


Глава 12. Системные журналы.

12.1. Служба rsyslog.


it
CLOUD
учебный центр

Служба rsyslog

- В современных дистрибутивах для системного журналирования обычно применяется rsyslogd
- Место хранения журналов обычно /var/log
- Помимо системных журналов приложения могут создавать свои журналы

В GNU/Linux принято сохранять информацию разной степени детализации о процессе работы программ в специальных текстовых файлах, называемых журналами.

Стандартное место расположения журналов - это каталог /var/log .

Все журнальные файлы принадлежат к одной из двух категорий: системные журналы и журналы прикладных программ.

Служба syslog (rsyslog) предназначена для обеспечения сохранения информации, поступающей от различных системных служб.

Некоторые службы ведут собственные журналы, которые не зависят от syslog, например веб сервер Apache.

Для таких служб принято создавать отдельные подкаталоги в /var/log.

Примечание: Она может быть также использована и прикладными программами, однако чаще всего прикладные программы, если уж возникает такая необходимость, ведут собственные журналы без службы syslog . При этом файлы журналов для таких программ обычно находятся где-либо в подкаталогах /var/log .

Примечание: Не все журналы, находящиеся в /var/log , обслуживаются службой syslog. Так, например, бинарный файл базы данных с информацией о последних входах в сеанс wtmp также находится в этом каталоге (имеется аналогичный файл /var/run/utmp с информацией о пользователях, находящихся в сеансе – см. команды who и last).

Syslog это не только служба для регистрации событий, но и протокол для передачи журнальных сообщений.

Настройка rsyslogd

- Основной конфигурационный файл `/etc/rsyslogd.conf`
- В файле определяете каналы поступления событий и каналы передачи
- Могут применяться шаблоны, например для автоматического формирования имени файла или формата сообщений
- Параметр `facility` определяет источник события
- Параметр `severity` уровень важности

Согласно протоколу syslog сообщения характеризуются двумя параметрами:

1. `facility` — источник события;
2. `severity` — важность события.

Определены следующие источники событий:

1. `LOG_AUTH` сообщения безопасности/авторизации
2. `LOG_AUTHPRIV` сообщения безопасности/авторизации (private)
3. `LOG_CRON` планировщик заданий (cron и at)
4. `LOG_DAEMON` системные службы без определенного значения объекта
5. `LOG_FTP` служба FTP
6. `LOG_KERN` сообщения ядра (не могут быть созданы пользовательскими процессами)
7. с `LOG_LOCAL0` по `LOG_LOCAL7` зарезервировано для локального использования
8. `LOG_LPR` подсистема печати
9. `LOG_MAIL` почтовая подсистема
10. `LOG_NEWS` подсистема новостей USENET
11. `LOG_SYSLOG` сообщения, сгенерированные самой службой syslog
12. `LOG_USER` (по умолчанию) обычные сообщения пользовательского уровня
13. `LOG_UUCP` подсистема UUCP

В других операционных системах могут быть определены и другие источники событий.

В качестве уровня событий используются:

Глава 12. Системные журналы.

1. LOG_EMERG система в нерабочем состоянии
2. LOG_ALERT необходимо срочное вмешательство
3. LOG_CRIT критические состояния
4. LOG_ERR ошибки
5. LOG_WARNING предупреждения
6. LOG_NOTICE обычные, но важные сообщения
7. LOG_INFO информационные сообщения
8. LOG_DEBUG сообщения уровня отладки

Уровни указаны в порядке убывания приоритета.

Служба syslog в большинстве современных Linux представлена демоном `rsyslogd` (The rocket-fast Syslog Server), запускаемым при старте операционной системы автоматически и работающем в фоновом режиме:

Пример:

```
$ ps -C rsyslogd
  PID TTY          TIME CMD
  862 ?            00:00:29 rsyslogd
```

Реже используется `syslog-ng`. Где-то может встретиться и другие реализации syslog.

Задачей демона `rsyslogd` является сбор сообщений от системных служб и сохранение этих сообщений в заранее известных файлах журналов.

Примечание: В GNU/Linux обычно сообщения, поступающие от различных служб, не записываются в один единственный журнал. Наоборот, принято называть файлы журналов так, чтобы по их названию можно было судить об источнике сообщений.

