

Verwaltung von Logfiles mit logrotate

Quellen: LINUX-Anwender-Handbuch

<http://www.linux-praxis.de/lpic1/manpages/logrotate.html>

Inhaltsverzeichnis

1.	Allgemeines.....	2
2.	Konfiguration und Steuerung.....	3
2.1	Konfigurationsdateien.....	3
2.2	wichtige Dateien und Verzeichnisse:.....	3
2.3	Steuerung.....	3
3.	Optionen.....	4
4.	wichtige Direktiven.....	5
5.	Beispiele.....	11
5.1	/etc/logrotate.conf.....	11
5.2	Script in /etc/logrotate.d.....	12



1. Allgemeines

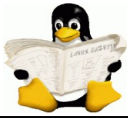
Zahlreiche System-Dienste (engl. Daemons) und auch der Kernel selbst protokollieren regelmäßig Systemzustände oder besondere Vorkommnisse in Protokoll-Dateien (engl. logfiles). So kann der Administrator zuverlässig feststellen, in welchem Zustand sich das System zu einem bestimmten Zeitpunkt befand, Fehler oder Fehlfunktionen erkennen und gezielt beheben. Diese Protokoll-Dateien werden in der Regel gemäß FHS unter `/var/log` abgelegt und werden von Tag zu Tag größer. Mit Hilfe von **logrotate** ist es möglich, das Wachsen der Protokoll-Dateien zu steuern.

logrotate ist dazu entworfen, die Verwaltung von Systemen zu vereinfachen, die eine große Anzahl von Log-Dateien generieren. Es erlaubt eine automatische Rotation, Kompression, Entfernung und Verschiebung der Logdateien. Jede Logdatei kann täglich, wöchentlich, monatlich oder dann erfolgen, wenn sie zu groß geworden ist.

Normalerweise wird **logrotate** als täglicher cron-job ausgeführt. Es bearbeitet eine Logdatei nicht mehr als einmal täglich, wenn das Kriterium für die Behandlung dieser Logdatei nicht seine Größe ist, **logrotate** mehr als einmal täglich aufgerufen wird oder die **-f** oder **--force** Option angegeben wurde.

Auf der Kommandozeile können beliebig viele Konfigurationsdateien angegeben werden. Spätere Konfigurationsdateien können Optionen von früheren überschreiben, so daß es wichtig ist, auf die richtige Reihenfolge zu achten. Normalerweise sollte eine Konfigurationsdatei benutzt werden, die ihrerseits weitere solcher Dateien per `include`-Befehl einfügt. Weiter unten wird die Anwendung der **include**-Direktive genauer beschrieben. Wird auf der Kommandozeile ein Verzeichnis angegeben, so werden alle darin enthaltenen Dateien als Konfigurationsdateien abgearbeitet.

Werden keine Kommandozeilenparameter angegeben, gibt **logrotate** einen kurzen Hilfstext und Versionsinformationen aus. Treten während der Rotation von Logdateien Fehler auf, so gibt **logrotate** einen Rückgabewert ungleich 0 zurück.



2. Konfiguration und Steuerung

2.1 Konfigurationsdateien

In der Konfigurationsdatei

```
/etc/logrotate.conf
```

wird das generelle Verhalten festgelegt. Mit der include-Angabe wird insbesondere konfiguriert, welche weiteren Dateien ausgewertet werden sollen. Bei OpenSUSE LINUX ist vorgesehen, dass die einzelnen Pakete in

```
/etc/logrotate.d
```

Dateien installieren (beispielsweise syslog oder yast). Auch benutzerdefinierte logrotate-Definitionen sollten in dem Verzeichnis abgestellt werden. Hier wird jedoch die Vergabe eines benutzerdefinierten Namens für die Parameterdatei empfohlen.

