

Запуск задач по расписанию

Использование планировщика atd

at 23:30

reboot

<Ctrl+D>

Одним из средств, позволяющих запустить какую-либо задачу в соответствии с расписанием, является сервис "atd". Он реализован в виде службы, которая управляется юнитом "atd.service" и предназначен для одноразового запуска заданий. Это означает, что после выполнения задания "atd" сам удаляет его из очереди, благодаря чему нет необходимости думать о его дальнейшей судьбе. Сначала убедимся, что сервис "atd" активен (если это не так, отложенный запуск задач работать, разумеется, не будет)

```
[demo@instructor ~]$ systemctl is-enabled atd.service
```

Для управления заданиями предусмотрены специальные команды, с помощью которых можно добавлять задания в очередь запланированных, просматривать очередь задач и их содержимое, а также отменять их в случае, если надобность в задании отпала.

По-умолчанию считается, что любой пользователь системы может планировать себе запуск заданий через службу "atd". Однако, администратор может переопределять это поведение. Для этого в системе присутствует файл "/etc/at.deny". Изначально он пустой. Если необходимо каким-либо пользователям запретить использование службы "atd", то имена (логины) этих пользователей добавляются в данный файл по одному на строку.

Можно поступить иначе: удалить файл "/etc/at.deny" и вместо него создать файл "/etc/at.allow", содержащий список пользователей. В этом случае использовать службу "atd" смогут только те, пользователи, которые перечислены в файле "at.allow".

Если оба файла отсутствуют, то использовать сервис "atd" может только администратор.

Для создания запускаемой по расписанию задачи используется команда `at`.

Например:

```
[demo@instructor ~]$ at 23:30
[demo@instructor ~]$ at 23:30 tomorrow
[demo@instructor ~]$ at now +5min
[demo@instructor ~]$ at now +2weeks
[demo@instructor ~]$ at 4pm September 1 2018
```

Команда `at` имеет довольно большое количество спецификаций, позволяющих описывать дату и время выполнения. Полный список этих спецификаций хранится в файле `/usr/share/doc/at-<версия_пакета>/timespec`

После вызова команды `at` (например, `at 23:30`) приглашение командной строки меняется на `"at>"`. Это означает, что утилита ожидает ввода команд, которые сервис `"atd"` затем исполнит в 23:30. Команд может быть несколько, окончанием ввода является комбинация «Ctrl-D», нажатая в пустой строке.

Если необходимо запустить по расписанию какой-либо сценарий, можно сделать это следующим образом:

```
[demo@instructor ~]$ at now +1hour -f /usr/local/bin/test.sh
```

Если запускаемые программы предполагают вывод какой-либо информации, имейте в виду, что на экране она не появится. Весь вывод пересылается в локальный почтовый ящик пользователя и может быть просмотрен, например, командой `mail`.

Отключить отправку вывода на почту можно с помощью опции `"-M"`:

```
[demo@instructor ~]$ at -M now +30min
```

Посмотреть список запланированных задач можно с помощью команды `atq`.

Команда отображает дату и время выполнения, пользователя и уникальный идентификатор задания. Посмотреть содержимое задания можно с помощью команды `at` с опцией `"-c"`, указав в качестве параметра идентификатор, например, так:

```
[demo@instructor ~]$ at -c 1 | tail
```

Сами задания хранятся в отдельных файлах в директории `/var/spool/at`

Удалить задание можно с помощью команды `atrm`, передав ей в качестве параметра уникальный идентификатор. Например:

```
[demo@instructor ~]$ atrm 5
```

Задание 1

1) Убедитесь, что в директории /tmp отсутствует файл с именем "date.out"

2) С помощью планировщика atd создайте задание, которое спустя две минуты создаст в директории /tmp файл с именем date.out, содержащий результат работы команды "date". Это можно сделать, не прибегая к интерактивному вводу команды, например, так:

```
[demo@instructor ~]$ echo "date >> /tmp/date.out" | now +2min
```

3) Убедитесь, что в очереди есть запланированное задание

```
[demo@instructor ~]$ atq
```

4) Подождите 2 минуты

```
[demo@instructor ~]$ sleep 120
```

5) Проверьте содержимое директории /tmp. Если всё было сделано правильно, что в ней должен был появиться файл date.out, содержащий результат работы команды date

Использование планировщика cron

crontab -e

```
.----- minute (0 - 59)
| .----- hour (0 - 23)
| | .----- day of month (1 - 31)
| | | .----- month (1 - 12) OR jan,feb,mar,apr ...
| | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR
sun,mon,tue,wed,thu,fri,sat
| | | | |
* * * * * command to be executed
```

Второй планировщик, применяемый в нашей системе, носит название cron и используется для циклического выполнения заданий. После выполнения задания, он не удаляет его из очереди, как это делает atd. Пользователи системы могут управлять планированием заданий с помощью команды crontab.

- crontab -l — отобразить все запланированные задания для текущего пользователя
- crontab -r — удалить все запланированные задания для текущего пользователя
- crontab -e — редактировать список заданий для текущего пользователя

Пользователь root также может использовать опцию «-u <имя_пользователя>», чтобы назначить задание не самому себе, а другому пользователю системы.

При редактировании заданий запускается используемый по-умолчанию редактор (обычно это vim). Файл с заданиями допускает использование комментариев, обозначаемых, как обычно символом "#" и пустых строк.

