cron

cron – программа-демон, предназначенная для выполнения заданий в определенное время, или через определенные промежутки времени. Для редактирования заданий используется утилита crontab

Нельзя изменять файл расписания, просто открыв его в текстовом редакторе. Для его редактирования необходимо использовать команду **crontab -e.**

Простой пример файла расписания. Создание снимка экрана каждый час:

```
SHELL=/bin/bash
MAILTO=username
0 * * * * scrot
```

Основные настройки

Выбор среды для исполняемых задач:

```
SHELL=/bin/bash
```

Имя пользователя, которому будет послано сообщение о выполнении задания:

```
MAILTO=username
```

Вместо имени, также можно использовать электронный адрес:

```
MAILTO=example@example.org
```

Необязательные настройки

Задаем каталог пользователя:

```
HOME=
```

Задаем каталоги исполняемых файлов:

```
PATH=/sbin:/bin:/usr/sbin:/usr/bin
```

Прописываем дисплей, если запускаемый по крону скрипт должен использовать графику:

```
DISPLAY=:0.0
```

Выбор времени выполнения задания

Время запуска представляется в таком виде:

```
*     *     *     *     *     выполняемая команда
-     -     -     -     -     -
|     |     |     |     |     |
|     |     |     |     |     День недели (0 - 7) (Воскресенье =0 или =7)
|     |     |     |     -------  Месяц (1 - 12)
|     |     |     ---------  День (1 - 31)
|     |     -----------  Час (0 - 23)
|     ------------  Минута (0 - 59)
```

Значение	Диапазон	Дополнительно
минуты	0-59	
часы	0-23	
дни месяца	1-31	
месяцы	1-12	можно задавать и в 3-х буквенном варианте
дни недели	0-6	можно задавать и в 3-х буквенном варианте (0=воскресенье)

Символ '*' подразумевает - любое значение.

Минимальное время 1-а минута. Это связано с тем что cron каждую минуту просматривает список заданий, и ищет которые нужно выполнить.

Дни недели и месяца в трех буквенном варианте:
sun mon tue wed thu fri sat
jan feb mar apr may jun jul aug sep oct nov dec

Дополнительные переменные cron

Переменная	Описание	Эквивалент
@reboot	Запуск при загрузке	
@yearly	Раз в год	0 0 1 1 *
@annually	Тоже что и @yearly	
@monthly	Раз в месяц	0 0 1 * *

@weekly	Раз в неделю	0 0 * * 0
@daily	Раз в день	0 0 * * *
@midnight	В полночь (00:00)	
@hourly	Каждый час	0 * * * *

Управление при помощи crontab

Добавление файла расписания:

```
crontab имя_файла_расписания
```

Вывести содержимое текущего файла расписания:
```
crontab -l
```

Удаление текущего файла расписания:
```
crontab -r
```

Редактирование текущего файла расписания (при первом запуске будет выведен список поддерживаемых текстовых редакторов):
```
crontab -e
```

Этот ключ позволяет выполнять вышеописанные действия для конкретного пользователя:
```
crontab -u username
```
Учтите, что `-u` можно воспользоваться только запуская crontab от лица суперпользователя.

Примеры

Каждую минуту:
* * * * *
Каждый час, с использованием переменных cron:
@daily ~/script.sh

Каждый день в 03:15 запускать скрипт:
15 3 * * * ~/script.sh

Каждый понедельник:
0 0 * * 1

Каждый четный час:
* */2 * * *

Тоже что и выше, но расписанное полностью:
* 0,2,4,6,8,10,12,14,16,18,20,22 * * *

Каждые 10 минут:
*/10 * * * *

Новый год :
59 23 31 dec *

Каждый будний день в 22:00:
0 22 * * 1-5

Запуск программы с графической оболочкой (GUI); каждую минуту будет отображаться сообщение с текстом «It work!»:
0-59 * * * * DISPLAY=:0 gdialog --msgbox "It work\!" 25 20 > /dev/null
* где DISPLAY=:0 - номер монитора

 # выполнять задание раз в час в 0 минут
0 */1 * * * /home/u12345/script.pl

 # выполнять задание каждые три часа в 0 минут
0 */3 * * * /home/u12345/script.pl

 # выполнять задание по понедельникам в 1 час 15 минут ночи
15 1 * * 1 /home/u12345/script.pl

 # выполнять задание 5 апреля в 0 часов 1 минуту каждый год
1 0 5 4 * /home/u12345/script.pl

 # выполнять задание ежемесячно 1 числа в 6 часов 10 минут
10 6 1 * * /home/u12345/script.pl

 # выполнять каждый день в 0 часов 5 минут, результат складывать в log/daily
5 0 * * * $HOME/bin/daily.job >> $HOME/log/daily 2>&1
 # выполнять 1 числа каждого месяца в 14 часов 15 минут
15 14 1 * * $HOME/bin/monthly
 # каждый рабочий день в 22:00
0 22 * * 1-5 echo "Пора домой" | mail -s "Уже 22:00" john

