

Отчёт по лабораторной работе №8

Планировщики событий

Ришард Когенгар

Содержание

1 Цель работы	5
2 Ход выполнения	6
2.1 Планирование задач с помощью cron	6
2.2 Планирование заданий с помощью at	10
2.3 Вывод	11
3 Контрольные вопросы	13

Список иллюстраций

2.1	Проверка состояния службы crond	6
2.2	Просмотр содержимого файла /etc/crontab	7
2.3	Редактирование пользовательского crontab	8
2.4	Проверка выполнения задания cron	8
2.5	Изменение задания cron	9
2.6	Создание сценария eachhour	9
2.7	Настройка задания в /etc/cron.d	10
2.8	Проверка выполнения задания at	11

Список таблиц

1 Цель работы

Получение навыков работы с планировщиками событий cron и at.

2 Ход выполнения

2.1 Планирование задач с помощью cron

1. В терминале были получены привилегии администратора (root) с помощью команды su.

После этого выполнена проверка состояния демона планировщика заданий crond командой systemctl status crond -l.

По результатам проверки установлено, что служба **crond.service** загружена, включена в автозапуск (**enabled**) и находится в состоянии **active (running)**, что подтверждает её корректную работу.

```
rishard@rishardkogengar:~$ su
Password:
root@rishardkogengar:/home/rishard# systemctl status crond -l
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; preset: enabled)
   Active: active (running) since Tue 2026-01-20 11:34:28 MSK; 7min ago
     Invocation: 19c6afb6155d462097c58580777c95cb
   Main PID: 1442 (crond)
      Tasks: 1 (limit: 23027)
        Memory: 1M (peak: 1.2M)
          CPU: 5ms
        CGroup: /system.slice/crond.service
                  └─1442 /usr/sbin/crond -n

Jan 20 11:34:28 rishardkogengar.localdomain systemd[1]: Started crond.service - Command Scheduler.
Jan 20 11:34:28 rishardkogengar.localdomain crond[1442]: (CRON) STARTUP (1.7.0)
Jan 20 11:34:28 rishardkogengar.localdomain crond[1442]: (CRON) INFO (Syslog will be used instead of sendmail.)
Jan 20 11:34:28 rishardkogengar.localdomain crond[1442]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 42% if used.)
Jan 20 11:34:28 rishardkogengar.localdomain crond[1442]: (CRON) INFO (running with inotify support)
root@rishardkogengar:/home/rishard#
```

Рис. 2.1: Проверка состояния службы crond

2. Для ознакомления с глобальным конфигурационным файлом cron выполнен просмотр содержимого файла /etc/crontab.

В файле заданы основные переменные окружения (**SHELL**, **PATH**, **MAILTO**), а также приведено описание формата записи заданий cron, включающее поля минут, часов, дня месяца, месяца, дня недели и имени пользователя.

```
-----  
root@rishardkogengar:/home/rishard# 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@rishardkogengar:/home/rishard# █
```

Рис. 2.2: Просмотр содержимого файла /etc/crontab

3. Далее выполнен просмотр пользовательского расписания заданий командой **crontab -l**.

В результате было получено сообщение об отсутствии заданий, что означает, что для пользователя root расписание ещё не было создано.

4. Для создания нового задания выполнено редактирование пользовательского файла crontab командой **crontab -e**.

В редакторе добавлена следующая строка:

```
*/1 * * * * logger This message is written from root cron
```

Пояснение синтаксиса записи:

- ***/1** — выполнение каждую минуту;
- ***** в остальных полях — отсутствие ограничений по времени и дате;
- **logger** — команда записи сообщения в системный журнал.

The screenshot shows a terminal window with the title bar "rishard@rishardkogengar:/home/rishard - crontab -e". The main area of the terminal contains the following text:

```
*/1 * * * * logger This message is written from root cron
```

Рис. 2.3: Редактирование пользовательского crontab

5. После сохранения изменений выполнен повторный просмотр расписания командой `crontab -l`.

В списке появилась добавленная запись, что подтверждает успешное сохранение задания.

