ФЕДЕРАЛЬНОЕ АГЕНТСТВО ОБРАЗОВАНИЯ
ДИМИТРОВГРАДСКИЙ ИНСТИТУТ ТЕХНОЛОГИЙ,
УПРАВЛЕНИЯ И ДИЗАЙНА

Лабораторная работа №4
по курсу: “Протоколирование в ОС Linux”

Выполнил студент гр. ВТ-41:
Потеренко А.Г.
Проверил преподаватель:
Петлинский В.П.

Димитровград 2006г.

В этой работе будет рассмотрен демон syslogd, а также как управлять протоколированием сообщений системы и ядра с помощью этого демона.

Прежде всего, нужно отметить, что демон находится в пакете sysklogd, поэтому перед его использованием нужно установить этот пакет. В большинстве случаев у вас пакет уже будет установлен, а демон syslogd - запущен.

```
[root@DBADOMAIN ~]# syslogd
syslogd: Already running.
```

В пакет sysklogd на самом деле входят две программы: **syslogd** и **klogd**. Syslogd отвечает за протоколирования сообщений системы, а klogd - ядра.

Демон Syslogd

Syslogd обеспечивает вид протоколирования, который используется большинством программ. Демон syslogd пишет сообщения в файл **/var/log/syslog**. Обычно записи в этом файле содержат такие поля: дата и время, хост, программа, сообщение.

Демон syslogd запускается автоматически при старте системы. Для его запуска предназначен сценарий **/etc/rc.d/init.d/syslog**. Как обычно, запустить демон самостоятельно мы можем с помощью команды:

```
[root@DBADOMAIN log]# /etc/rc.d/init.d/syslog start
Запускается служба журналирования системы:                 [  OK  ]
Запускается служба журналирования ядра:                    [  OK  ]
```

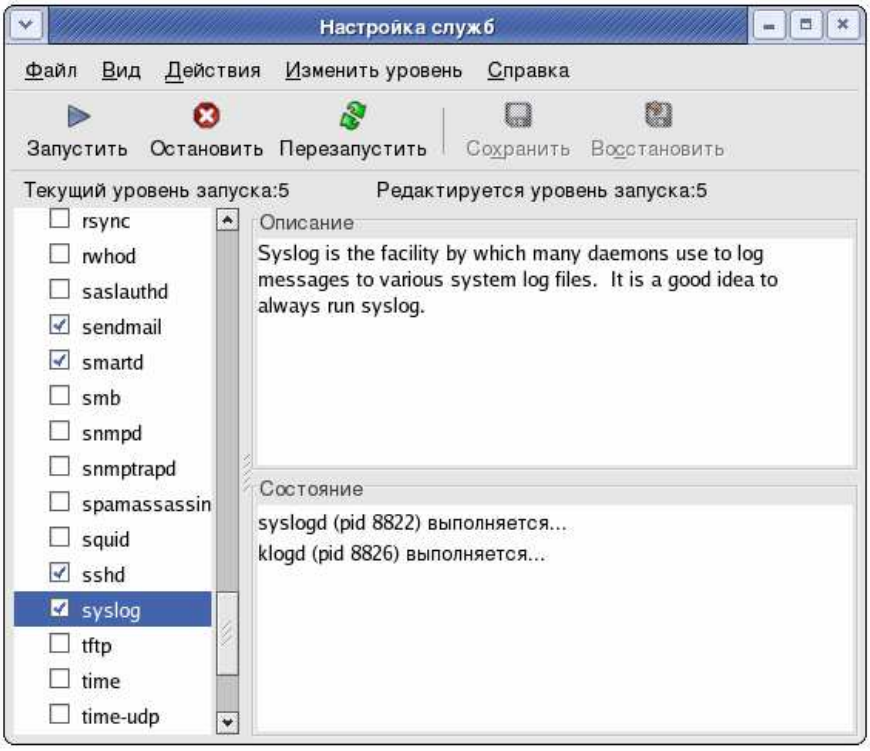
а остановить –

```
[root@DBADOMAIN log]# /etc/rc.d/init.d/syslog stop
Останавливается служба журналирования ядра:                [  OK  ]
Останавливается служба журналирования системы:             [  OK  ]
```

Для обеспечения автоматической загрузки нужно создать символическую ссылку на этот файла, например:

```
ls -s /etc/rc.d/rc5.d/@S30syslog /etc/rc.d/init.d/syslog
```

В этом случае мы обеспечим запуск демона на пятом уровне запуска (автоматический запуск X Window). То же самое можно выполнить, используя сервисные утилиты с графическим интерфейсом.



