

Лабораторная работа №8

Управление событиями

Перфилов Александр Константинович | группа НПИбд 03-24

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Планирование задач с помощью cron	7
3.2	Планирование заданий с помощью at	13
4	Ответы на контрольные вопросы	15
5	Выводы	17

Список иллюстраций

3.1	Запуск терминала и получение полномочий администратора, просмотр статуса демона <code>crond</code> и содержимого файла конфигурации <code>/etc/crontab</code>	8
3.2	Просмотр списка заданий в расписании, открытие файла расписания на редактирование.	8
3.3	Открытие текстового редактора <code>vi</code> и добавление строки в файл расписания.	9
3.4	Просмотр списка заданий в расписании, просмотр журнала системных событий.	10
3.5	Изменение записи в расписании <code>crontab</code>	10
3.6	Просмотр списка заданий в расписании.	11
3.7	Открытие каталога <code>/etc/cron.hourly</code> и создание в нём файла сценария с именем <code>eachhour</code>	11
3.8	Открытие файла <code>eachhour</code> для редактирования и прописывание в нём скрипта.	11
3.9	Делаем файл сценария <code>eachhour</code> исполняемым, открытие каталога <code>/etc/cron.d</code> и создание в нём файла с расписанием <code>eachhour</code> , открытие файла на редактирование.	11
3.10	Добавление содержимого в файл и сохранение изменений.	12
3.11	Просмотр журнала системных событий через 2 часа.	13
3.12	Запуск терминала и получение полномочий администратора, проверка загрузки и включения службы <code>atd</code> , создание задачи выполнения команды <code>logger message from at</code> в 17:00, закрытие оболочки. Проверка планировки задачи и выполнения её в указанное время.	14

Список таблиц

1 Цель работы

Целью данной работы является получение навыков работы с планировщиками событий cron и at.

2 Задание

1. Выполните задания по планированию задач с помощью `cron` (см. раздел 8.4.1).
2. Выполните задания по планированию задач с помощью `atd` (см. раздел 8.4.2).

3 Выполнение лабораторной работы

3.1 Планирование задач с помощью cron

Запустим терминал и получим полномочия администратора: `su -`.
Посмотрим статус демона `crond`: `systemctl status crond -l` и содержимое файла конфигурации `/etc/crontab`: `cat /etc/crontab`

```

root@akpperfilov:~# systemctl status crond -l
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-10-24 15:22:52 MSK; 21min ago
 Invocation: 18d9bc5eb373415e862554a3032cb799
    Main PID: 1033 (crond)
      Tasks: 1 (limit: 48908)
     Memory: 1M (peak: 1.1M)
        CPU: 15ms
    CGroup: /system.slice/crond.service
           └─1033 /usr/sbin/crond -n

Okt 24 15:22:52 akpperfilov.localdomain systemd[1]: Started crond.service - Command Scheduler.
Okt 24 15:22:52 akpperfilov.localdomain crond[1033]: (CRON) STARTUP (1.7.0)
Okt 24 15:22:52 akpperfilov.localdomain crond[1033]: (CRON) INFO (Syslog will be used instead of syslogd)
Okt 24 15:22:52 akpperfilov.localdomain crond[1033]: (CRON) INFO (RANDOM_DELAY will be scaled with respect to allowed time)
Okt 24 15:22:52 akpperfilov.localdomain crond[1033]: (CRON) INFO (running with inotify support)
root@akpperfilov:~# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .----- 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
# | | | | |
# * * * * * user-name  command to be executed

root@akpperfilov:~#

```

Рисунок 3.1: Запуск терминала и получение полномочий администратора, просмотр статуса демона crond и содержимого файла конфигурации /etc/crontab.

Теперь посмотрим список заданий в расписании: `crontab -l`. Ничего не отобразилось, так как расписание ещё не задано. Далее откроем файл расписания на редактирование: `crontab -e`

```

root@akpperfilov:~# crontab -l
no crontab for root
root@akpperfilov:~# crontab -e

```

Рисунок 3.2: Просмотр списка заданий в расписании, открытие файла расписания на редактирование.

Предыдущая команда запустила интерфейс редактора (по умолчанию используется vi). Добавим следующую строку в файл расписания (запись сообщения в системный журнал), используя Ins для перехода в vi в режим ввода: `/1 * * * logger This message is written from root cron`. Закроем сеанс редактирования vi и сохраним изменения, используя команду vi: `Esc : wq`.