2.2 wichtige Dateien und Verzeichnisse:

/etc/logrotate.conf generelle Konfigurationsdatei für logrotate

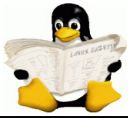
/etc/logrotate.d Scripten, die durch logrotate ausgeführt werden sollen

2.3 Steuerung

logrotate selbst wird über cron gesteuert und einmal täglich von

```
/etc/cron.daily/logrotate
```

angestoßen.

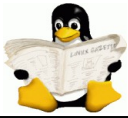


3. Optionen

- d** Schaltet den Debug-Modus ein und impliziert **-v**.
Im Debug-Modus werden keine der Änderungen an den Logdateien und der Status-Datei vorgenommen.
- f, --force** Weist **logrotate** an, die Rotation zu erzwingen, auch wenn sie eigentlich nicht notwendig wäre. Manchmal kann das nützlich sein, wenn neue Einträge in der Konfigurationsdatei vorgenommen wurden, oder wenn alte Logdateien von Hand gelöscht wurden, weil die neuen Dateien so erstellt werden und das Loggen fehlerfrei weiterläuft.
- s, --state** *Statusdatei*

Weist **logrotate** an, eine alternative Statusdatei zu benutzen.
Das kann nützlich sein, wenn mehrere User **logrotate** für verschiedene Logdateien benutzen.
Die voreingestellte Statusdatei ist **/var/lib/logrotate/status**.
- usage** Gibt eine kurze Hilfsmeldung aus
- v, --verbose**

Gibt Meldungen während der Rotation aus.



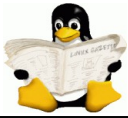
4. wichtige Direktiven

- compress** Alte Versionen der Logdateien werden komprimiert.
- compresscmd** Gibt das Kommando an, mit dem komprimiert werden soll.
Voreingestellt ist gzip
- uncompresscmd** Gibt das Kommando an, mit dem dekomprimiert werden soll.
Voreingestellt ist gunzip
- compressext** Gibt die Namensweiterung an, die für komprimierte Logdateien verwendet werden soll.
Die Voreinstellung ist abhängig vom verwendeten Kompressionsprogramm (.gz für gzip)
- compressoptions** Kommandozeilenoptionen für das verwendete Kompressionsprogramm.
Voreingestellt ist für gzip die Option -9 (maximale Kompression)
- copytruncate** Schneidet das originale Logfile auf die Länge 0 ab, nachdem eine Kopie gezogen wurde, anstatt das alte Logfile zu verschieben und ein neues zu erstellen.
Das kann verwendet werden, wenn ein Programm nicht in der Lage sein sollte, sein Logfile zu schließen und so ständig weiter in diese Datei schreibt. Es kann zu einer kurzen Verzögerung kommen, so daß einige Meldungen verloren gehen können.
Wenn diese Option benutzt wird, dann hat die **create** Option keine Bedeutung, weil die alte Logdatei ja bestehen bleibt.

create *Modus User Gruppe*

Direkt nach der Rotation (bevor das **postrotate**-Script abläuft) wird die Logdatei wieder erstellt (mit dem selben Namen, den die gerade rotierte Datei hatte).

Modus spezifiziert dabei den Zugriffsmodus in oktaler Form
(wie beim Kommando **chmod**),



User ist der User, der zum Eigentümer der Logdatei wird und *Gruppe* bestimmt die Gruppenzugehörigkeit der Datei.

Jedes dieser Attribute kann weggelassen werden, in diesem Fall wird das neue Logfile die selben Attribute bekommen, die schon das alte hatte. Diese Option kann durch die Verwendung von **nocreate** abgeschaltet werden.

daily Logdateien werden täglich rotiert.

delaycompress Verschiebt die Kompression des letzten Logfiles auf den nächsten Rotationszyklus. Das hat nur im Zusammenhang mit **compress** Bedeutung. Es kann verwendet werden, wenn Programme nicht dazu veranlasst werden können, ihre Logdateien zu schließen und daher noch in die alte Datei weiterschreiben wollen.

