollen. Normalerweise werden alle alten Artikel geladen, was allerdings sehr zeitaufwendig werden kann, wenn in dieser Gruppe sehr viele Artikel gepostet werden. Dies entspricht einem Wert von 0 für initialfetch; wenn Sie keine alten Artikel empfangen möchten, setzen Sie den Wert, wie im Beispiel, auf 1.

`groupexpire sehr.volle.Gruppe = 1`

`groupexpire sehr.volle.Hierarchie.* = 1`

"groupexpire" ermöglicht es, unterschiedliche Haltezeiten für verschiedene Gruppen festzulegen. Die Haltezeit wird in Tagen angegeben, der Wert wird von **texpire** (8) benutzt. Sie können so viele groupexpire-Einträge anlegen wie Sie möchten und dabei **glob** (7)-Wildcards verwenden. Der Wert 0 veranlasst leafnode, den Default zu wählen, negative Werte verhindern das Entfernen alter Artikel (als eine Art Archivmodus).

`maxage = 10`

Wenn ein Artikel auf dem Upstream-Server auftaucht, der älter als "maxage" Tage ist, wird er von fetchnews nicht geladen, auch wenn er noch nicht lokal vorhanden ist. Dies ist nützlich, wenn Ihr Upstream-Server gelegentlich den 'Schluckauf' bekommt. Die Standardeinstellung beträgt 10 Tage; wenn Sie maxage deaktivieren möchten, tragen Sie einen möglichst großen Wert ein, z. B. 100000 (also ca. 270 Jahre)

`maxold = 10`

Ist synonym zu maxage, siehe oben.

`maxlines = 2000`

Wenn Sie es vermeiden möchten, sehr große Artikel zu laden, setzen Sie den "maxlines"-Parameter auf die Anzahl der Zeilen, die ein Artikel höchstens haben darf. Üblicherweise ist dieses Feature deaktiviert.

`minlines = 2`

Manchmal werden Newsgruppen mit leeren Artikeln überhäuft. Um solche Artikel abzuweisen, können sie "minlines" verwenden. "minlines" auf einen Wert größer als 4 zu setzen, ist wahrscheinlich keine gute Idee, da dann u. U. auch echte Artikel abgelehnt werden könnten. Üblicherweise ist dieses Feature deaktiviert.

`maxbytes = 100000`

Wenn Sie große Artikel ablehnen möchten, können Sie statt "maxlines" auch "maxbytes" verwenden. Auch dieses Feature ist normalerweise deaktiviert.

`maxcrosspost = 5`

Wenn Sie Spam bekämpfen möchten, können Sie alle Artikel filtern, die über mehr als "maxcrosspost" Gruppen crossgepostet werden. Den Wert von "maxcrosspost" zu niedrig anzusetzen ist wahrscheinlich keine gute Idee. Auch dieses Feature ist standardmäßig deaktiviert.

`maxgroups = 5`

Ist synonym zu maxcrosspost, siehe oben.

`filterfile = /opt/ln1/etc/filters`

leafnode kann die Header eingehender Artikel nach regulären Ausdrücken filtern, die im "filterfile" gespeichert sind. Das Format dieser Filterdatei ist einfach, ein regulärer Ausdruck pro Zeile. Entspricht einer der regulären Ausdrücke dem Header eines eingehenden Artikels, wird der Body dieses Postings abgelehnt. Dieses Feature ist üblicherweise deaktiviert. Reguläre Ausdrücke werden in **pcr(3)** beschrieben.

`timeout_short = 2`

Normalerweise wird eine Gruppe, die nur zufällig abonniert wurde, 2 Tage lang bezogen. Hier können Sie diese Zeit abändern.

`timeout_long = 7`

Normalerweise wird eine Gruppe, die nicht mehr gelesen wird, noch sieben Tage lang bezogen. Diese Zeitspanne kann hier geändert werden.