Конфигурационным файлом демона `rsyslogd` является `/etc/rsyslog.conf`.

Строки этого файла, начинающиеся с решетки `#` являются комментариями.

Проект `rsyslog` относительно молодой (начат в 2004 г.), поэтому в разных версиях `rsyslog` применяются разные форматы конфигурации.

Пример: проверим версию `rsyslog`:

```
$ rsyslogd -v | head -1
rsyslogd 8.2010.0 (aka 2020.10) compiled with:
```

`Rsyslog` модульная система, это означает, что вам необходимо для начала определить какие модули вы будете использовать.

Модули делятся на две категории:

1. Входящие (Input) с префиксом `im` в названии. Такие модули открывают канал поступления информации.
2. Исходящие (Output) — `om`. Эти модули позволяют передавать сообщения в разные каналы.
3. Парсер (Parser) — `pm`. Разбирают содержимое сообщения на части.
4. Модификаторы сообщений (Message Modification) — `mm`. Используются для

Глава 12. Системные журналы.

изменения содержимого сообщения.

5. Генераторы строки (String Generator) — sm. Генерируют строки на основе содержимого сообщения.

6. Библиотечные модули (Library). Часть самого rsyslog загружаются автоматически.

Пример:

```
$ grep module /etc/rsyslog.conf
module(load="builtin:omfile" Template="RSYSLOG_TraditionalFileFormat")
module(load="imuxsock"      # provides support for local system logging (e.g. via
logger command)
module(load="imjournal"      # provides access to the systemd journal
#module(load="imklog") # reads kernel messages (the same are read from journald)
#module(load="immark") # provides --MARK-- message capability
#module(load="imudp") # needs to be done just once
#module(load="imtcp") # needs to be done just once
```

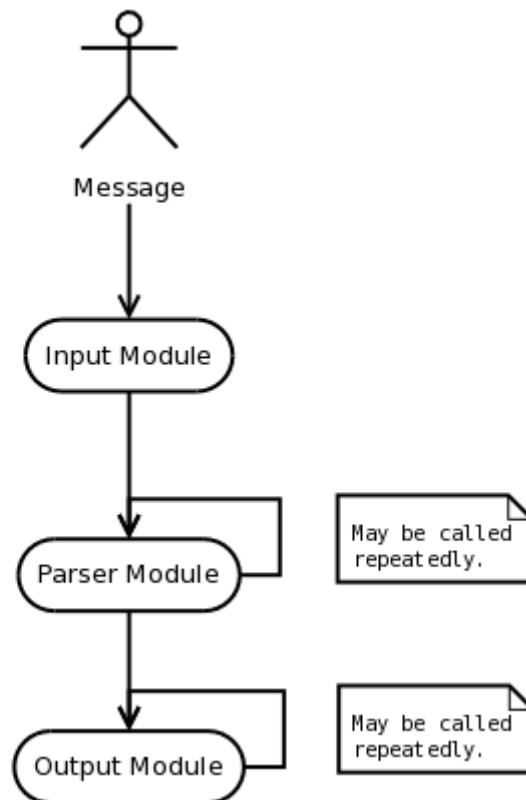


Рисунок 1: Обработка сообщений

Структура строк, каждая из которых направляет некоторый поток сообщений в заданный файл (или на удаленный компьютер - сервер ведения журналов), представлена двумя полями:

1. Определение сообщения (selector) - поле, в котором указывается от каких классов программ должны собираться сообщения в данный поток. И какие именно сообщения.
2. Поле действия (action), указывающее куда должен быть записан поток сообщений. Чаще всего - это имя файла журнала в `/var/log`.

Пример:

```
$ grep '/var/log/' /etc/rsyslog.conf
```

Глава 12. Системные журналы.

*.info;mail.none;authpriv.none;cron.none	/var/log/messages
authpriv.*	/var/log/secure
mail.*	-/var/log/maillog
cron.*	/var/log/cron
uucp,news.crit	/var/log/spooler
local7.*	/var/log/boot.log

Примечание: Из приведенного выше примера заметно, что поле определения сообщения состоит из двух частей, разделенных точкой. Первая часть - источник сообщения (facility), а вторая - уровень важности (priority) сообщения. Эти две части уникально определяют все возможные сообщения, обрабатываемые syslog.