6. Через несколько минут без перезагрузки системы выполнен просмотр системного журнала командой `grep written /var/log/messages`.

В журнале зафиксированы повторяющиеся записи с сообщением **This message is written from root cron**, что подтверждает корректное выполнение задания по расписанию.

```
root@rishardkogengar:/home/rishard# 
root@rishardkogengar:/home/rishard# crontab -l
no crontab for root
root@rishardkogengar:/home/rishard# crontab -e
no crontab for root - using an empty one
crontab: installing new crontab
root@rishardkogengar:/home/rishard# crontab -l
*/1 * * * * logger This message is written from root cron
root@rishardkogengar:/home/rishard# grep written /var/log/messages
Jan 20 11:44:01 rishardkogengar root[5016]: This message is written from root cron
root@rishardkogengar:/home/rishard# grep written /var/log/messages
Jan 20 11:44:01 rishardkogengar root[5016]: This message is written from root cron
Jan 20 11:45:01 rishardkogengar root[5153]: This message is written from root cron
Jan 20 11:46:01 rishardkogengar root[5280]: This message is written from root cron
Jan 20 11:47:01 rishardkogengar root[5406]: This message is written from root cron
root@rishardkogengar:/home/rishard#
```

Рис. 2.4: Проверка выполнения задания cron

7. Далее запись в пользовательском расписании была изменена на следующую:

```
0 */1 * * 1-5 logger This message is written from root cron
```

Пояснение синтаксиса:

- 0 — запуск в нулевую минуту часа;
- */1 — выполнение каждый час;
- 1-5 — выполнение только в рабочие дни (с понедельника по пятницу).

```
rishard@rishardkogengar:/home/rishard - crontab -e
0 */1 * * 1-5 logger This message is written from root cron
```

Рис. 2.5: Изменение задания cron

8. В каталоге /etc/cron.hourly был создан файл сценария с именем eachhour.

В файл внесён сценарий, выполняющий запись сообщения в системный журнал с указанием текущей даты и времени.

```
GNU nano 8.1                                     eachhour
#!/bin/sh
logger This message is written at $(date)
```

Рис. 2.6: Создание сценария eachhour

while 9. В каталоге /etc/cron.d создан файл расписания eachhour, в который добавлена строка:

```
11 * * * * root logger This message is written from /etc/cron.d
```

Пояснение синтаксиса записи: - 11 – выполнение на 11-й минуте каждого часа; - * * * * – отсутствие ограничений по дате и дню недели; - root – пользователь, от имени которого выполняется команда; - logger – запись сообщения в системный журнал.



The screenshot shows a terminal window titled 'rishard@rishardkogengar:/etc/cron.d – nano eachhour /etc/cron.d'. The terminal displays the command 'GNU nano 8.1' and the cron entry '11 * * * * root logger This message is written from /etc/cron.d'. The cursor is positioned at the end of the line.

Рис. 2.7: Настройка задания в /etc/cron.d

10. По прошествии требуемого времени выполнен просмотр системного журнала.

По журналу установлено, что задания, созданные через пользовательский crontab и каталог /etc/cron.d, выполнялись в соответствии с данным расписанием.

2.2 Планирование заданий с помощью at

1. В терминале получены полномочия администратора и выполнена проверка состояния службы отложенного выполнения заданий atd.
Служба **atd.service** загружена, включена в автозапуск и находится в состоянии **active (running)**.
2. Создано разовое задание с помощью команды at на заданное время.

В интерактивном режиме указана команда `logger message from at`, после чего ввод был завершён.

3. Для проверки наличия задания в очереди использована команда `atq`.

В списке отображено созданное задание с указанием времени запуска и пользователя root.

4. После наступления заданного времени выполнен просмотр системного журнала.