Параметры запуска демона ***syslogd***

```
syslogd [ -a socket ] [ -d ] [ -f config file ] [ -h ] [ -l hostlist ]
      [ -m interval ] [ -n ] [ -p socket ] [ -r ] [ -s domainlist ] [ -v ] [ -x ]
```

Опция	Описание
-a socket	Этот параметр позволяет указать дополнительный сокет, который syslogd должен прослушивать
-d	Включает режим отладки. В этом режиме демон не будет использовать системный вызов fork(2) для переключения себя в фоновый режим и будет выводить больше отладочной информации
-f file	Этот параметр определяет альтернативный файл конфигурации
-h	По умолчанию демон не перенаправляет сообщения, которые он получает от других узлов. Этот параметр позволяет перенаправить сообщения другим хостам, которые определены
-n	Этот параметр нужен, если syslogd запускается и контролируется программой init
-p socket	Позволяет задать другой сокет Unix вместо /dev/log
-r	Позволяет принимать сообщения из сети. Данная опция появилась в версии syslogd 1.3
-v	Выводит версию демона syslogd

Сигналы

Демон syslogd реагирует на следующие сигналы: SYGTERM, SIGINT, SIGQUIT, SIGHUP, SIGUSR1, SIGCHLD.

Сигнал	Реакция
SIGTERM	Завершает работу демона
SIGINT, SIGQUIT	Завершает работу демона, если выключена отладка (debugging). Если же отладка включена, эти сигналы игнорируются
SIGUSR1	Включает/выключает отладку
SIGHUP	Перезапуск демона

Файл конфигурации

По умолчанию используется файл конфигурации **/etc/syslog.conf**. Вы можете указать другой файл конфигурации с помощью опции -f. Указания по каждому виду протоколирования задаются в виде отдельных строк файла. В файле можно записывать комментарии, которые определяются по первому символу # в строке.

```
[root@DBADOMAIN log]# more /etc/syslog.conf
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console
# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none                /var/log/messages
# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure
# Log all the mail messages in one place.
mail.*                                                  -/var/log/maillog
# Log cron stuff
cron.*                                                  /var/log/cron
# Everybody gets emergency messages
*.emerg                                                 *
# Save news errors of level crit and higher in a special file.
uucp,news.crit                                          /var/log/spooler
# Save boot messages also to boot.log
local7.*                                                /var/log/boot.log
```

Каждая строка файла представляет собой набор правил маршрутизации сообщений. Каждое правило состоит из **селектора** и **действия**, которые разделяются табуляциями (в старых системах - Solaris 5) или пробелами (Linux). Получив сообщение для записи в журнал (от klogd, от локальной или удаленной программы), syslogd для каждого правила проверяет не подходит ли сообщение под шаблон, определяемый селектором. Если подходит, то выполняется указанное в правиле действие. Для одного сообщения может быть выполнено произвольное количество действий (т.е. обработка сообщения не прекращается при первом успехе). Таким образом, файл /etc/syslog.conf состоит из двух столбцов. В первом указывается правило отбора записей для журнала. Во втором содержится описание действий, которые будут предприняты для обработки подошедшей записи. Большинство затруднений вызывает полное понимание того, как точно указать правило отбора журналируемых записей.

Задание селектора

Источник журналируемых записей описывается указанием категории **(facility)** и уровня **(level)**. Категория это или источник записей, или программа, которая шлет сообщения демону syslogd. Правило отбора записывается в первом столбце строки в виде - **facility.level**. Категории сообщения**(facility)** задаются ключевыми словами:

- kern - Сообщения от ядра.
- user - Предназначена для сбора разнообразных сообщений. Если не указать категорию журналирования для программ пользователя, то они будут использовать эту категорию.
- mail - Сообщения от почтовой системы.
- daemon - Ловушка для сообщений от всех остальных системных демонов, которые не имеют явно описанных категорий.
- auth - Все, что связано с авторизацией пользователей, вроде login и su.
- syslog - Система журналирования может журналировать сообщения от самой себя.
- lpr - Сообщения от системы печати.
- news - Сообщения от сервера новостей.
- uucp - Собирает сообщения от UNIX-to-UNIX Copy Protocol. До сих пор определенная часть почтовых сообщений доставляется через UUCP.
- cron - Сообщения от системного планировщика.
- authpriv - Тоже самое, что и auth, однако пишет журнал в файл, который могут читать лишь некоторые пользователи (сообщения, собираемые в этой категории могут содержать открытые пароли пользователей, которые не должны попадать на глаза посторонним людям, и, следовательно, файлы журналов должны иметь соответствующие права доступа).
- ftp - При помощи этой категории можно сконфигурировать ваш FTP сервер, что бы он записывал свои действия.
- Ntp - Сообщения от сервера точного времени.
- console - Сообщения, обычно печатаемые на системной консоли, могут быть записаны в журнал при помощи этой категории.
- mark - Эта категория используется для того, что бы помещать в журнал сообщение каждые 20 минут. Она может быть полезна в комбинации с некоторыми другими журналами (например, можно узнать с 20-ти минутной точностью, когда же завис ваш сервер).
- security - Сообщения от различных служб безопасности, таких как ipfw или ipf.
- LOCAL0 - LOCAL7 - зарезервированы для локального использования.

Большинство систем записывают далеко не все, что сообщают их программы - зачастую незначительные сообщения отбрасываются, а записываются только важные события. Однако то, что кажется одному человеку незначительным, другому может показаться существенным. Здесь мы встречаемся с уровнями подробности сообщений - **level**.

Linux предоставляет **восемь уровней важности сообщений**. С их помощью, вы можете сообщить syslog, что записывать в журнал, а что отбросить. Вот эти уровни, в порядке уменьшения важности (уровень серьезности при программировании кодируется числом от 0 до 7):

- emerg - Система в панике. Сообщения немедленно выводятся на все активные терминалы. Система обычно накрывается медным тазом, или остается чрезвычайно, чрезвычайно нестабильной. Продолжение работы невозможно.
- alert - Это плохо, но не настолько плохо как уровень emerg. Система может продолжить работу, но эту ошибку следует устранить немедленно.
- crit - Это критические ошибки, такие как проблемы с аппаратным обеспечением или серьезные нарушения работы программного обеспечения. Если ваш жесткий диск содержит плохие блоки, они проявятся в виде критических ошибок. Если вы очень смелый, попробуйте продолжить работу.
- err - Разнообразные ошибки. Это скверно, такие ошибки должны быть устранены, но они не разрушат вашу систему.
- warning - Разнообразные предупреждения.
- notice - Общая информация, которая должна быть записана, если она вам нужна, но вероятно она не потребует вашей реакции.
- info - Различная системная информация.
- debug - Этот уровень обычно используется программистами и иногда системными администраторами, которые пытаются понять - почему же эта программа так поступает? Отладочные сообщения могут содержать всю информацию, которую счел необходимым вывести ее разработчик для отладки кода; между прочим, она может содержать данные, нарушающие приватность ваших пользователей.
- none - Это специальный уровень означающий - 'ничего не записывать в данной категории'. Он обычно применяется для исключения информации из групповых записей. Номер источника умножается на 8 и складывается с уровнем серьезности, получившееся число заключается в угловые скобки и образует поле PRI.

Описание правила отбора источника информации включает в себя категорию и уровень детализации, разделенные точкой. Когда вы указываете уровень, по умолчанию в журнал записываются сообщения, уровень которых выше или равен указанному. В качестве примера рассмотрим эту запись из файла /etc/syslog.conf:

```
mail.info /var/log/maillog
```

В журнал /var/log/maillog будут записаны сообщения от почтовой системы, с уровнем выше или равным уровню info.

Если возникнет потребность, то вы можете воспользоваться символом '*' в описании журналируемого источника. Например, для записи абсолютно всех сообщений от почтовой системы вы можете воспользоваться следующим правилом:

```
mail.* /var/log/maillog
```

Задание действия

Действие задается во втором столбце строки. Оно определяет куда отправить сообщение и распознается по первому символу описателя действия. Возможны следующие действия:

- / - запись в обычный файл (символ слэш).
- | - запись в именованный канал - файл типа FIFO.
- /dev/ttyi или /dev/console - запись в ту или иную консоль (файл - устройство).
- @ - запись на удаленный хост по сети. Имя или IP-адрес хоста указывается после символа @.
- При отсутствии любого из перечисленных выше символов рассматривается, как список учетных записей пользователей, присутствующих в системе, которым будет послано сообщение. Учетные записи в списке разделяются символом ",".
- * - Сообщение отправляется всем зарегистрированным пользователям.