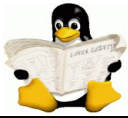
extension *Erweiterung*

Logdateien bekommen die abschließende Namensweiterung. Wenn die Kompression aktiviert ist, bekommen die Logdateien die Kompressionserweiterung (.gz bei gzip) nach der angegebenen *Erweiterung*.

ifempty Die Logdatei wird selbst dann rotiert, wenn sie leer ist. Das ist der voreingestellte Modus, der durch Verwendung von **notifempty** überschrieben werden kann.

include *Datei_oder_Verzeichnis*

Liest die angegebene Datei und führt die darin enthaltenen Befehle aus, als ständen sie an der Stelle, an der die **include** Anweisung steht. Wurde ein Verzeichnisname angegeben, so werden die meisten der in diesem Verzeichnis liegenden Dateien an dieser Stelle eingelesen. Ausgenommen sind nur die Dateien, die keine regulären Dateien sind (wie etwa weitere Verzeichnisse oder Pipes) oder die, deren Namen auf eine der Erweiterungen enden, die unter **tabooext** angegeben wurden. Die **include** Anweisung darf nicht innerhalb einer Logdatei-Definition stehen.



mail Adresse

Wenn eine Logdatei das letzte Mal rotiert wird (so dass sie danach nicht mehr existiert), wird sie an die angegebene Adresse gemailt. Wenn keine solche Mail für eine bestimmte Datei erzeugt werden soll, kann die **nomail** Direktive benutzt werden.

mailfirst

Wenn das **mail**-Kommando benutzt wird, wird die rotierte Datei statt der zu löschenden Datei gemailt.

maillast

Wenn das **mail**-Kommando benutzt wird, wird die zu löschende Datei gemailt.

(Das ist die Voreinstellung)

missingok

Wenn die Logdatei nicht existiert, soll ohne weitere Fehlermeldung mit der nächsten Logdatei weitergemacht werden.

Siehe auch **nomissingok**.

monthly

Logdateien werden rotiert, wenn **logrotate** das erste Mal in einem neuen Monat aufgerufen wird. Das ist normalerweise der erste Tag des Monats.

nocompress

Alte Versionen der Logdatei werden nicht komprimiert.

Siehe auch **compress**

nocopytruncate

Die Original-Datei wird nicht abgeschnitten, nachdem eine Kopie erstellt wurde.

Diese Direktive überschreibt **copytruncate**

nocreate

Neue Logdateien werden nicht angelegt.

Diese Direktive überschreibt **create**

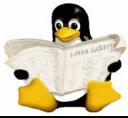
nodelaycompress

Die Kompression der letzten Datei wird nicht auf den nächsten Rotationszyklus verschoben.

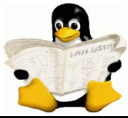
Diese Direktive überschreibt **delaycompress**

nomail

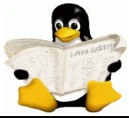
Alte Logdateien werden nicht gemailt



- nomissingok** Wenn eine Logdatei nicht existiert, wird eine Fehlermeldung erzeugt.
Das ist die Voreinstellung.
- noolddir** Logdateien werden in dem Verzeichnis rotiert, in dem auch das Original liegt.
Diese Direktive überschreibt **olddir**
- nosharedscripts** Die **prerotate** und **postrotate** Scripts werden für jede Logdatei ausgeführt, die rotiert wird. Das ist die Voreinstellung und überschreibt die **sharedscripts** Option.
- notifempty** Leere Logdateien werden nicht rotiert.
Diese Direktive überschreibt **ifempty**
- olddir Verzeichnis** Logs werden in das angegebene *Verzeichnis* verschoben um sie zu rotieren.
Das Verzeichnis muss sich auf dem selben physikalischen Gerät befinden, wie die Logdatei selbst.
Ist diese Option angegeben, dann enden alle alten Versionen des Logfiles in dem angegebenen Verzeichnis.
Diese Direktive überschreibt **noolddir**
- postrotate/endscript**
Die Zeilen zwischen **postrotate** und **endscript** (beide müssen in einer eigenen Zeile stehen) werden ausgeführt, nachdem die Logdatei rotiert wurde.
Diese Anweisung darf nur innerhalb einer Logdatei-Definition stehen (nicht global).
- prerotate/endscript** Die Zeilen zwischen **prerotate** und **endscript** (beide müssen in einer eigenen Zeile stehen) werden ausgeführt, bevor die Logdatei rotiert wurde.
Diese Anweisung darf nur innerhalb einer Logdatei-Definition stehen (nicht global).



