

Отчёт по лабораторной работе №8

Планировщики событий

Яковлева Дарья Сергеевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Планирование задач с помощью stop	6
2.2	Планирование заданий с помощью at	10
3	Контрольные вопросы	12
4	Заключение	14

Список иллюстраций

2.1	Содержимое файла <code>/etc/crontab</code>	7
2.2	Редактирование <code>crontab</code> для <code>root</code>	8
2.3	Сообщения из системного журнала, созданные <code>cron</code>	8
2.4	Создание и редактирование сценария <code>eachhour</code>	9
2.5	Создание задания в <code>/etc/cron.d</code>	9
2.6	Работа заданий	10
2.7	Планирование заданий с помощью <code>at</code>	10

Список таблиц

1 Цель работы

Получение навыков работы с планировщиками событий cron и at.

2 Выполнение лабораторной работы

2.1 Планирование задач с помощью cron

Запускаю терминал и получаю права администратора командой `su -`.

Проверяю состояние службы планировщика заданий `crond` с помощью команды `systemctl status crond.service -l`.

Служба активна и запущена (см. рис. fig. 2.1).

Открываю файл конфигурации `/etc/crontab` и просматриваю его содержимое (см. рис. fig. 2.2).

Файл содержит переменные окружения и пример синтаксиса задания `cron`.

```

Password:
root@dsyakovleva:/home/dsyakovleva#
root@dsyakovleva:/home/dsyakovleva# systemctl status crond.service -l
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; preset: enabled)
   Active: active (running) since Sat 2025-10-04 12:01:39 MSK; 2min 39s ago
  Invocation: f793f8c7c8844b5491c52d5ac77ab498
    Main PID: 1244 (crond)
      Tasks: 1 (limit: 24776)
     Memory: 1M (peak: 1.1M)
        CPU: 22ms
    CGroup: /system.slice/crond.service
            └─1244 /usr/sbin/crond -n

Oct 04 12:01:39 dsyakovleva.localdomain systemd[1]: Started crond.service - Command Scheduler.
Oct 04 12:01:39 dsyakovleva.localdomain crond[1244]: (CRON) STARTUP (1.7.0)
Oct 04 12:01:39 dsyakovleva.localdomain crond[1244]: (CRON) INFO (Syslog will be used instead of sendmail)
Oct 04 12:01:39 dsyakovleva.localdomain crond[1244]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 100)
Oct 04 12:01:39 dsyakovleva.localdomain crond[1244]: (CRON) INFO (running with inotify support)
root@dsyakovleva:/home/dsyakovleva# 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@dsyakovleva:/home/dsyakovleva#

```

Рис. 2.1: Содержимое файла /etc/crontab

Открываю файл расписания для пользователя root с помощью команды `crontab -e` и добавляю задание, которое записывает сообщение в системный журнал каждую минуту.

Эта запись означает, что команда выполняется каждую минуту.

Поля расписания интерпретируются следующим образом:

- `*/1` — каждая минута,
- `*` — каждый час,
- `*` — каждый день месяца,
- `*` — каждый месяц,
- `*` — каждый день недели.

Выполняемая команда `logger` записывает сообщение в системный журнал (см. рис. fig. 2.2).

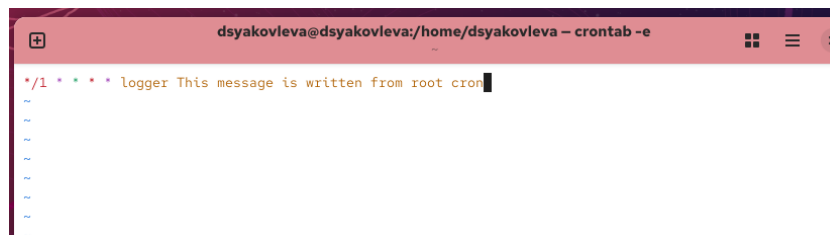


Рис. 2.2: Редактирование crontab для root

После сохранения проверяю текущее расписание (`crontab -l`) и убеждаюсь, что задание добавлено.

Через несколько минут просматриваю журнал `/var/log/messages` и вижу записи, создаваемые cron (см. рис. fig. 2.3).

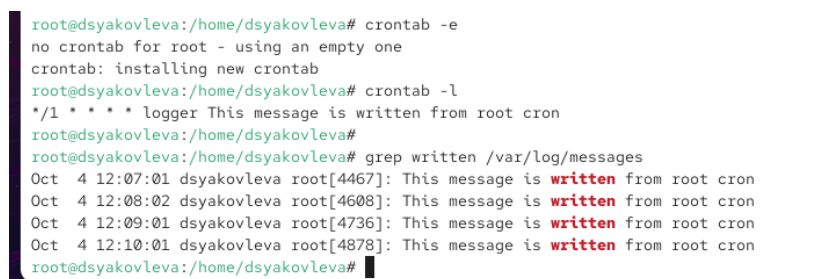
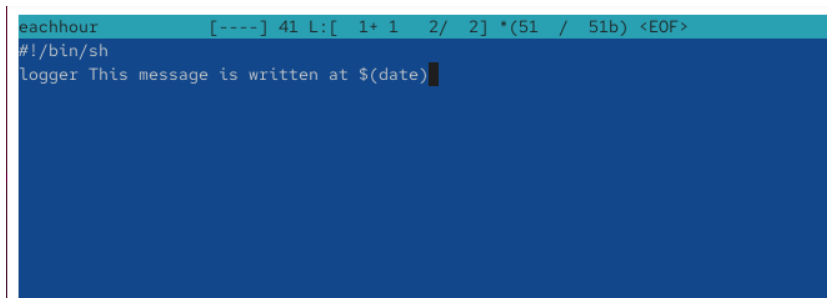