`timeout_active = 90`

Normalerweise wird das Active-File alle 90 Tage vom Upstream-Server gezogen. Dieser Abstand kann verändert werden, indem Sie "timeout\_active" einen anderen Wert zuweisen. Bedenken sie dabei aber, daß beim Laden des Active-Files von einem Server, der ca 20.000 Gruppen vorrätig hat, rund ein Megabyte an Informationen übertragen werden.

`timeout_client = 900 (seit v1.9.23)`

Normalerweise beendet **leafnode** nach 900 Sekunden (15 Minuten) Inaktivität die Verbindung zum Newsreader. Mit "timeout\_client" kann diese Zeitspanne angepaßt werden, etwa für defekte Newsreader, die sich nicht neu verbinden können. Es ist nicht ratsam, dies unter 5 Minuten zu setzen, da Neuverbindungen die Systemlast steigern.

`timeout_fetchnews = 300 (seit v1.9.52)`

Fetchnews nimmt, seit v1.9.52, an, dass der Upstream-Server festhängt, wenn es 300 Sekunden gewartet hat. Sie können die Frist hier verändern.

`timeout_lock = 5 (seit v1.9.54)`

Mit dieser Variable, die als natürliche Zahl oder 0 interpretiert wird, kann eingestellt werden, wie viele Sekunden die Programme (applyfilter, checkgroups, fetchnews, texpire) versuchen sollen, die Lock-Datei zu setzen. Voreingestellt sind 5 Sekunden. Der Wert 0 bedeutet, dass die Programme unbegrenzt auf die Lock-Datei warten sollen.

`delaybody = 1`

Wenn dieser Parameter gesetzt ist, lädt **fetchnews** (8) nur die Header eines Artikels zur Ansicht. Nur, wenn die Header gelesen werden, lädt **fetchnews** (8) beim nächsten Start auch den Body des Postings. Sie können so Plattenplatz und On-Line-Zeit sparen.

`delaybody_in_situ = 1 (seit v1.9.41)`

Gilt nur, wenn delaybody=1.

Normalerweise vergibt leafnode für komplett geladene Artikel neue Artikelnummern, damit sie im Newsreader als neu angezeigt werden. Mit einigen Newsreadern funktioniert dies aber nicht. Nach dem Setzen dieser Option wird leafnode stattdessen beim Download die ursprüngliche Artikelnummer erhalten. Sie müssen Ihren Newsreader dazu alte Artikel anzeigen lassen. Diese Option ist normalerweise abgeschaltet. Es wird **eindringlich** empfohlen, sie abgeschaltet zu lassen.

`debugmode = 1`

Wenn Sie diesen Parameter angeben, geben **fetchnews** (8), **texpire** (8) und **leafnode** (8) Debugging-Informationen via **syslog** (8) mit "facitily" "news" und Priorität "debug" aus. Benutzen Sie diese Option, wenn Sie Probleme mit Ihrem Feed haben. Im Normalfall sollte die Option jedoch 0 sein, weil sie enorme Datenmengen erzeugen kann. Zum Debugging werden Werte von 1 bis 3 verwendet. Höhere Werte als 3 machen derzeit keinen Unterschied zur 3.

`allow_8bit_headers = 1` (seit v1.9.25)

Normalerweise läßt Leafnode das Posten von Artikeln, die 8-Bit-Daten im Nachrichtenkopf tragen, nicht zu, weil diese gegen einschlägige Normen, speziell RFC-2822 (auf den RFC-1036 baut), verletzen, und die verlangen, daß Usenet-News-Köpfe nur aus 7-bit US-ASCII-Zeichen bestehen, und aus den Steuerzeichen nur die Freiräume zuläßt.

Da allerdings UTF-8 eingeführt werden soll und einige nationale Hierarchien, besonders die norwegische und dänische (no.\*, dk.\*) sich darauf geeinigt zu haben scheinen, 8-bit Daten einer Codierung mittels RFC-2047 vorzuziehen, können Sie diese Option benutzen, um 8-bit-Daten im Nachrichtenkopf zuzulassen. Leafnode wird in diesem Fall, wenn 8-bit-Daten vorhanden sind, aber einen zusätzliche Kopfzeile zur Warnung einfügen, die erwähnt, daß der Administrator das erlaubt hat.

