

6. Формирование отчетов по расписанию



Немного истории

Для периодического выполнения пользовательских или системных программ необходимы специальные сервисы - планировщики. Еще в 70-х годах прошлого века появился cron - сервис для регулярного выполнения задач суперпользователя. Типичный современный cron — несложная программа, но алгоритм работы оригинального варианта был ещё проще: сервис просыпался раз в минуту, читал табличку с задачами из единственного файла (/etc/lib/crontab) и выполнял для суперпользователя те задачи, которые следовало выполнить в текущую минуту.

`cron` – это утилита, позволяющая выполнять скрипты на сервере в назначенное время с заранее определенной периодичностью.

К примеру, у вас есть скрипт, который собирает какие-либо статистические данные каждый день в 6 часов вечера. Такие скрипты называют «заданиями», а их логика описывается в специальных файлах под названием `crontab`.

`crontab` – это таблица с расписанием запуска скриптов и программ, оформленная в специальном формате, который умеет считывать компьютер.

Для каждого пользователя системы создается отдельный `crontab`-файл со своим расписанием.

Конфигурационные файлы

Следующие файлы ограничивают доступ к планировщику (по умолчанию они не существуют):

- `/etc/cron.allow` - если существует, пользователи указанные в этом файле имеют возможность запуска заданий планировщика;
- `/etc/cron.deny` - если существует, пользователи указанные в этом файле НЕ имеют возможность запуска заданий планировщика.

Если `cron.allow` существует, только юзерам, перечисленным в нем, разрешено использовать `cron`, при этом файл `cron.deny` игнорируется. Если `cron.allow` не

существует, юзерам, указанным в `cron.deny`, не разрешается использовать планировщик `cron`.

Формат записей в обоих файлах - одно имя пользователя в каждой строке. Пробелы не разрешены.

Файлы контроля доступа считываются каждый раз, когда пользователь пытается добавить или удалить задачу `cron`.

Параметры команды `Crontab`

<code>crontab -e</code>	Редактирование или создание файла расписания для текущего пользователя
<code>crontab -l</code>	Вывод содержимого расписания текущего пользователя
<code>crontab -r</code>	Удаление файла расписания текущего пользователя
<code>crontab -u user</code>	Работа с расписаниями конкретных пользователей. Доступно только суперпользователю<

Каждая запись в `crontab` состоит из шести полей, указываемых в следующем порядке:

```
minute(s) hour(s) day(s) month(s) weekday(s) command(s)
```

Поля разделяются пробелами или знаками табуляции. Первые пять представляют собой числовые значения, а шестое - это команда, которая будет выполняться. В следующей таблице кратко описывается каждое из полей:

Поле	Диапазон значений	Описание
minute	0-59	Минута запуска команды
hour	0-23	Час запуска
day	1-31	Число (день) запуска
month	1-12	Месяц запуска
weekday	0 - 6	День недели запуска (воскресенье = 0, понедельник = 1, вторник = 2 и т.д.)
command		Последовательность команд для выполнения. Это могут быть команды, исполняемые файлы (например, скрипты) или комбинации файлов.

*Примечание: значение первых пяти полей может быть либо * (звездочкой), что означает все допустимые значения, либо списком элементов, разделенных запятыми. Элемент является либо числовым, либо диапазоном чисел, обозначенным двумя значениями, разделенными тире (например, 10-12).*

Примеры использования

Запуск скрипта каждый понедельник в 9:00 и 18:00 часов:

```
0 9,18 * * 1 /home/ivan/script.sh
```

Запуск скрипта каждый день в 0:30 и 12:30:

```
30 */12 * * * script.sh
```

Запуск команды каждый месяц 1 числа в 12:00:

```
0 12 1 * * command
```

Запуск скрипта в полночь с понедельника по пятницу:

```
0 0 * * 1-5 /home/ivan/script.sh
```

Есть сервисы чтобы посмотреть как расшифровывается cron-расписание. Например, <https://crontab.cronbox.ru/>, <https://crontab.guru/>, <http://crontab.cronhub.io/> и т. д.

Для чего обычно используют Cron?

Представьте себе, что вам раз в неделю надо сдавать отчет по многомиллионной выборке и этот запрос отработывает где-то полчаса. Это вполне реальная ситуация, есть некоторые отчеты, которые формируются даже по 2-3 часа. Для этого ставится специальная машина. Но вы - человек опытный, вы написали запрос и не хотите сейчас сидеть и ждать. Вы в себе уверены и формируете материализованное представление без заполнения, после вы ставите планировщик задач (любой, классический Крон, Windows, любая операционная система), к примеру в понедельник в 5 утра у вас будет запущено заполнение этого материализованного запроса

данными. Вы приходите на работу, запускаете - 5 секунд и у вас готовы данные. Т.е. надпись with NO DATA именно об этом и говорит - сейчас не ждем эти данные, сейчас только формируем запрос.

А если мы не настолько уверены в себе и хотим сделать тестовый отчет - загрузить только 10 строк из него? Тогда надо изначально сделать дополнительную выборку только по 10 записям. Обычно по 1000 записей.

Как понять, на какую дату и время сохранены данные view?

У вас есть таблица "название представления | дата обновления". Эту таблицу делаем не представлением (оно не записывает данные), а триггером для автоматической отработки. Триггер запускает триггерную функцию, которая и запишет в таблицу данные.