Рис. 2.3: Сообщения из системного журнала, созданные cron

Изменяю расписание задания, чтобы сообщение записывалось в начале каждого часа по будням.

Теперь команда выполняется в 00 минут каждого часа с понедельника по пятницу.

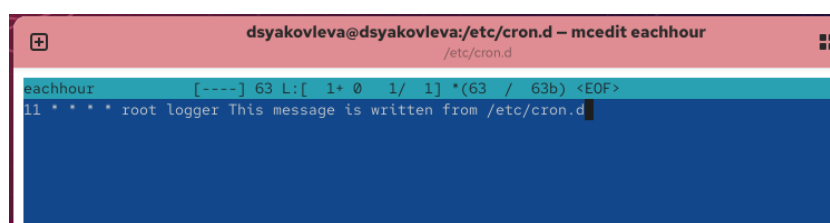
Создаю сценарий `eachhour` в каталоге `/etc/cron.hourly` (см. рис. fig. 2.4). В сценарии реализована запись сообщения в системный журнал с указанием текущей даты и времени. После этого делаю сценарий исполняемым.

A terminal window with a blue background. The prompt is 'eachhour'. The first line shows the file size and line count: '[----] 41 L:[1+ 1 2/ 2] *(51 / 51b) <EOF>'. The second line is the shebang '#!/bin/sh'. The third line is the command 'logger This message is written at \$(date)' followed by a cursor.

```
eachhour [----] 41 L:[ 1+ 1 2/ 2] *(51 / 51b) <EOF>
#!/bin/sh
logger This message is written at $(date)
```

Рис. 2.4: Создание и редактирование сценария eachhour

Затем создаю файл `/etc/cron.d/eachhour` с расписанием, по которому команда `logger` будет выполняться каждый час в 11 минут от имени пользователя `root`. Это позволяет запускать задание централизованно через системный каталог `cron.d` (см. рис. fig. 2.5).

A terminal window titled 'dsyakovleva@dsyakovleva:/etc/cron.d - mcedit eachhour'. The prompt is 'eachhour'. The first line shows the file size and line count: '[----] 63 L:[1+ 0 1/ 1] *(63 / 63b) <EOF>'. The second line is the cron job entry '11 * * * root logger This message is written from /etc/cron.d' followed by a cursor.

```
eachhour [----] 63 L:[ 1+ 0 1/ 1] *(63 / 63b) <EOF>
11 * * * root logger This message is written from /etc/cron.d
```

Рис. 2.5: Создание задания в `/etc/cron.d`

Проверяю журнал `/var/log/messages`, чтобы убедиться, что задания выполняются согласно расписанию.

Сообщения подтверждают успешное выполнение команд `cron` (см. рис. fig. 2.6).

```

root@dsyakovleva:/home/dsyakovleva#
root@dsyakovleva:/home/dsyakovleva# crontab -e
crontab: installing new crontab
Backup of root's previous crontab saved to /root/.cache/crontab/crontab.bak
root@dsyakovleva:/home/dsyakovleva# crontab -l
0 */1 * * 1-5 logger This message is written from root cron
root@dsyakovleva:/home/dsyakovleva# cd /etc/cron.hourly/
root@dsyakovleva:/etc/cron.hourly# touch eachhour
root@dsyakovleva:/etc/cron.hourly# mcedit eachhour

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

root@dsyakovleva:/etc/cron.d# grep written /var/log/messages
Oct  4 12:07:01 dsyakovleva root[4467]: This message is written from root cron
Oct  4 12:08:02 dsyakovleva root[4608]: This message is written from root cron
Oct  4 12:09:01 dsyakovleva root[4736]: This message is written from root cron
Oct  4 12:10:01 dsyakovleva root[4878]: This message is written from root cron
Oct  4 12:11:01 dsyakovleva root[5128]: This message is written from root cron
root@dsyakovleva:/etc/cron.d#

```

Рис. 2.6: Работа заданий

2.2 Планирование заданий с помощью at

Запускаю терминал и получаю права администратора командой `su -`.

Проверяю состояние службы отложенного выполнения заданий `atd` с помощью команды `systemctl status atd`.