Сетевое протоколирование

Сейчас более подробно разберем как обеспечить протоколирование в сети. Это означает перенаправление сообщений на демон syslogd, запущенный на другой машине, где они будут записаны на диск.

Для передачи сообщений используется протокол UDP. Он менее надежный, чем TCP, но отправление пакетов происходит несколько быстрее. Убедитесь, что в вашем файле /etc/service раскомментирована строка

```
syslog 514/udp
```

Затем нужно внести некоторые коррективы в наш файл конфигурации. Как и прежде, определите объекты протоколирования, а вместо файлов протоколов используйте параметр @hostname, где hostname - это имя компьютера, на который будут перенаправлены сообщения. Например, для перенаправления всех сообщений об ошибках на узел сети hostname можно использовать такую запись:

```
*.err      @hostname
```

Для перенаправления всех сообщений используется запись:

```
*.*        @hostname
```

Имя узла желательно указать в файле /etc/hosts, так как демон syslogd может быть запущен после сервера доменных имен или сервер DNS окажется недоступным.

Вы можете организовать центральный сервер протоколирования для всей вашей локальной сети. Для того чтобы указать, какие хосты вы хотите протоколировать, используйте опцию

```
-l список_хостов
```

В списке указываются простые имена машин, то есть без указания имени домена. Имена машин разделяются двоеточием (:). Возможно, вы захотите использовать опцию -s для указания дополнительного сокета для прослушивания. Для перенаправления сообщений используйте опцию -r на машине-клиенте для перенаправления сообщений на сервер.

Демон klogd

Демон klogd предназначен для перехвата и протоколирования сообщений ядра Linux.

```
klogd [ -c n ] [ -d ] [ -f fname ] [ -iI ] [ -n ] [ -o ] [ -p ] [ -s ]
      [ -k fname ] [ -v ] [-x ] [ -2 ]
```

Опция	Описание
-c n	Устанавливает уровень сообщений, которые будут выводиться на экран
-d	Режим отладки
-f fname	Записывать сообщения в указанный файл раньше демона syslogd
-i	Позволяет перезагрузить символьную информацию ядра о модулях.
-I	Перезагружает статическую символьную информацию и информацию о модулях ядра
-n	Не переходить в фоновый режим. Этот параметр используется, когда демон управляется программой init
-o	Демон читает и протоколирует все сообщения, которые он найден в буферах сообщений ядра. После одно цикла чтения/протоколирования демон завершает работу
-s	Заставляет демон klogd использовать системные вызовы для обращений к буферам сообщений ядра
-k file	Использует указанный файл в качестве файла, содержащего символьную информацию ядра
-v	Выводит версию и завершает работу

Для просмотра сообщений ядра используется команда **dmesg**. Обычно она используется так:

```
dmesg | less
```

Данная программа выводит сообщения ядра при запуске системы. С помощью параметра -c этой программы можно очистить ring-буфер ядра. Параметр -n задает уровень сообщений, которые будут выводиться на консоль.

По умолчанию демон klogd вызывается системным вызовом для того, чтобы препятствовать отображению всех сообщений на консоль. Это не распространяется на критические сообщения ядра (kernel panic). Эти сообщения все равно будут отображены на консоли.

Демон реагирует на сигналы: SIGHUP, SIGKILL, SIGINT, SIGTERM, SIGTSTP, SIGUSR1, SIGUSR2, SIGCONT. Сигналы SIGTSTP и SIGCONT используются для начала и завершения протоколирования сообщений ядра. Сигналы SIGUSR1 и SIGUSR2 аналогичны опциям -i и -I соответственно. То есть первый перезагружает информацию о модулях, а второй статическую информацию и информацию о модулях. Использовать сигнал GIGUSR1 (как и все остальные) можно так:

```
kill -USR1 PID
```

Параметры ядра

Параметр debug ядра Linux задает уровень отладки. Сообщения ядра (важные и не очень) передаются через функцию printk(). Если сообщение очень важно, его копия будет передана на консоль, а также демону klogd для его регистрации на жестком диске. Сообщения передаются на консоль, потому что иногда невозможно запротоколировать сообщение на жестком диске (например, отказ диска). Предел того, что будет отображаться на консоли, задается переменной console_loglevel. По умолчанию на консоли отображается все, что выше уровня DEBUG (7). Список уровней можно найти в файле **kernel.h**.