Знак звездочка является метасимволом, обозначающим либо все источники, если он указан перед точкой – разделителем, либо все уровни важности, если этот метасимвол установлен после точки.

При указании источника сообщения и уровня важности, разделенных точкой, определяется, что сообщения, поступающие от этого источника и имеющие указанный и вышележащие уровни важности, будут записаны в данный канал.

Пример:

uucp,news.crit	/var/log/spooler
----------------	------------------

При необходимости запретить запись в какой-либо канал сообщений с заданным уровнем важности и выше, можно использовать знак восклицания перед уровнем важности.

Пример:

daemon.info;daemon.!err	-/var/log/daemons
-------------------------	-------------------

Примечание: В этом случае все сообщения от демонов с уровнями важности от info до warning будут записываться в журнал /var/log/daemons, а сообщения с уровнями важности, начиная с err, записаны туда не будут.

Если же необходимо записывать в журнал сообщения только с определенным уровнем важности и ни с какими другими более, то перед требуемым уровнем важности следует поставить знак равно.

Пример:

daemon.=err	/var/log/daemons.err
-------------	----------------------

Примечание: В файл /var/log/daemons.err будут записываться только сообщения об ошибках в работе демонов.

Для исключения из потока сообщений тех из них, которые имеют заданный уровень важности используют восклицательный знак и знак равенства.

Пример:

kern.*;kern.!=info	/var/log/kernel
--------------------	-----------------

Примечание: Здесь в журнал будут записываться все сообщения от ядра, кроме информационных.

Если в канал не должны быть записаны любые сообщения от каких-либо источников, то удобно использовать директиву none :

Пример:

*.crit;lpr,cron,mail.none	/var/log/critical
---------------------------	-------------------

Примечание: При этом в файл `/var/log/critical` будут записываться сообщения от всех источников о критических и более важных событиях, кроме любых сообщений от служб печати, почты, `at` и `cron`.

Rsyslog поддерживает и другие фильтры записи событий в журналы.

1. Property-Based Filters — фильтр, который анализирует различные свойства сообщений.

2. Expression-Based Filters — фильтр, в котором вы можете определять различные логические выражения всестороннего для анализа сообщений.

Правила фильтрации сообщений могут быть объединены в наборы правил (RuleSet). Наборы правил помогают более эффективно управлять потоком информации.

Для составления сообщений в правильном формате или для формирования имен файлов вы можете использовать шаблоны (Templates)

Пример: разделение локальных сообщений и сообщений из сети, причем имена файлов куда записываются сообщения формируются на основе имени узла, которое это сообщение прислало.

```
# cat /etc/rsyslog.conf | egrep -v '^(#|$)'  
$ModLoad imuxsock # provides support for local system logging (e.g. via logger  
command)  
$ModLoad imklog    # provides kernel logging support (previously done by rklogd)  
$ModLoad imudp  
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat  
$RuleSet local  
kern.*                                /dev/console  
*.info;mail.none;authpriv.none;cron.none /var/log/messages  
authpriv.*                          /var/log/secure  
mail.*                              -/var/log/maillog  
cron.*                              /var/log/cron  
*.emerg                             *  
uucp,news.crit                      /var/log/spooler  
local7.*                            /var/log/boot.log  
$DefaultRuleset local  
$template RemHost, "/var/log/network/%HOSTNAME%.log"  
$RuleSet remote_udp  
*.* ?RemHost  
$InputUDPServerBindRuleset remote_udp  
$UDPServerRun 514
```

12.2. Журналы systemd



Журналы systemd

- Файл `/etc/systemd/journald.conf` определяет параметры журналирования systemd
- Для анализа журналов используется команда `journalctl`
- Журналы systemd бинарные

Система systemd собирает сведения о своем функционировании в бинарные файлы.

Файл `/etc/systemd/journald.conf` определяет параметры журналирования.

Важнейшая опция `Storage` в этом файле определяет способ ведения журнала постоянный (`persistent`), непостоянный (`volatile`) или никакой (`none`).

Команда `journalctl --header` показывает в том числе название файла в который в данный момент записывается информация.

Каждый раз при старте системы создается новый журнал.

Команда `journalctl` выводит текущий журнал с начала с самого раннего события.

Полезные опции:

1. `-e` — показывает журнал с конца.
2. `-x` — показывает объяснения событий, если они есть.
3. `-u юнит` — показывать события связанные с юнитом.
4. `-f` — показывать вновь поступающие сообщения (аналогично `tail -f`).