Если запускаемые программы предполагают запись данных в стандартные потоки вывода, то результат не отобразится на экране. Вместо этого он будет отправлен на e-mail. (Для этой цели используется локальный почтовый сервер).

Сами файлы с заданиями хранятся в директории /var/spool/cron. Имя файла с заданиями соответствует имени пользователя, например /var/spool/cron/user

- Значение минут может лежать в диапазоне от 0 до 59
- Значение часов может лежать в диапазоне от 0 до 23
- Значение дней месяца может лежать в диапазоне от 1 до 31
- Значение месяцев может лежать в диапазоне от 1 до 12 (допускается использовать сокращенные названия: Jan, Feb и т.д.)
- Значение дней недели может лежать в диапазоне от 0 до 7. Единица соответствует понедельнику (допускается использовать сокращенные названия: Mon, Tue и т.д.)

При этом в каждом из первых пяти полей, помимо цифры могут находиться следующие символы:

- * Означает: «любое из возможных значений»
- 1,3 Означает список. Т.е. если эти цифры стоят в поле «дни недели», то это подразумевает понедельник и среду
- 2-5 Означает диапазон. Т.е. если эти цифры стоят в поле «месяцы», то это подразумевает февраль, март, апрель и май
- */х Означает, что учитываться будут значения, которые нацело делятся на х. Например, */5 говорит о том, что задание будет запускаться каждые 5 минут.

Если поля «день месяца» и «день недели» оба имеют значения, отличные от «*», то задание будет выполняться, при совпадении с любым из них. Т.е. если в расписании указано «30 16 1,15 * 5», то такое задание будет выполняться в 16:30 первого и 15-го числа каждого месяца, плюс каждую пятницу.

Примеры заданий

Задание	Описание
<code>*/10 * * * * /usr/local/stat.sh</code>	Раз в десять минут
<code>0 23 * * 5 /usr/local/back.sh</code>	Каждую пятницу в 23:00
<code>30 6,22 * * 1-5 /data/check.sh</code>	В 6:30 и 22:30 по рабочим дням
<code>59 23 31 12 * mail -s "happy NY" user < message.txt</code>	Один раз в год, 31го декабря в 23:59

Администратор системы может ограничивать пользователей в применении сервиса cron. Для этого предусмотрены два файла: `/etc/cron.allow` и `/etc/cron.deny`. Работает механизм ограничений следующим образом:

- Если в системе присутствует только файл `/etc/cron.allow`, то планировать задачи с использованием cron могут только пользователи, перечисленные в этом файле.
- Если в системе присутствует только файл `/etc/cron.deny`, то планировать задачи с использованием cron могут все пользователи, кроме перечисленных в этом файле. (Если файл пуст, то использовать сервис могут все пользователи).
- Если оба файла отсутствуют, то использовать сервис cron может только администратор.

О добавлении, удалении и изменении заданий, а также о результатах их выполнения можно узнать из специального log-файла, который ведёт сервис cron: `/var/log/cron`

Помимо запланированных пользователями заданий, могут быть определены системные задания, которые добавляются непосредственно в конфигурационный файл без использования команды `crontab`. Файл с системными заданиями располагается в директории `/etc` и называется `crontab`. Формат файла `/etc/crontab`:

Минуты	Часы	Дни_месяца	Месяцы	Дни_недели	Пользователь	Команда
--------	------	------------	--------	------------	--------------	---------

Устанавливаемые приложения (если они используют запускаемые по расписанию задания) вместо редактирования `/etc/crontab` могут просто создавать свои собственные файл в директории `/etc/cron.d`. Формат файлов, создаваемых в `/etc/cron.d` аналогичен формату файла `/etc/crontab`.

Задание 2

- 1) Запланируйте задание для текущего пользователя (узнать имя пользователя можно с помощью команды "whoami"), которое будет выполняться ежеминутно с 9:00 до 19:00 с понедельника по пятницу. Задание должно добавлять текущее время к содержимому файла /tmp/times. Подождите несколько минут, пока cron несколько раз исполнит ваше задание. Чтобы не терять времени, выполните пока следующие задания.
- 2) Команда "crontab -l" вернула следующее значение:

```
5 * * * * /usr/local/getstat.sh
```

Сколько раз в сутки выполнится скрипт getstat.sh?
- 3) Убедитесь, что в системе присутствует пакет с именем sysstat. Если пакет отсутствует, установите его командой "sudo yum install sysstat -y". Откройте файл /etc/cron.d/sysstat.
- 4) С какой частотой выполняется файл sa1? _____
- 5) С какой частотой выполняется файл sa2? _____
- 6) Измените файл так, чтобы файл sa1 выполнялся один раз в полчаса.
- 7) Вернемся к началу упражнения. Убедитесь, что в директории /tmp появился файл times, содержащий несколько значений времени. Удалите задание, добавляющее время в этот файл.

1) crontab -e

Вид задания, которое будет выполнять нужные действия:

```
* 9-19 * * 1-5 /usr/bin/date +%H:%M >> /tmp/times
```

2) Сколько раз в сутки выполнится скрипт getstat.sh? 24

4) С какой частотой выполняется файл sa1? Каждые 10 минут

5) С какой частотой выполняется файл sa2? Один раз в сутки в 23:53

```
6) */30 * * * * root /usr/lib64/sa/sa1 1 1
```