Es gibt derzeit keine Möglichkeit, beim Posten in Leafnode andere Steuerzeichen als die Leerzeichen (einschließlich Tabulator) im Nachrichtenkopf zuzulassen.

`allowSTRANGERS = MAGIC` (seit v1.9.23)

Normalerweise verweigert leafnode Verbindungen von außerhalb des lokalen Netzes (LAN). In `config.example`, wird beschrieben, wie dieses Verhalten geändert werden kann. An Stelle von `MAGIC` muß eine Nummer eingesetzt werden, siehe `config.example`. Beachten Sie die Groß- und Kleinschreibung.

`linebuffer = 1`

Leafnode verwendet normalerweise voll gepufferte Standardausgabe und -fehlerausgabe, außer bei Verbindung mit einem Terminal. Mit diesem Parameter kann explizit zeilengepufferte Ausgabe erzwungen werden.

`clamp_maxage = 0`

Üblicherweise werden Artikel, die älter sind als die für die aktuelle Gruppe konfigurierte expire Zeit, nicht geladen (siehe `expire` und `groupexpire` neben dem Parameter `maxage`). Dadurch wird verhindert, daß Artikel, die von **texpire**(8) gelöscht wurden, nochmal heruntergeladen werden. Mit `clamp_maxage = 0` wird dieses Verhalten abgestellt.

`article_despite_filter = 1` (seit v1.9.33)

Normalerweise fragt fetchnews Nachrichtenköpfe und -rümpfe separat ab, wenn ein **filterfile** definiert und **delaybody** abgeschaltet ist. Für Verbindungen mit hoher Latenz, aber hohem Durchsatz (wie zum Beispiel DSL mit Interleaving oder Satellitenverbindungen), kann es schneller sein, den kompletten Artikel per "ARTICLE" zu verlangen und bei zutreffenden Filtern den Nachrichtenrumpf mit wegzuwerfen (obwohl das nicht preiswerter sein muß, wenn die Verbindung nach Datenvolumen bezahlt wird). Das Aktivieren dieser Option zwingt fetchnews dazu, den ARTICLE-Befehl auch dann zu benutzen, wenn **filterfile** definiert ist. (Hinweis: wenn **delaybody** aktiviert ist, werden Nachrichtenköpfe und -rümpfe immer getrennt angefordert.)

`newsadmin = news@leafnode.example.org` (seit v1.9.47)

Diese Option stellt den Absender des Platzhalter-Artikels ein, der sollte die Adresse des Leafnode-Administrators sein. Voreingestellt ist `news@HOSTNAME`, wobei `HOSTNAME` Leafnodes Rechnername ist.

## **SERVERSPEZIFISCHE OPTIONALE PARAMETER**

Diese Optionen können nur nach der `server=-`Zeile desjenigen Servers angegeben werden, dessen Funktion sie einstellen sollen, und gehören immer zur vorausgegangenen `server=-`Zeile. Sie vor der ersten `server=-`Zeile anzugeben, ist ein Fehler.

`username = meinname`

Wenn einer Ihrer News-Server eine Authentifizierung erwartet, können Sie hier Ihren Usernamen auf diesem System eingeben. Dieses Feld darf mehrmals erscheinen, einmal nach jeder Server-Definition. Der Anfang dieses Abschnitts beschreibt, wie Sonderzeichen in *meinname* angegeben werden können.

password = meinpassword

Wenn einer Ihrer News-Server eine Authentifizierung erwartet, können Sie hier Ihr Paßwort auf diesem System eingeben. Dieses Feld darf mehrmals erscheinen, einmal nach jeder Server-Definition. Da das Paßwort im Klartext angegeben wird, ist es ratsam, die Rechte an der config-Datei so stark wie möglich einzuschränken, da sonst andere Benutzer Ihres Rechners Ihr/e Paßwort/Paßwörter herausfinden könnten. Der Anfang dieses Abschnitts beschreibt, wie Sonderzeichen in *meinpassword* angegeben werden können.

port = 8000

In den Grundeinstellungen versucht fetchnews, sich mit dem Upstream-Server über den NNTP-Port (119) zu verbinden. Wenn ihr Server auf einem anderen Port läuft, können Sie ihn hier angeben. Dieses Feld darf mehrmals erscheinen, einmal nach jeder Server-Definition.

