

## Глава 11. Отложенное и регулярное выполнение заданий.

### 11.1. Отложенное выполнение заданий с помощью `at`.



#### Отложенное выполнение заданий с помощью `at`

- Команда `at` выполняет команды в указанное время
- Время может быть абсолютное или относительное
- Команды запускаются с переменными пользователя, который запустил `at`
- Для контроля доступа к `at` используются файлы `/etc/at.allow` и `/etc/at.deny`
- Команда `batch` запускает команды, когда загрузка системы снизится

Часто возникает необходимость выполнить какую-либо команду в заданный момент будущего или же в тот момент, когда загрузка системы минимальна.

Команда `at` позволяет указать момент времени, в который должна быть исполнена команда. При этом заданная команда выполняется только один раз в заданный момент времени.

#### Пример:

```
$ date
Втр Янв 13 22:12:38 YEKT 2004

$ at 22:20
warning: commands will be executed using /bin/sh
at> ps -ef | mail -s 'At command' user1
at> <EOT>
job 2 at 2004-01-13 22:20
```

*Примечание: В 22:12 была запущена команда `at 22:20`. Команда `at`, вызванная таким образом предоставляет пользователю возможность интерактивно ввести после приглашения `at>` команды, которые необходимо выполнить в будущем. В этом примере была введена команда, посылающая в заданное время пользователю `user1` список всех процессов в системе в подробном формате.*

Помимо указания астрономического времени выполнения команды можно задавать период времени, через который должна быть выполнена заданная команда.

## Глава 11. Отложенное и регулярное выполнение заданий.

### **Пример:**

```
$ at now +10 minutes
```

*Примечание:* В этом случае команда `at` выполнит команду, заданную в ее интерактивном сеансе, через десять минут относительно текущего момента времени.

Задавать команды для исполнения можно также направляя командую строку в стандартный поток ввода для команды `at` или же указывая после опции `-f` имя файла с командами.

Опция `-m` команды `at` посылает вызвавшему ее пользователю сообщение по электронной почте о том, что задание выполнено.

Для просмотра списка заданий можно использовать команду `atq` или `at -l`:

### **Пример:**

```
$ atq
3      2004-01-14 11:13 a user1
```

*Примечание:* При исполнении этой команды обычный пользователь получает список только его заданий, а суперпользователь - список заданий всех пользователей.

Для удаления задания из очереди следует вызвать команду `atrm` или `at -d`. Номер задания должен быть указан в качестве аргумента:

### **Пример:**

```
$ echo 'wall Hello!' | at -m +1 minutes
warning: commands will be executed using /bin/sh
job 5 at 2004-01-14 12:04
```

```
$ atq
5      2004-01-14 12:04 a user1
```

```
$ atrm 5
```

```
$ atq
```

*Примечание:* В этом примере была задана команда `wall`, которая должна была вывести сообщение `Hello!` на терминалы всех пользователей, находящихся в сеансе, через минуту после ее вызова. Однако, эта команда снята с исполнения командой `atrm`.

Команда `batch` отличается от `at` тем, что она выполняет задание не через четко определенный промежуток времени, а в тот момент, когда средняя загрузка системы уменьшается до 0.8. Значение средней загрузки системы берется из файла `/proc/loadavg`. В остальном команда `batch` аналогична команде `at`.

В системе могут иметься файлы `/etc/at.allow` и `/etc/at.deny`. Если существует файл `/etc/at.allow`, то только пользователи, перечисленные в нем, могут вызывать команду `at`.

Если этого файла нет, то проверяется наличие файла `/etc/at.deny`, в котором указываются пользователи, которым запрещено вызывать `at`.

## 11.2. Автоматизация выполнения регулярных рутинных задач.



### Автоматизация выполнения регулярных рутинных задач

- Для выполнения заданий по расписанию используется демон `cron`
- В `systemd` можно создать юнит типа `timer` для запуска периодических заданий
- Служба `anacron` может запускать пропущенные периодические задания, в том случае, когда машина была выключена
- В отличие от `cron` в `anacron` указывается не точное время заданий, а желаемая периодичность

В отличие от `at` команда система `cron` позволяет выполнить задание не единожды, а выполнять его на регулярной основе с заданной периодичностью.

В различных версиях GNU/Linux используются различные пакеты `cron`.

**Пример:** `viixie-cron` или `cronie`. Их работа несколько отличается в деталях.

Основой регулярного выполнения заданий в любом из вариантов `cron` являются таблицы, в которых кодируется периодичность выполнения заданий.