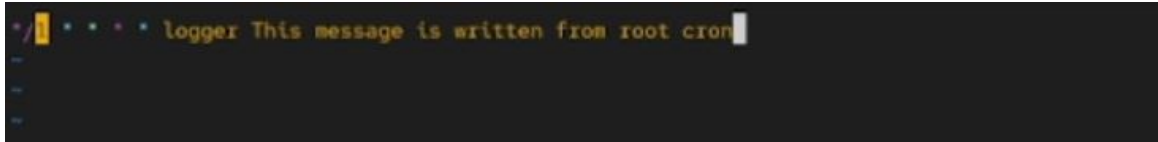


Рисунок 3.3: Открытие текстового редактора vi и добавление строки в файл расписания.

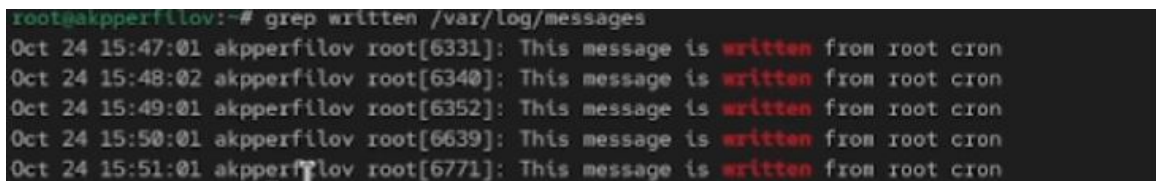
Давайте разберём синтаксис этой записи:

1. `/1` - *Запуск задания каждую минуту. Частота определяется первой звездочкой: `/1` означает «каждую минуту». Часто вместо `/1` можно увидеть просто `,` так как каждую минуту является условием по умолчанию.*
2. `****` - Упоминание отдельных звездочек говорит о том, что задание будет выполняться каждую минуту каждого часа, дня, месяца, и дня недели, поскольку второе, третье, четвертое и пятое поля - это день месяца, месяц и день недели, соответственно.
3. `logger` - Команда, которая будет выполняться каждый раз, когда cron запускает задачу. В данном случае это команда `logger`.
4. `This message is written from root cron` - Сообщение, отправляемое командой `logger` в системный журнал.

Таким образом, эта запись установит задачу, которая будет каждые 60 секунд записывать сообщение «This message is written from root cron» в системный журнал вашего компьютера.

Посмотрим список заданий в расписании: `crontab -l` (в расписании появилась запись о запланированном событии). Не выключая систему, через

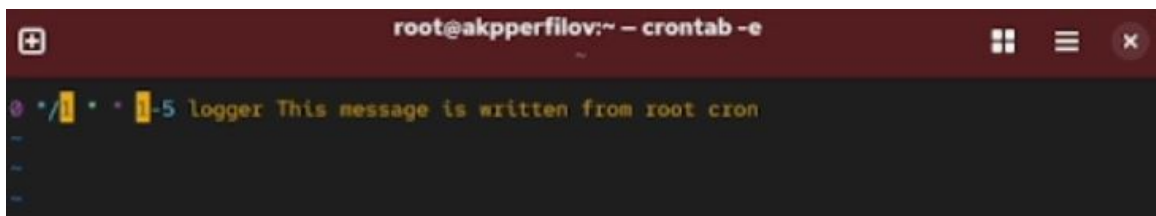
некоторое время (2–3 минуты) посмотрим журнал системных событий: `grep written /var/log/messages`



```
root@akpperfilov:~# grep written /var/log/messages
Oct 24 15:47:01 akpperfilov root[6331]: This message is written from root cron
Oct 24 15:48:02 akpperfilov root[6340]: This message is written from root cron
Oct 24 15:49:01 akpperfilov root[6352]: This message is written from root cron
Oct 24 15:50:01 akpperfilov root[6639]: This message is written from root cron
Oct 24 15:51:01 akpperfilov root[6771]: This message is written from root cron
```

Рисунок 3.4: Просмотр списка заданий в расписании, просмотр журнала системных событий.

Вернёмся в текстовый редактор `vi` и изменим запись в расписании `crontab` на следующую: `0 /1 * 1-5 logger This message is written from root cron`



```
root@akpperfilov:~ - crontab -e
* */1 * * 1-5 logger This message is written from root cron
```

Рисунок 3.5: Изменение записи в расписании `crontab`.

