

Отчёт по лабораторной работе №8

Планировщики событий

Амина Аджигалиева

Содержание

1	Цель работы	5
2	Ход выполнения работы	6
2.1	Планирование задач с помощью <code>stop</code>	6
2.2	Планирование заданий с помощью <code>at</code>	9
3	Контрольные вопросы	11
4	Заключение	13

Список иллюстраций

2.1	Проверка состояния службы crond	6
2.2	Создание нового задания в crontab	7
2.3	Проверка выполнения задания через журнал /var/log/messages . .	8
2.4	Изменение расписания выполнения задачи	8
2.5	Создание и редактирование скрипта eachhour	9
2.6	Создание задания в каталоге /etc/cron.d	9
2.7	Планирование и выполнение одноразового задания с помощью at	10

Список таблиц

1 Цель работы

Получение навыков работы с планировщиками событий cron и at.

2 Ход выполнения работы

2.1 Планирование задач с помощью cron

Сначала я проверила состояние службы планировщика заданий crond с помощью команды `systemctl status crond -l`.

Вывод показал, что служба активна (running) и включена для автозапуска.

```
root@aradzhigalieva:~# systemctl status crond -l
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-10-06 12:10:43 MSK; 4min 1s ago
  Invocation: 01e4286cf5014a3ba9fd8a5cc3d8614c
    Main PID: 1212 (crond)
      Tasks: 1 (limit: 24779)
     Memory: 1M (peak: 1.1M)
        CPU: 7ms
    CGroup: /system.slice/crond.service
            └─1212 /usr/sbin/crond -n

Notice: journal has been rotated since unit was started, output may be incomplete.
root@aradzhigalieva:~#
root@aradzhigalieva:~# crontab -l
no crontab for root
root@aradzhigalieva:~# 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@aradzhigalieva:~#
```

Рис. 2.1: Проверка состояния службы crond

Затем я просмотрела содержимое системного файла конфигурации `/etc/crontab`, чтобы убедиться в корректности настроек среды и формате задания расписания.

Файл содержал примеры структуры задания cron и стандартные переменные окружения.

Проверка существующих заданий для пользователя root через команду `crontab -l` показала, что расписание пустое.

После этого я открыла файл `crontab` для редактирования и добавила строку, выполняющую запись сообщения в системный журнал каждую минуту.

Эта запись означает выполнение команды каждую минуту.

Синтаксис строки: `- /1` — *каждая минута*

- — каждый час, день, месяц и день недели

- `logger` — отправка сообщения в системный журнал



Рис. 2.2: Создание нового задания в `crontab`

После сохранения изменений команда `crontab -l` отобразила добавленное задание.

Через несколько минут я выполнила проверку журнала системных событий, чтобы убедиться, что сообщения действительно записываются каждую минуту.

Логи подтвердили успешное выполнение задачи.

```

root@aradzhigalieva: ~
root@aradzhigalieva:~# crontab -e
no crontab for root - using an empty one
crontab: installing new crontab
root@aradzhigalieva:~# crontab -l
*/1 * * * * logger This message is written from root cron
root@aradzhigalieva:~# grep written /var/log/messages
Oct 6 12:18:01 aradzhigalieva root[4527]: This message is written from root cron
root@aradzhigalieva:~# grep written /var/log/messages
Oct 6 12:18:01 aradzhigalieva root[4527]: This message is written from root cron
Oct 6 12:19:01 aradzhigalieva root[4662]: This message is written from root cron
Oct 6 12:20:01 aradzhigalieva root[4789]: This message is written from root cron
root@aradzhigalieva:~# grep written /var/log/messages
Oct 6 12:18:01 aradzhigalieva root[4527]: This message is written from root cron
Oct 6 12:19:01 aradzhigalieva root[4662]: This message is written from root cron
Oct 6 12:20:01 aradzhigalieva root[4789]: This message is written from root cron
Oct 6 12:21:02 aradzhigalieva root[4905]: This message is written from root cron
root@aradzhigalieva:~# █

```

Рис. 2.3: Проверка выполнения задания через журнал /var/log/messages

Затем я изменила периодичность задания, установив выполнение в начале каждого часа с понедельника по пятницу.

Синтаксис новой строки: - 0 — минута запуска

- */1 — каждый час

- 1-5 — с понедельника по пятницу

Я проверила новое расписание с помощью команды `crontab -l` и убедилась, что изменения применены.

```

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

```

Рис. 2.4: Изменение расписания выполнения задачи

Далее я перешла в каталог `/etc/cron.hourly` и создала файл сценария `eachhour`. В файл был добавлен скрипт, записывающий сообщение в системный журнал с текущей датой и временем при каждом запуске.