Задания выполняет демон `cron`.

Обычно имеется общесистемная таблица заданий `cron`, находящаяся в файле `/etc/crontab`, и индивидуальные таблицы периодических заданий пользователей, находящиеся в файлах, обычно расположенных в каталоге `/var/spool/cron`. Имена таких файлов совпадают с именами пользователей, для которых созданы таблицы периодических заданий.

*Примечание:* В некоторых системах общесистемный файл `/etc/crontab` не используется, а вместо этого в качестве общесистемной таблицы заданий `cron` работает таблица заданий суперпользователя.

Файл общесистемной таблицы заданий `/etc/crontab` имеет семь полей, при этом строки, начинающиеся с решетки считаются комментариями.

Первые пять полей используются для указания периодичности выполнения задания. Шестое поле указывает от имени какого пользователя должно быть исполнено задание. В седьмом поле указывают саму команду, которая должна быть выполнена.

Формат задания времени следующий:

1. Минуты часа (0-59).

## Глава 11. Отложенное и регулярное выполнение заданий.

2. Час суток (0-23).
3. Календарное число (день месяца 1-31).
4. Месяц (1-12).
5. День недели начиная с воскресенья (0-6).

*Примечание: Если в команде задаются аргументы и опции командной строки, то команда не экранируется кавычками. Поэтому иногда создается впечатление, что в строках таблицы crontab больше, чем семь полей.*

### **Пример:**

```
# /etc/crontab: system-wide crontab
SHELL=/bin/sh
PATH=/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
# m h dom mon dow user  command
25 6 * * * root run-parts /etc/cron.daily
47 6 * * 7 root run-parts /etc/cron.weekly
52 6 1 * * root run-parts /etc/cron.monthly
08 * * * * root /usr/X11R6/bin/log_clean
```

*Примечание: В таблице crontab, показанной выше, установлены две переменные окружения - SHELL и PATH, и определены четыре задания. Первая из них указывает с помощью какой оболочки должны быть запущены задания cron, а вторая переопределяет пути доступа к исполняемым командам. При этом в первом, втором и третьем задании командой является run-parts, получающая в качестве аргумента имя каталога. Задание, определенное в последней строке crontab из примера выше, будет запускаться в восемь минут каждого часа, а предпоследнее задание - в 6 часов 52 минуты первого числа каждого месяца.*

Общесистемную таблицу заданий имеет право изменять лишь суперпользователь.

Вместо редактирования файла /etc/crontab принято создавать таблицы в виде отдельных файлов в каталоге /etc/cron.d.

Использование /etc/cron.d обычно удобнее, т. к. позволяет определить индивидуальные переменные для заданий в таких файлах.

Обычные пользователи могут иметь право устанавливать свои собственные таблицы заданий.

Таблицы заданий cron для обычных пользователей не имеют поля, в котором указывается имя пользователя, поэтому в них всего шесть полей.

### Формат указания времени в crontab

m h dom mon dow

Где

- m — минуты (0-59)
- h — часы (0-23)
- dom — дни месяца (1-31)
- mon — месяцы (1-12)
- dow — дни недели (0-7), 0 и 7 — воскресенье

Специальные символы

- \* - любое значение
- / - периодичность, например \*/5
- - - интервал, например 1-3
- , - список, например 1,6,12,18

Для манипулирования с заданиями `crond` существует специальная команда `crontab`. Если ей в качестве аргумента указан файл, содержащий таблицу заданий `crond`, то он будет считан, старая таблица `crn` будет стерта, а новая из файла будет сохранена.

#### **Пример:**

```
$ echo '30 21 * * * wall Hello!' > crontab.eg
$ crontab crontab.eg
```

*Примечание: В этом примере был создан файл `crontab.eg`, содержащий единственную строку с заданием, которое должно выполняться регулярно в 21:30. Эта таблица была считана и установлена командой `crontab`.*

Если вызвать команду `crontab` с опцией `-l`, то она выведет список регулярно выполняемых заданий пользователя:

#### **Пример:**

```
$ crontab -l

# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (crontab.eg installed on Wed Jan 14 21:51:49 2004)
# (Cron version -- $Id: crontab.c,v 1.15 2002/11/18 13:44:30 ldv Exp $)
30 21 * * * wall Hello!
```

Суперпользователь может обращаться к таблицам пользователей, указывая их после опции `-u` команды `crontab`:

#### **Пример:**

```
# crontab -l -u user1

# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (crontab.eg installed on Wed Jan 14 21:51:49 2004)
```

## Глава 11. Отложенное и регулярное выполнение заданий.

```
# (Cron version -- $Id: crontab.c,v 1.15 2002/11/18 13:44:30 ldv Exp $)
30 21 * * * wall Hello!
```

### Порядок работы с индивидуальными таблицами

- Текущая таблица пользователя сохраняется во временном файле для редактирования:  
`crontab -l > crontab.eg`
- Временный файл редактируется в текстовом редакторе: `vi crontab.eg`
- Отредактированная таблица заданий запоминается: `crontab crontab.eg`

Для редактирования текущей таблицы заданий в текстовом редакторе можно использовать команду `crontab -e`. При вызове этой команды будет запущен текстовый редактор по умолчанию, в нем будет открыта текущая таблица заданий `crontab`, которую можно отредактировать. При выходе с сохранением будет установлена отредактированная версия таблицы заданий.

Крайне рекомендуется избегать подобной процедуры, поскольку ошибка при редактировании таблицы заданий может привести к ее порче и даже к потере.

Предлагается другой вариант редактирования:

1. Текущая таблица заданий пользователя сохраняется во временном файле для редактирования: `crontab -l > crontab.eg`.

2. Временный файл редактируется в текстовом редакторе: `vi crontab.eg`.

3. Отредактированная таблица заданий запоминается: `crontab crontab.eg`.

Если необходимо полностью удалить таблицу заданий, то необходимо выполнить команду `crontab -r`.

#### Пример:

```
$ crontab -l
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (crontab.eg installed on Wed Jan 14 21:51:49 2004)
30 21 * * * wall Hello!
```

```
$ crontab -r
$ crontab -l
crontab: no crontab for user1
```



### Контроль доступа к cron

- В файле `/etc/cron.allow` кому можно использовать cron. Остальным нельзя
- Файл `/etc/cron.deny` используется если нет `/etc/cron.allow` и в нем перечисляется кому нельзя использовать cron
- В некоторых случаях используется специальная группа для контроля доступа

Можно ограничивать доступ пользователей к команде `crontab` с помощью файлов `/etc/cron.allow` и `/etc/cron.deny`. При этом:

1. Если существует файл `/etc/cron.allow`, то только пользователи, указанные в нем, имеют право использовать команду `crontab`.
2. Если этого файла нет, но имеется файл `/etc/cron.deny`, то пользователи, указанные в нем, не могут вызывать команду `crontab`.





## Специальные каталоги

- В каталоги `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly` и `/etc/cron.monthly` можно поместить сценарии или программы, которые будут запускаться соответственно ежечасно, ежедневно, еженедельно или ежемесячно

Помимо создания таблиц с заданиями в `/etc/crontab`, вы можете создать скрипт с нужными вам командами и поместить этот сценарий в один из каталогов:

1. `/etc/cron.hourly` — задания выполняемые каждый час.
2. `/etc/cron.daily` — ежедневные задания.
3. `/etc/cron.weekly` — еженедельные задания.
4. `/etc/cron.monthly` — ежемесячные.

В дополнение к `cron` имеется еще одно средство выполнения регулярных заданий — `anacron`.

`Anacron` выполняет задания с заданной периодичностью, но при этом не предполагает что машина постоянно включена.

Конфигурация заданий находится в файле `/etc/anacrontab`.

### Пример:

```
$ cat /etc/anacrontab
# /etc/anacrontab: configuration file for anacron

# See anacron(8) and anacrontab(5) for details.

SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
# the maximal random delay added to the base delay of the jobs
RANDOM_DELAY=45
# the jobs will be started during the following hours only
START_HOURS_RANGE=3-22

#period in days   delay in minutes   job-identifier   command
1       5       cron.daily       nice run-parts /etc/cron.daily
7      25      cron.weekly      nice run-parts /etc/cron.weekly
```

## Глава 11. Отложенное и регулярное выполнение заданий.

```
@monthly 45 cron.monthly          nice run-parts /etc/cron.monthly
```

Задания в `/etc/anacrontab` описываются также в виде таблицы, где указывается:

1. Периодичность в днях.
2. Задержка в минутах.
3. Идентификатор заданиям.
4. Выполняемая команда.

Информация о выполненных заданиях записывается в каталоге `/var/spool/anacron`

### **Пример:**

```
$ sudo cat /var/spool/anacron/cron.daily
20210211
```