Когда планировщик задач, к примеру, в понедельник в 5 утра обновляет ваше представление, в таблицу очередного аудита. После того, когда данные будут обновлены, транзакция закончится успешно, в табличку будет занесено: представление такое обновилось тогда-то.

Базовые принципы работы с Cron и crontab (через панель управления)

Многие хостинг-провайдеры предлагают отдельное меню в панели управления для настройки расписания запланированного выполнения скриптов.

Разберем подобное меню на примере панели управления Timeweb. Чтобы создать новую задачу, необходимо открыть раздел Crontab в боковой панели веб-интерфейса, кликнуть по кнопке «Добавить новую задачу» и указать параметры повторяющейся команды. Поговорим подробнее о параметрах.

- Сначала надо придумать для команды название (подойдет любой текст без спецсимволов).
- Затем указываем исполнителя (нужно выбрать, будет ли планировщик работать с исполняемым файлом, PHP-скриптом или HTTP-запросом).
- В графе «Путь до файла» вводим абсолютный путь до скрипта, запуск которого хотим запланировать. К примеру: /home/u/myusername/mytestscript.php. При желании можно воспользоваться встроенным файловым менеджером и выбрать заранее предзагруженный на сервер скрипт.
- После этого указываем периодичность выполнения выбранного скрипта или исполняемого файла (в списке доступны предустановки в духе «Каждую минуту» или «Раз в день», но можно выбрать и пункт «Продвинутые настройки»).

- Кликаем по кнопке «Создать задачу».

На этом все. Скрипт запланирован и будет регулярно повторяться.

Базовые принципы работы с Cron и crontab (через SSH-протокол)

Планировать задачи через панель управления удобно, но не всегда возможно. Не все хостинг-провайдеры предлагают такие функциональные веб-интерфейсы. В этом случае придется воспользоваться командной строкой, подключившись к серверу по протоколу Secure Shell.

Для работы с планировщиком в системе есть ряд команд, помогающих решать основные задачи:

- `crontab -e` – открывает конфигурационный файл (поговорим о нем чуть подробнее в разделе с первичной настройкой).
- `crontab -l` – показывает список задач из конфигурационного файла (все, что было запланировано).
- `crontab -r` – удаляет конфигурационный файл вместе со всеми запланированными задачами.
- `crontab -v` – показывает, когда в последний раз открывался конфигурационный файл.

Чтобы запланировать задачи, используя командную строку, необходимо выполнить базовую настройку Cron, проверить, не установлены ли ограничения, и заполнить расписание задач в соответствии с синтаксисом `crontab`.

Первичная настройка Cron

Как мы уже выяснили ранее, планировщик черпает параметры для выполнения своих задач из `crontab`-файлов (таблиц с расписанием). У каждого пользователя, включая `root`, должен быть свой `crontab`-файл. По умолчанию он не существует, поэтому придется создать его вручную.

Для этого существует команда **`crontab -e`**. Она автоматически генерирует таблицу в директории `/var/spool/cron`.

Вновь созданный файл будет пустым текстовым полем. Необходимо добавлять в него все параметры самостоятельно с нуля, опираясь на синтаксис `crontab` (более подробно поговорим о нем ниже). После ввода параметров нужно сохранить параметры редактора, нажав на клавишу F2, а затем покинуть конфигурационный файл, нажав на клавишу F10. При введении корректных параметров в терминале отобразится строка **`crontab: installing new crontab`**.

Опытные разработчики и системные администраторы не рекомендуют использовать для редактирования расписания текстовые редакторы в духе Nano, Emacs или Vi. Команды `crontab` позволяют не только внести изменения в таблицу запланированных задач, но и перезапустить фоновый процесс `crond`, отвечающий за работу утилиты после сохранения настроек.

Ограничения Cron

У самого Cron имеется всего одно ограничение – период исполнения задач. Если вы перегружаете сервер своего хостинга, он может поставить запрет на запуск Cron чаще, чем раз в несколько минут или час. Кроме этого Cron ограничен только ресурсами сервера, такими как: оперативная память, объемы данных, время выполнения команд и т.д.

Если сервер перезагрузится, то Cron сохранит все ваши задачи, перезагрузившись вместе с ним.

Отчет

У Cron существует настройка, чтобы настроить уведомления о его работе на электронную почту. Это особенно полезно, если при работе случится какая-либо ошибка. Чтобы включить такие уведомления на почту, необходимо в crontab вписать строчку:

```
MAILTO= info@mydomain.com
```

После MAILTO= указывается необходимая почта. Указать их можно сразу несколько, перечислив через запятую. Теперь, если вдруг случится ошибка, вы будете проинформированы об этом. При чем, к вам на почту будут приходить и результаты работы скриптов. Однако, если данная функция вам мешает, ее можно и отключить, введя в конце задачи команду `> /dev/null 2>&1`.

О работе с Cron стоит знать еще пару важных вещей.

Во-первых, при указании «исполнителя» в панели управления Timeweb важно делать корректный выбор, чтобы он соответствовал запускаемой задаче.

Во-вторых, информацию о работе Cron можно собирать в отдельный файл с помощью команд в духе:

```
30 18 * * * rm /home/myusername/tmp/* > /home/myusername/cronlogs/clean_tmp_dir.log
```

На этом все. Следуйте инструкциям, не путайте порядок параметров и внимательно изучайте журнал ошибок, если что-то пойдет не так. После недолгой практики вы поймете, что работать с Cron не так уж и сложно!