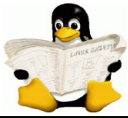
- rotate** *Anzahl* Logdateien werden *Anzahl* mal rotiert, bevor sie gelöscht oder gemailt werden.
Wenn *Anzahl* 0 ist, werden alte Versionen gelöscht, anstatt rotiert.
- sharedscripts** Normalerweise werden die **prerotate** und **postrotate** Scripts für jede einzelne Datei angewandt. Das heißt, ein Script wird womöglich mehrmals ausgeführt, wenn in einer Logfile-Definition mehrere Dateien oder Namensmuster (wie etwa `/var/log/news/*` in unserem Beispiel) angegeben wurden.
Wenn **sharedscripts** angegeben wurde, dann werden die Scripts nur einmal ausgeführt, egal wieviele Dateien auf das Namensmuster passen.
Ein Nebeneffekt dieser Anweisung ist, daß das Script auch ausgeführt wird, wenn keine Datei auf das Muster passt, also keine Dateien rotiert werden.
Wenn diese Option nicht angegeben wird, werden die Scripts nur ausgeführt, wenn die Logdateien tatsächlich rotiert werden.
- size** *Größe*[**M**|**k**] Logdateien werden rotiert, wenn sie größer als die angegebene Größe gewachsen sind.
Folgt der Größenangabe ein **M**, so werden Megabyte angenommen, mit einem **k** sind Kilobyte gemeint. Also sind **size 100**, **size 100M** und **size 100k** alles gültige Angaben.
- tabooext** [**+**] *Liste* Die aktuelle Liste der Taboo-Erweiterungen wird verändert. (Siehe die **include** Direktive für Informationen über die Taboo-Erweiterungen).
Ist ein **+** vor der Liste angegeben, so werden die angegebenen Erweiterungen an die bestehende Liste angehängt, ansonsten wird die bestehende Liste durch die angegebene Liste ersetzt.
Zu Beginn enthält die Taboo-Erweiterungsliste die folgenden Einträge:
.rpmorig, .rpmsave, .dpkg-dist,
.dpkg-old, .dpkg-new, .disabled,
.v, .swp, .rpmnew,
und ~.



Die Mitglieder der Liste werden durch Leerzeichen voneinander getrennt, nicht durch Kommas.

weekly

Logdateien werden rotiert, wenn der aktuelle Wochentag kleiner als der Wochentag der letzten Rotation ist, oder die letzte Rotation länger als eine Woche her ist.



5. Beispiele

5.1 /etc/logrotate.conf

```
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

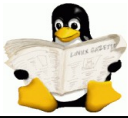
# uncomment this if you want your log files compressed
#compress

# uncomment these to switch compression to bzip2
#compresscmd /usr/bin/bzip2
#uncompresscmd /usr/bin/bunzip2

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp -- we'll rotate them here
#/var/log/wtmp {
#   monthly
#   create 0664 root utmp
#   rotate 1
#}

# system-specific logs may be also be configured here.
```



5.2 Script in /etc/logrotate.d

```
/var/log/USERLOG/bootlog.log
/var/log/USERLOG/login.log
/var/log/USERLOG/postgreSQL.log
/var/log/USERLOG/useradmin.log
{
    compress
    dateext
    maxage 365
    rotate 99
    missingok
    notifempty
    size +4096k
    create 640 root root
    sharedscripts
    postrotate
        /etc/init.d/syslog reload
    endscript
}
```

