

# Планировщик Cron

Cron — это встроенная в Linux программа, которая позволяет запускать какие-то действия по расписанию. Такие действия могут быть одиночными командами, а могут быть довольно сложными скриптами.

Cron берёт задачи из файла специального формата. Как и в случае с `sudo` или добавлением пользователей, есть программа, которая позволяет его редактировать, — `crontab`. Это удобно, если мы вручную добавляем какие-то действия. Если же это какая-то система конфигурации кластера, то удобнее пользоваться возможностью просто создать файл и положить его крону в папку `/etc/cron.d/`.

Директория, название которой заканчивается на `.d`, в некотором смысле особенная. В ней можно сохранять небольшие файлы конфигурации, которые программа потом сама соберёт в один большой.

Например, в случае с Cron мы могли бы записать 50 сценариев в один файл, а могли бы создать 50 отдельных файлов, по сценарию в каждом. Для чего это могло быть удобно? Например, если разные файлы управляются разными людьми — такое поведение прозрачнее. Или можно было бы поделить сценарии на логические группы и скомпоновать их по файлам с соответствующими названиями — все сценарии для обновления, все сценарии для резервного копирования и так далее.

Давайте рассмотрим самый простой случай описания задачи:

```
@hourly date > /tmp/date
```

Это самый простой способ реализации скрипта. Раз в час мы будем запускать программу `date`, которая выводит день недели, дату, время, часовой пояс и год, и перенаправлять её вывод в файл. Задание запустится от того пользователя, который создаёт задачу.

Таких коротких обозначений несколько:

`@hourly` — раз в час.

`@daily` — раз в день в полночь.

`@weekly` — раз в неделю в полночь воскресенья.

`@monthly` — раз в месяц в полночь на первый день месяца.

`@yearly` — раз в год в полночь на 1 января.

Но такие обозначения — нестандартные. Они могут не поддерживаться в каких-то дистрибутивах Linux или в других UNIX-подобных системах.

Давайте рассмотрим более стандартный способ задать расписание.

```
03 05 * * * date > /tmp/date
```

Такая задача будет запускаться в 05:03 ежедневно.

У Cron есть специальный способ задания периодичности выполнения задач. У нас есть возможность выставить минуты, часы, день месяца и день недели. Если в поле стоит звёздочка, это значит «каждый», число — соответственно порядковый номер числа, минуты, часа и так далее.

\* / 10 \* \* \* \* так мы можем запускать задачи каждые десять минут.  
 \* \* \* \* / 7 а так — каждое воскресенье.

```
.----- минута (0 - 59)
| .----- час (0 - 23)
| | .----- день месяца (1 - 31)
| | | .---- месяц (1 - 12) OR jan, feb, mar, apr ...
| | | | .-- день недели (0 - 6) (Воскресенье=0 или 7)
| | | | |
* * * * *
```

С этим знанием мы в принципе можем переписать «короткие» обозначения, приведённые выше, в обычном формате. Они будут выглядеть так:

@yearly	раз в год в полночь на 1 января	0 0 1 1 *
@monthly	раз в месяц в полночь на первый день месяца	0 0 1 * *
@weekly	раз в неделю в полночь воскресенья	0 0 * * 0
@daily	раз в день в полночь	0 0 * * *
@hourly	каждый час	0 * * * *

Ещё несколько интересных примеров.

```
59 23 31 dec *
```

Так мы сможем запускать задачу каждый новый год. То есть в последнюю минуту последнего часа 31 декабря (день недели нам не важен).

```
0 22 * * 1-5
```

Так мы сможем запускать задачу в 22:00 каждый будний день.

Cron позволит нам запускать задачи для того, чтобы мы могли не перемещать все файлы вбиванием команд или даже запуском скрипта, а чтобы это происходило автоматически, без нашего участия.

Cron работает, только пока работает операционная система. В случае серверов эту подробность уже не нужно держать в голове — они работают всегда, их выключение — авария. А вот наша виртуальная машина, скорее всего, включается периодически, только когда вы учитесь или делаете практические задания (всё-таки она потребляет ресурсы вашего компьютера или ноутбука). Поэтому задача, которая запланирована на время, когда виртуальная машина будет выключена, не выполнится.

### **Создать простой Cron-сценарий**

Чтобы посмотреть все существующие сценарии для текущего пользователя, используйте команду:

```
crontab -l
```

Для редактирования конфигурации используется команда:

```
crontab -e
```

Когда вы пользуетесь командой `crontab` первый раз, обычно она спрашивает, какой текстовый редактор следует использовать. Мы, конечно же, выбираем Vim.

```
* * * * * date >> /tmp/date
```

Теперь подождём пару минут и посмотрим:

```
cat /tmp/date
```

Работает.