**Hinweis:** Um den Port einzustellen, auf dem Leafnode selbst Verbindungen entgegennimmt, müssen Sie ihre Konfigurationsdatei inetd.conf, xinetd.conf oder die Befehlszeile von tcpsvd oder tcpserver anpassen. Leafnode öffnet diesen Port nicht selbst.

timeout = 30

Normalerweise versucht **leafnode** 10 Sekunden lang, mit einem Server Verbindung aufzunehmen und gibt diesen Versuch dann auf. Wenn Sie einen langsamen Server haben, können Sie **leafnode** mehr Zeit dafür geben, indem Sie timeout auf einen höheren Wert setzen (im Beispiel 30 Sekunden). "timeout" kann für jeden Server einzeln angepaßt werden.

noactive = 1 (seit v1.9.25)

Wenn diese Option gesetzt ist, wird die Liste der verfügbaren Gruppen nie von diesem Upstream-Server geladen. Diese Option ist sinnvoll für sehr langsame Verbindungen, wenn andere Server die Gruppen ebenfalls führen.

nodesc = 1

Manche Server geben die Beschreibungen der Newsgroups nicht korrekt an, da sie das "LIST NEWSGROUPS"-Kommando nicht verstehen. Fügen Sie in diesem Fall nach dem "server"-Eintrag diese Zeile ein.

nopost = 1 (seit v1.9.23)

Zu diesem Server keine Postings versenden. Nützlich, wenn Ihr Server das Veröffentlichen von Artikel nicht erlaubt und dies falsch mitteilt (NNTP-Antwort 200) oder Ihre Postings nicht zuverlässig weiterleitet.

noread = 1 (seit v1.9.33)

Von diesem Server keine Artikel oder Gruppenlisten beziehen. Nützlich, wenn dieser Server zwar gut zum Posten ist, aber zu langsam zum Newsdownload.

noxover = 1 (seit v1.9.47)

Die Gruppenübersichten werden von diesem Server mit XHDR statt XOVER abgerufen.

only\_groups\_match\_all = 1 (seit v1.9.52)

Normalerweise crosspostet fetchnews bei Definition der only\_groups\_pcre-Option einen Artikel, wenn der reguläre Ausdruck (PCRE) auf WENIGSTENS EINE der in der Newsgroups:-Kopfzeile aufgeführte Newsgruppe passt. Mit dieser Option muss der reguläre Ausdruck auf ALLE Newsgruppen passen, damit er Artikel gepostet wird. Dies kann in Verbindung mit mehreren Upstream-Servern zur Vermeidung von "poison" (Gift)-Newsgroups verwendet werden.

only\_groups\_pcre = PCRE (seit v1.9.28)

Diese Option erlaubt es, einen Perl-kompatiblen, regulären Ausdruck derjenigen Gruppen anzugeben, die von diesem Server geholt oder zu diesem gepostet werden können. Der reguläre Ausdruck wird linker Hand automatisch verankert, so dass der ^ am Anfang ausgelassen werden kann.

Wenn diese Option fehlt, werden alle Gruppen von diesem Server bezogen und zu ihm gepostet.

Hinweis: Sie müssen fetchnews mit der "-f"-Option laufen lassen wann immer Sie eine only\_groups\_pcre-Option hinzugefügt, entfernt oder verändert haben.

Denken Sie aber daran, Punkte zu "escapen" (ihre Joker-Funktion zu unterbinden), zum Beispiel so:

```
de\.comp\|de\.comm\.
```

Tipp: Sie können ihre PCREs beispielsweise so testen:

```
cut -f1 -d" " @spooldir@/leaf.node/groupinfo \
| pcregrep 'PCRE'
```

post\_anygroup = 1 (seit v1.9.37)

Mit dieser Option wird das Posten auf einem Server ohne vorherige Prüfung der Verfügbarkeit der Newsgroup erzwungen. Voreinstellung ist post\_anygroup=0, das heißt, leafnode wird mit einem "GROUP"-Befehl prüfen, ob der Server die Newsgroup führt, in die der Artikel gehen soll. Nützlich auf Servern, die nur Posten zulassen, aber den "GROUP"-Befehl verweigern. Hinweis: unbedachter Einsatz dieser Option kann zum Fehlschlag des Postings und Ablegen der Postings im Verzeichnis failed.postings führen.