Этот сценарий выполняется системой автоматически каждый час.



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

Рис. 2.5: Создание и редактирование скрипта eachhour

После этого я перешла в каталог `/etc/cron.d` и создала файл расписания `eachhour`, в который добавила строку, запускающую команду каждый час в 11-й минуте от имени пользователя `root`.

Таким образом, система будет выполнять указанную задачу каждый час в заданное время.



```
root@aradzhigalieva:/etc/cron.d - nano eachhour
GNU nano 8.1      eachhour
11 * * * * root logger This message is written from /etc/cron.d
```

Рис. 2.6: Создание задания в каталоге `/etc/cron.d`

2.2 Планирование заданий с помощью `at`

Для планирования однократного выполнения команд я использовала службу `atd`.

Сначала я проверила её состояние, убедившись, что служба активна и работает.

Затем я задала выполнение команды `logger message from at` в определённое время.

После завершения ввода задание появилось в списке запланированных, что подтвердилось командой `atq`.

Через несколько минут сообщение появилось в системном журнале, что свидетельствует об успешном выполнении задачи.

```
root@aradzhigalieva:/etc/cron.d#
root@aradzhigalieva:/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 Mon 2025-10-06 12:10:43 MSK; 15min ago
 Invocation: 53e4458921cf4ff4b1f9291c75951666
    Docs: man:atd(8)
   Main PID: 1210 (atd)
     Tasks: 1 (limit: 24779)
    Memory: 320K (peak: 1.1M)
       CPU: 4ms
    CGroup: /system.slice/atd.service
            └─1210 /usr/sbin/atd -f

Notice: journal has been rotated since unit was started, output may be incomplete.
root@aradzhigalieva:/etc/cron.d# at 12:29
warning: commands will be executed using /bin/sh
at Mon Oct  6 12:29:00 2025
at> logger message from at
at> <EOT>
job 1 at Mon Oct  6 12:29:00 2025
root@aradzhigalieva:/etc/cron.d# atq
1          Mon Oct  6 12:29:00 2025 a root
root@aradzhigalieva:/etc/cron.d# grep 'from at' /var/log/messages
root@aradzhigalieva:/etc/cron.d# grep 'from at' /var/log/messages
Oct  6 12:29:00 aradzhigalieva root[6167]: message from at
root@aradzhigalieva:/etc/cron.d# █
```

Рис. 2.7: Планирование и выполнение одноразового задания с помощью `at`

3 Контрольные вопросы

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

Можно указать выполнение задания в определённый день недели с интервалом через 14 дней, например:

0 0 */14 * * команда — выполняется раз в две недели в полночь.

Однако cron не поддерживает прямую запись “раз в 2 недели”, поэтому обычно задают выполнение в конкретный день недели и контролируют через скрипт.

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

Здесь 0 — минуты, 2 — часы, 1 и 15 — числа месяца, * — каждый месяц, * — любой день недели.

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

*/2 * * * * команда

Символ */2 означает интервал в две минуты.

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

0 0 19 9 * команда

Здесь 0 — минуты, 0 — часы, 19 — день месяца, 9 — месяц (сентябрь), * — любой день недели.

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

0 0 * 9 4 команда

0 — минуты, 0 — часы, * — любой день месяца, 9 — сентябрь, 4 — четверг (где 0 или 7 — воскресенье, 1 — понедельник и т.д.).

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

Используется команда `crontab -u alice -e`

Пример:

`crontab -u alice -e` — откроет файл расписания пользователя alice для редактирования.

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

В файл `/etc/cron.deny` нужно добавить имя пользователя bob.

Пример содержимого файла:

bob

После этого пользователь bob не сможет создавать задания cron.

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

Можно использовать службу `anacron`, которая запускает пропущенные задания после восстановления работы системы.

В отличие от `cron`, `anacron` не зависит от непрерывной работы системы.

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

Для этого используется команда `atq`, которая отображает список всех запланированных заданий для текущего пользователя.

4 Заключение

В ходе выполнения лабораторной работы я изучила механизм планирования заданий в операционной системе Linux с помощью служб **cron** и **at**.

Я научилась создавать и редактировать расписания выполнения задач через файл `crontab`, настраивать периодические и однократные задания, а также использовать системные каталоги `/etc/cron.hourly` и `/etc/cron.d` для организации автоматических запусков.

В результате работы были получены практические навыки администрирования системных задач, что является важным элементом автоматизации обслуживания и мониторинга серверов Linux.