В журнале зафиксировано сообщение **message from at**, что подтверждает успешное выполнение разового задания.

```
root@rishardkogengar:/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 Tue 2026-01-20 11:34:28 MSK; 19min ago
    Invocation: b6b6f664d6224e178c38eaaf4dcc91285
   Docs: man:atd(8)
 Main PID: 1440 (atd)
   Tasks: 1 (limit: 23027)
  Memory: 320K (peak: 1.2M)
    CPU: 2ms
   CGroup: /system.slice/atd.service
           └─1440 /usr/sbin/atd -f

Jan 20 11:34:28 rishardkogengar.localdomain systemd[1]: Started atd.service - Deferred execution scheduler.
Jan 20 11:34:28 rishardkogengar.localdomain (atd)[1440]: atd.service: Referenced but unset environment variable evaluates
root@rishardkogengar:/etc/cron.d# at 11:56
warning: commands will be executed using /bin/sh
at Tue Jan 20 11:56:00 2026
at> logger message from at
at> <EOT>
job 1 at Tue Jan 20 11:56:00 2026
root@rishardkogengar:/etc/cron.d# atq
1      Tue Jan 20 11:56:00 2026 a root
root@rishardkogengar:/etc/cron.d# grep 'fom at' /var/log/messages
root@rishardkogengar:/etc/cron.d# grep 'fom at' /var/log/messages
root@rishardkogengar:/etc/cron.d# grep 'fom at' /var/log/messages
root@rishardkogengar:/etc/cron.d# grep 'from at' /var/log/messages
Jan 20 11:56:00 rishardkogengar root[6855]: message from at
root@rishardkogengar:/etc/cron.d#
```

Рис. 2.8: Проверка выполнения задания at

2.3 Вывод

В ходе работы были изучены и отработаны механизмы планирования задач в операционной системе Linux с использованием утилит **cron** и **at**.

Было подтверждено, что демон **crond** корректно выполняет периодические

задания как из пользовательского расписания, так и из системных каталогов /etc/cron.hourly и /etc/cron.d.

Также была проверена работа службы **atd**, обеспечивающей однократный запуск команд в заданное время.

3 Контрольные вопросы

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

В cron отсутствует прямое поле для указания интервала «раз в две недели», поэтому используется комбинация дня недели и номера недели месяца либо конкретной даты старта.

На практике чаще всего задание запускают раз в 14 дней, начиная с определённой даты, с помощью проверки даты в команде или через anacron.

Пример (каждые 14 дней, начиная с 1 января 2026 года): `0 2 */14 * * command`
Это означает запуск в 02:00 каждые 14 дней, считая от начала месяца.

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

Пример записи в crontab: `0 2 1,15 * * command`

Пояснение: - 0 – нулевая минута; - 2 – 2 часа ночи; - 1,15 – первое и пятнадцатое число месяца; - * * – любой месяц и любой день недели.

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

Пример записи: `*/2 * * * * command`

Пояснение: - */2 – каждые две минуты; - остальные поля * – без ограничений по времени и дате.

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

Пример записи: `0 0 19 9 * command`

Пояснение: - 0 0 – запуск в 00:00; - 19 – девятнадцатое число; - 9 – сентябрь; -

* — любой день недели.

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

Пример записи: 0 2 * 9 4 command

Пояснение: - 0 2 — запуск в 02:00; - * — любой день месяца; - 9 — сентябрь; - 4 — четверг (0 — воскресенье, 4 — четверг).

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

Для назначения задания cron другому пользователю используется команда с ключом -u:

```
crontab -u alice -e
```

Пример: администратор открывает расписание пользователя alice и добавляет туда задание: 0 1 * * * /usr/bin/backup.sh

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

Для запрета используется файл /etc/cron.deny.

Если имя пользователя присутствует в этом файле, ему запрещено использовать cron.

Пример: В файл /etc/cron.deny добавляется строка: bob

После этого пользователь bob не сможет использовать команду crontab.

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

Для этой цели используется утилита **anacron**, предназначенная для выполнения периодических заданий на системах, которые могут быть выключены.

Anacron запускает пропущенные задания при следующем включении системы.

Пример: настройка ежедневного задания в файле /etc/anacrontab.

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

Для просмотра очереди заданий планировщика at используется команда: atq

Она выводит список всех запланированных разовых заданий с указанием времени выполнения и пользователя.