12.3. Служба ротации журналов.



Служба ротации журналов

- Программа `logrotate` используется для ротации журналов
- Обычно `logrotate` запускается по расписанию демоном `cron`
- Файла конфигурации `logrotate` - `/etc/logrotate.conf`
- В `/var/lib/logrotate/logrotate.status` содержится статус ротации журналов

С течением времени накапливающиеся сообщения в файлах журналов могут переполнить файловую систему.

Для предотвращения этого предназначена программа `logrotate`, обеспечивающая ротацию журналов. Стандартный путь ее вызова – использование ее, как ежедневного задания `crond`.

Пример:

```
$ cat /etc/cron.daily/logrotate

#!/bin/sh
exec /usr/sbin/logrotate /etc/logrotate.conf
```

***Примечание:** В данном примере показано, что скрипт вызова команды `logrotate` находится в каталоге `/etc/cron.daily`, задания в котором выполняются каждую ночь (это настройки данной системы). Файл `/etc/logrotate.conf` содержит настройки для этой утилиты.*

Утилита `logrotate` способна производить следующие действия с файлами журналов:

1. Удалять.
2. Переименовывать.
3. Сжимать с помощью программ – компрессоров.
4. Создавать новые пустые файлы журналов.
5. Посылать ротируемые файлы журналов по электронной почте.

Пример: Ротации с заданной периодичностью файла `/var/log/messages` :

```
# ls -w 1 /var/log/messages*
```


Глава 12. Системные журналы.

```
/var/log/messages  
/var/log/messages.1.bz2  
/var/log/messages.2.bz2  
/var/log/messages.3.bz2  
/var/log/messages.4.bz2  
/var/log/messages.5.bz2
```

Примечание: При наступлении момента времени, когда необходимо осуществить первую ротацию файл `messages` переименовывается в `messages.1.bz2` (в данном примере используется компрессия журналов утилитой `bzip2`).

При второй ротации файл `messages.1.bz2` переименовывается в `messages.2.bz2`, а файл `messages` переименовывается в `messages.1.bz2`.

При третьей ротации файл `messages.2.bz2` переименовывается в `messages.3.bz2` и так далее ...

Утилита `logrotate` удаляет архивные копии старых журналов по достижении заданного количества копий. В этом примере ротация первой архивной копии журнала (файл `messages.1.bz2`) осуществляется четыре раза (до `messages.5.bz2`).

Файла конфигурации `logrotate` - `/etc/logrotate.conf`.

Настройки, находящиеся в начале файла `/etc/logrotate.conf` и не связанные с именами файлов журналов, являются глобальными.

Для каждого конкретного файла журнала можно указывать отдельные настройки.

Обычно применяются следующие настройки:

1. `daily`, `weekly` и `monthly` определяют периодичность ротации равной, соответственно, одному дню, неделе или месяцу.
2. Настройка `rotate` определяет количество ротаций первой архивной копии журнала до ее удаления.
3. `create` заставляет создавать пустой файл журнала после его ротации. Причем, настройка `create` позволяет указывать права доступа и владения создаваемых журнальных файлов.
4. `compress` сжатие ротированных файлов. Если не надо сжимать файлы архивных копий, то следует использовать настройку `nocompress`.
5. `copy` копировать файлы оставляя при этом оригинальные файлы журналов нетронутыми.
6. `notifempty` позволяет не осуществлять ротацию пустых файлов.
7. `include` позволяет включать в файл конфигурации дополнительные настройки, указанные в файле – аргументе этой директивы. Если аргументом является каталог, то в основной файл конфигурации включается содержимое всех конфигурационных файлов, находящихся в этом каталоге.
8. `mail` позволяет получать копии ротируемых журнальных файлов по электронной почте.
9. `prerotate` и `postrotate` позволяют указывать скрипты, которые будут исполнены, соответственно, до и после ротации.

Глава 12. Системные журналы.

10.size можно указывать размер файла журнала, по превышении которого должна осуществляться его ротация.

Пример:

```
weekly
rotate 4
create
compress
notifempty
include /etc/logrotate.d
/var/log/wtmp {
    monthly
    create 0664 root utmp
    rotate 4
}
```