Синтаксис записи в `crontab`:

1. `0 /1 * 1-5` - это шаблон запуска задачи. Он состоит из пяти полей, разделенных пробелами:
2. • `0` - Минута. Задача будет запускаться в 0-ю минуту каждого часа.
3. • `/1` - Час. Звездочка (`*`) означает «каждый», `/1` означает «каждый час».
4. • `*` - День месяца. Звездочка означает «каждый день месяца».
5. • `*` - Месяц. Звездочка означает «каждый месяц».
6. `1-5` - День недели. Диапазон `1-5` означает «с понедельника по пятницу».
7. `logger This message is written from root cron` - Это команда, которая будет выполняться по расписанию. В данном случае команда `logger` записывает сообщение в системный журнал (`syslog`).

В итоге, данная запись crontab означает: «Каждую минуту, начиная с 0-ой минуты каждого часа, с понедельника по пятницу, запускать команду logger с сообщением This message is written from root cron».

Теперь посмотрим список заданий в расписании: crontab -l

```
root@akpperfilov:~# crontab -l
0 */1 * * 1-5 logger This message is written from root cron
```

Рисунок 3.6: Просмотр списка заданий в расписании.

Перейдём в каталог /etc/cron.hourly и создадим в нём файл сценария с именем eachhour

```
root@akpperfilov:~# cd /etc/cron.hourly
root@akpperfilov:/etc/cron.hourly# touch eachhour
root@akpperfilov:/etc/cron.hourly# nano eachhour
```

Рисунок 3.7: Открытие каталога /etc/cron.hourly и создание в нём файла сценария с именем eachhour.

Далее откроем файл eachhour для редактирования и пропишем в нём следующий скрипт (запись сообщения в системный журнал)

```
root@akpperfilov:/etc/cron.hourly – nano eachhour
GNU nano 8.1      eachhour      Изменён
#!/bin/sh
logger This message is written at $(date)
```

Рисунок 3.8: Открытие файла eachhour для редактирования и прописывание в нём скрипта.

Сделаем файл сценария eachhour исполняемым. Теперь перейдём в каталог /etc/cron.d и создадим в нём файл с расписанием eachhour

```
root@akpperfilov:/etc/cron.hourly# chmod +x eachhour
root@akpperfilov:/etc/cron.hourly# cd /etc/cron.d
root@akpperfilov:/etc/cron.d# touch eachhour
```

Рисунок 3.9: Делаем файл сценария eachhour исполняемым, открытие каталога /etc/cron.d и создание в нём файла с расписанием eachhour, открытие файла на редактирование.

Откроем этот файл для редактирования и поместим в него следующее содержимое

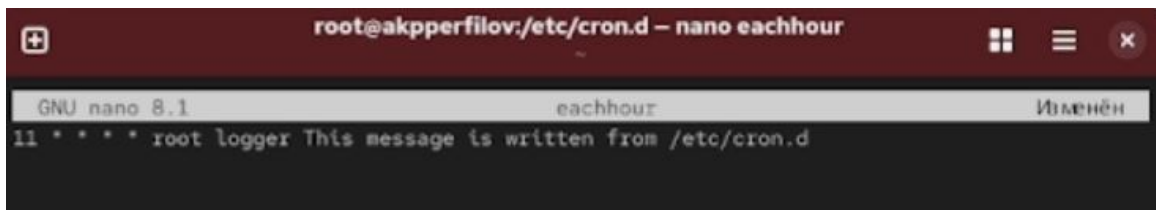


Рисунок 3.10: Добавление содержимого в файл и сохранение изменений.

Синтаксис записи в crontab:

1. 11 * * * * - это шаблон запуска задачи. Он состоит из пяти полей, разделенных пробелами:
2. 11 - Минута. Задача будет запускаться в 11-ю минуту каждого часа.
3. • - Час. Звездочка означает «каждый час».
4. • - День месяца. Звездочка означает «каждый день месяца».
5. • - Месяц. Звездочка означает «каждый месяц».
6. • - День недели. Звездочка означает «каждый день недели».
7. root - Это имя пользователя, от которого будет запускаться задание. В данном случае, задание будет запускаться от пользователя root.
8. logger This message is written from /etc/cron.d - Это команда, которая будет выполняться по расписанию. В данном случае команда logger записывает сообщение в системный журнал (syslog).