### **OBSOLETE PARAMETER**

supplement

ist synonym zur Option server und bei neuen Installationen nicht mehr zu verwenden.

fqdn

ist synonym zur Option hostname und bei neuen Installationen nicht mehr zu verwenden.

### **PROTOKOLL**

**leafnode** unterstützt die folgenden NNTP-Befehle:

ARTICLE, BODY, DATE, GROUP, HDR, HEAD, HELP, LAST, LIST, LIST ACTIVE, LIST ACTIVE.TIMES, LIST EXTENSIONS, LIST NEWSGROUPS, LIST OVERVIEW.FMT, LISTGROUP, MODE, NEWGROUPS, NEXT, POST, OVER, SLAVE, STAT, XHDR, XOVER.

Die Befehle, außer HDR und OVER, sind in RFC-977 und RFC-2980 spezifiziert. Zu beachten ist, daß sich die Syntax von HDR und OVER wegen bislang fehlender Standardisierung noch ändern kann.

### **FEHLER**

Leafnode weist Artikel mit UTF-8-Zeichen in Header-Zeilen, die nicht nach RFC-2047 codiert sind, zurück. Der Usefor-Standardentwurf schlägt UTF-8 ohne RFC-2047-Kodierung für Header vor.

Leafnode beendet das Einlesen einer Zeile beim ersten NUL-Zeichen.

Leafnode kann Probleme mit hierarchieübergreifenden Crosspostings haben, wenn mehr als ein Upstream-Server vorhanden und only\_groups\_pcre konfiguriert ist.

Leafnode versucht nur auf Systemen, die die tm\_gmtoff-Komponente in der tm-Struktur haben (typisch BSD und GNU-Systeme), den Zeitversatz der Zeitzone für von leafnode erzeugte Date:-Kopfzeilen zu ermitteln.

Die Rechtschreibung der Dokumentation ist gemischt alt/aktuell.

### **AUTOR**

Die ersten Versionen von leafnode wurden von Arnt Gulbrandsen <agulbra@troll.no> geschrieben; Copyright 1995 Troll Tech AS, Postboks 6133 Etterstad, 0602 Oslo, Norway, fax +47 22646949.

Weiterentwickelt wurde leafnode von Cornelius Krasel <krasel@wpxx02.toxi.uni-wuerzburg.de>, Randolf Skerka <Randolf.Skerka@gmx.de> und Markus Enzenberger <enz@cip.physik.uni-muenchen.de>. Copyright (an den Veränderungen) 1997-1999.

Weiterentwickelt von Matthias Andree <matthias.andree@gmx.de>, Copyright 1999 - 2002, und Ralf Wildenhues <ralf.wildenhues@gmx.de>, Copyright 2002.

Jonathan Larmour <jlarmour@redhat.com> implementierte die timeout\_client Option.

Andreas Meininger <a.meininger@gmx.net> fügte Code hinzu, so daß texpire Gruppen mit 'groupexpire = -1' ignoriert.

Mark Brown <broonie@debian.org> implementierte die noactive-Option.

Unzählige Beiträge von weiteren Personen.

Die ursprüngliche Entwicklung wurde durch Uninett AS (<http://www.uninett.no/>) finanziert.

Deutsche Übersetzung: Alexander Reinwarth <a.reinwarth@gmx.de>, ergänzt von Ralf Wildenhues.

#### **SIEHE AUCH**

**applyfilter(8), checkgroups(8), fetchnews(8), newsq(1), texpire(8).**

**tcpd(8), hosts\_access(5), glob(7), pcre(3), RFC 977, RFC 2980.**