Служба активна и работает (см. рис. fig. 2.7).

```

root@dsyakovleva:/etc/cron.d#
root@dsyakovleva:/etc/cron.d# systemctl status atd
● atd.service - Deferred execution scheduler
   Loaded: loaded (/usr/lib/systemd/system/atd.service; enabled; preset: enabled)
   Active: active (running) since Sat 2025-10-04 12:01:39 MSK; 15min ago
  Invocation: da94ea5a8d6b40debe155a24d50ac348
     Docs: man:atd(8)
    Main PID: 1243 (atd)
      Tasks: 1 (limit: 24776)
    Memory: 316K (peak: 1.2M)
       CPU: 11ms
    CGroup: /system.slice/atd.service
            └─1243 /usr/sbin/atd -f

Oct 04 12:01:39 dsyakovleva.localdomain systemd[1]: Started atd.service - Deferred execution scheduler.
Oct 04 12:01:39 dsyakovleva.localdomain (atd)[1243]: atd.service: Referenced but unset environment va
root@dsyakovleva:/etc/cron.d# at 12:19
warning: commands will be executed using /bin/sh
at Sat Oct  4 12:19:00 2025
at> logger message from at
at> <EOT>
job 1 at Sat Oct  4 12:19:00 2025
root@dsyakovleva:/etc/cron.d# atq
1          Sat Oct  4 12:19:00 2025 a root
root@dsyakovleva:/etc/cron.d# grep 'from ar' /var/log/messages
root@dsyakovleva:/etc/cron.d# grep 'from ar' /var/log/messages
root@dsyakovleva:/etc/cron.d# grep 'from at' /var/log/messages
Oct  4 12:19:00 dsyakovleva root[7084]: message from at
root@dsyakovleva:/etc/cron.d#

```

Рис. 2.7: Планирование заданий с помощью at

Для планирования выполнения команды в определённое время использую утилиту `at`.

Задаю выполнение команды `logger message from at` в заданный момент времени (в примере — в 12:19).

После ввода команды `at 12:19` открывается интерактивная оболочка, в которой указываю команду `logger message from at`, затем завершаю ввод комбинацией клавиш **Ctrl + D**.

Появляется подтверждение создания задания с его номером (см. рис. fig. 2.7).

Проверяю список запланированных заданий с помощью команды `atq` — задание отображается в очереди.

После наступления указанного времени проверяю системный журнал `/var/log/messages` при помощи команды `grep 'from at' /var/log/messages`.

Вижу запись о выполнении команды, что подтверждает успешное выполнение задания, запланированного через `at` (см. рис. fig. 2.7).

3 Контрольные вопросы

1. Как настроить задание cron, чтобы оно выполнялось раз в 2 недели?

В cron отсутствует прямой параметр “раз в 2 недели”, но можно использовать комбинацию дней недели и месяца, например:

`0 0 */14 * * команда`

— запуск каждые 14 дней в полночь.

Альтернативно можно добавить проверку даты в скрипте и запускать его раз в 2 недели при помощи `cron.weekly`.

2. Как настроить задание cron, чтобы оно выполнялось 1-го и 15-го числа каждого месяца в 2 часа ночи?

`0 2 1,15 * * команда`

— запуск в 2:00 ночи, 1-го и 15-го числа каждого месяца.

3. Как настроить задание cron, чтобы оно выполнялось каждые 2 минуты каждый день?

`*/2 * * * * команда`

— выполнение каждые 2 минуты, ежедневно.

4. Как настроить задание cron, чтобы оно выполнялось 19 сентября ежегодно?

`0 0 19 9 * команда`

— запуск 19 сентября каждого года в полночь.

5. Как настроить задание cron, чтобы оно выполнялось каждый четверг сентября ежегодно?

0 0 * 9 4 команда

— выполнение в полночь каждого четверга (день недели 4) в сентябре (месяц 9).

6. **Какая команда позволяет вам назначить задание cron для пользователя *alice*? Приведите подтверждающий пример.**

Команда: `crontab -u alice -e`

Пример:

`sudo crontab -u alice -e` — откроет расписание пользователя *alice* для редактирования.

7. **Как указать, что пользователю *bob* никогда не разрешено назначать задания через cron? Приведите подтверждающий пример.**

Добавить имя пользователя в файл `/etc/cron.deny`: `echo "bob" >> /etc/cron.deny`

После этого пользователь *bob* не сможет использовать `crontab`.

8. **Вам нужно убедиться, что задание выполняется каждый день, даже если сервер во время выполнения временно недоступен. Как это сделать?**

Использовать `anacron`, который гарантирует выполнение пропущенных заданий после включения системы.

Конфигурационные файлы: `/etc/anacrontab`.

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

Команда: `atq`

— показывает очередь заданий `at`, ожидающих выполнения.

4 Заключение

В ходе лабораторной работы были изучены принципы планирования заданий в Linux с использованием планировщиков `cron` и `at`.

Были освоены способы создания, редактирования и проверки расписаний для периодических и однократных задач,

а также работа с системными каталогами `/etc/cron.*` и файлами `/etc/crontab` и `/etc/cron.d`.

Получены практические навыки настройки интервалов выполнения, ограничения доступа пользователей и проверки выполнения заданий через системные журналы.