В итоге, данная запись crontab означает: «Каждую минуту, начиная с 11-ой минуты каждого часа, запускать от имени пользователя root команду logger с сообщением This message is written from /etc/cron.d».

Не выключая систему, через некоторое время (2 часа) посмотрим журнал системных событий. По журналу видим, что запуск сценария eachhour в соответствии с заданным расписанием был осуществлён

```

root@akpperfilov:/etc/cron.d# grep written /var/log/messages
Oct 24 15:47:01 akpperfilov root[6331]: This message is written from root cron
Oct 24 15:48:02 akpperfilov root[6340]: This message is written from root cron
Oct 24 15:49:01 akpperfilov root[6352]: This message is written from root cron
Oct 24 15:50:01 akpperfilov root[6639]: This message is written from root cron
Oct 24 15:51:01 akpperfilov root[6771]: This message is written from root cron
Oct 24 15:52:01 akpperfilov root[7007]: This message is written from root cron
Oct 24 16:00:01 akpperfilov root[7612]: This message is written from root cron
Oct 24 16:01:01 akpperfilov root[7645]: This message is written at Пн 24 окт 2025 16:01:01
MSK
Oct 24 16:11:01 akpperfilov root[7703]: This message is written from /etc/cron.d
Oct 24 17:00:01 akpperfilov root[7977]: This message is written from root cron
Oct 24 17:01:01 akpperfilov root[8000]: This message is written at Пн 24 окт 2025 17:01:01
MSK
Oct 24 17:11:02 akpperfilov root[8041]: This message is written from /etc/cron.d

```

Рисунок 3.11: Просмотр журнала системных событий через 2 часа.

3.2 Планирование заданий с помощью at

Запустим терминал и получим полномочия администратора: `su -`. Проверим, что служба `atd` загружена и включена: `systemctl status atd`. Зададим выполнение команды `logger message from at` в 17:00. Для этого введём `at 17:00`, а затем `logger message from at`. Используем `Ctrl + d`, чтобы закрыть оболочку. Убедимся, что задание действительно запланировано: `atq`. С помощью команды `grep from at /var/log/messages` посмотрим, появилось ли соответствующее сообщение в лог- файле в указанное вами время (сообщение появилось)

```
root@akpperfilov:/etc/cron.d# at 18:02
warning: commands will be executed using /bin/sh
at Sat Oct 25 18:02:00 2025
at> logger message from at
at> <EOT>
job 2 at Sat Oct 25 18:02:00 2025
root@akpperfilov:/etc/cron.d# atq
2          Sat Oct 25 18:02:00 2025 a root
root@akpperfilov:/etc/cron.d# grep 'from at' /var/log/messages
Oct 24 18:01:12 akpperfilov root[8678]: message from at
root@akpperfilov:/etc/cron.d# grep 'from at' /var/log/messages
Oct 24 18:01:12 akpperfilov root[8678]: message from at
```

Рисунок 3.12: Запуск терминала и получение полномочий администратора, проверка загрузки и включения службы atd, создание задачи выполнения команды logger message from at в 17:00, закрытие оболочки. Проверка планировки задачи и выполнения её в указанное время.

4 Ответы на контрольные вопросы

1. Как настроить задание cron, чтобы оно выполнялось раз в 2 недели? 00 00 1,15 * * logger task
2. Как настроить задание cron, чтобы оно выполнялось 1-го и 15-го числа каждого месяца в 2 часа ночи? 00 02 1,15 * * logger task
3. Как настроить задание cron, чтобы оно выполнялось каждые 2 минуты каждый день? /2 * * * logger task
4. Как настроить задание cron, чтобы оно выполнялось 19 сентября ежегодно? * * 19 9 logger task
5. Как настроить задание cron, чтобы оно выполнялось каждый четверг сентября ежегодно? * * * * 4 logger task
6. Какая команда позволяет вам назначить задание cron для пользователя alice? Приведите подтверждающий пример. * * * * alice logger task
7. Как указать, что пользователю bob никогда не разрешено назначать задания через cron? Приведите подтверждающий пример. Записать его в /etc/cron.deny
8. Вам нужно убедиться, что задание выполняется каждый день, даже если сервер во время выполнения временно недоступен. Как это сделать? Найти задание в логах grep cron /var/log/messages

9. Какая команда позволяет узнать, запланированы ли какие-либо задания на выполнение планировщиком atd? atq

5 Выводы

В ходе выполнения лабораторной работы были получены навыки работы с планировщиками событий `cron` и `at`