23 */2 * * * echo "Выполняется в 0:23, 2:23, 4:23 и т. д."
5 4 * * sun echo "Выполняется в 4:05 в воскресенье"
0 0 1 1 * echo "С новым годом!"
15 10,13 * * 1,4 echo "Эта надпись выводится в понедельник и четверг в 10:15 и 13:15"
0-59 * * * * echo "Выполняется ежеминутно"
0-59/2 * * * * echo "Выполняется по четным минутам"
1-59/2 * * * * echo "Выполняется по нечетным минутам"

 # каждые 5 минут

*/5 * * * * echo "Прошло пять минут"

 # каждое первое воскресенье каждого месяца. -eq 7 это код дня недели, т.е. 1 -> понедельник , 2 -> вторник и т.д.

0 1 1-7 * * ["$(date '+\%u')" -eq 7] && echo "Эта надпись выводится каждое первое воскресенье каждого месяца в 1:00"

КАК ЗАПУСКАТЬ PHP-СКРИПТЫ ПО РАСПИСАНИЮ

Вы можете выполнять запуск скриптов на языке PHP в указанное время с желаемой периодичностью. Для этого требуется использовать PHP-CLI интерпретатор, пример использования которого описан здесь.

Поскольку не все PHP-программы могут работать через CLI SAPI без предварительной модификации, можно запускать их через wget. Например:
/usr/local/bin/wget -O /dev/null -q http://mysite.tld/cron.php?action=123

Если в скрипте используются функции require, include, причём в них указаны относительные пути, то в начале выполняемого скрипта используйте вызов функции chdir(), которая задаст текущую рабочую директорию.

КАК ПОЛУЧАТЬ СООБЩЕНИЯ ОБ ОШИБКАХ ОТ ПРОГРАММ, ЗАПУСКАЕМЫХ CRON

Если при выполнении программы, которая запускается из cron, возникли ошибки, наверняка вы захотите получать сообщения об этих ошибках, чтобы полностью контролировать работу периодически запускаемых заданий. Для этого в начале cron-сценария поместите такую строку:

MAILTO=адрес@домен.ru

Конечно, адрес@домен.ru нужно заменить на реально существующий адрес электронной почты куда надо будет доставлять уведомления. Если нужно получать сообщения об ошибках на несколько адресов, укажите все эти адреса через запятую.

Обратите внимание на то, что cron будет присылать по почте то, что выводят запускаемые скрипты. Например, если вы напишете скрипт, который будет печатать строчку «Hello, world» и поставите его на выполнение через cron, вы будете получать по почте письмо со строкой «Hello, world» каждый раз, когда cron будет запускать такой скрипт.

Чтобы избежать этого, например когда текст, выводимый скриптом, вам не нужен, надо добавить в конец строки-сценария для cron символы

> /dev/null 2>&1

Полностью строка для cron будет выглядеть так:
0 1 * * * /usr/local/bin/php -q $HOME/script.php > /dev/null 2>&1

Рекомендуем проверять корректность синтаксиса скриптов, которые вы устанавливаете на выполнение через cron. Скрипты могут содержать ошибку, могут неодинаково работать при запуске через веб-сервер и через cron, и так далее. Для того, чтобы убедиться что скрипт будет правильно работать через cron, предварительно проверьте его такой командой в unix shell:

/usr/local/bin/php -l script.php

Если ошибок в скрипте нет, вы увидите сообщение «No syntax errors detected in script.php».

КАК ЗАПУСКАТЬ MYSQL ПО РАСПИСАНИЮ

для этого достаточно знать как запускать команды mysql из командной строки:

mysql -u USER -p PASSWORD -h MYSQLSERVERNAME -e 'select * from foo...' database-name

-u : указывает на пользователем под которым запускать запрос
-p : парол-u : Specify mysql database user name
-p : Prompt for password
-e : Execute sql query
database : Specify database name на этого пользователя
-e : запрос который необходимо выполнить
database : Имя базы данных

ПРИМЕРЫ

просмотреть все базы данных:

$ mysql -u vivek -p -e 'show databases;'

To list count all rows, enter:

$ mysql -u vivek -p -e 'SELECT COUNT(*) FROM quotes' cbzquotes