Отчёт по лабораторной работе №8

Планировщики событий

Амина Аджигалиева

Содержание

# 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.

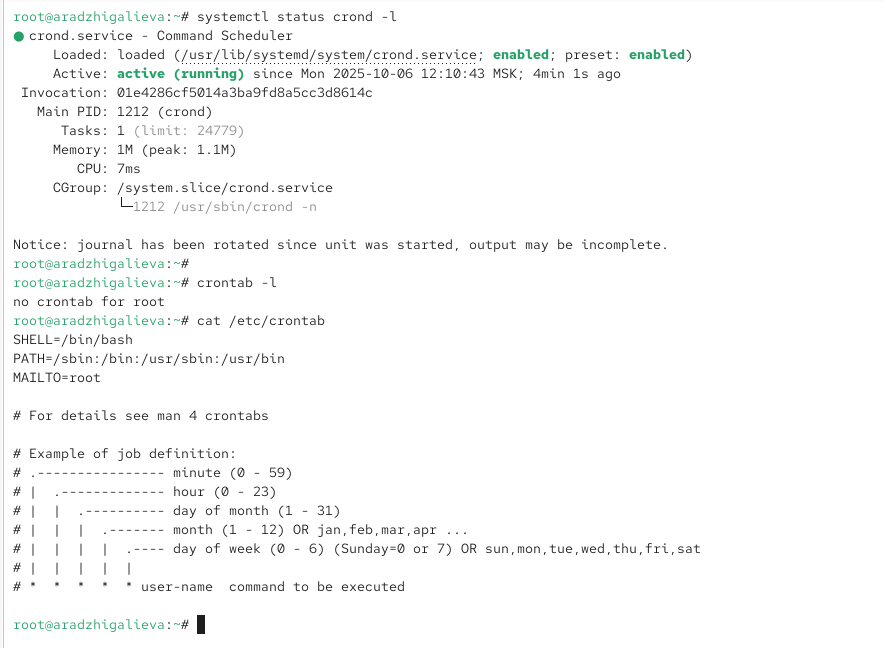


Рис. 1: Проверка состояния службы crond

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

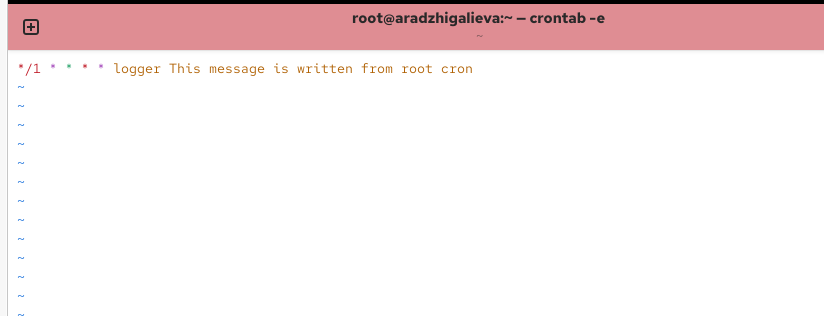


Рис. 2: Создание нового задания в crontab

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.

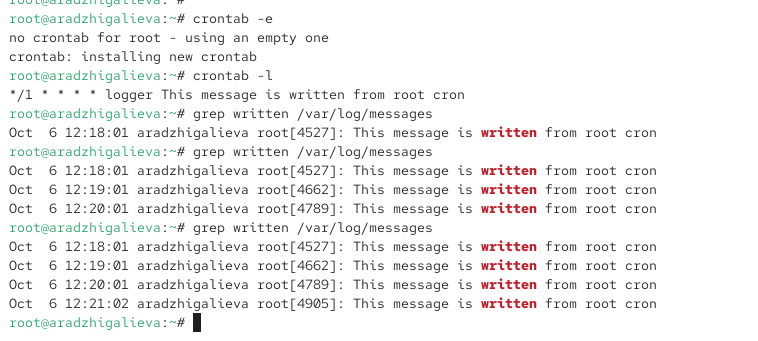


Рис. 3: Проверка выполнения задания через журнал /var/log/messages

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.

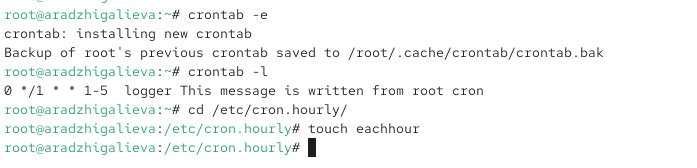


Рис. 4: Изменение расписания выполнения задачи

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.

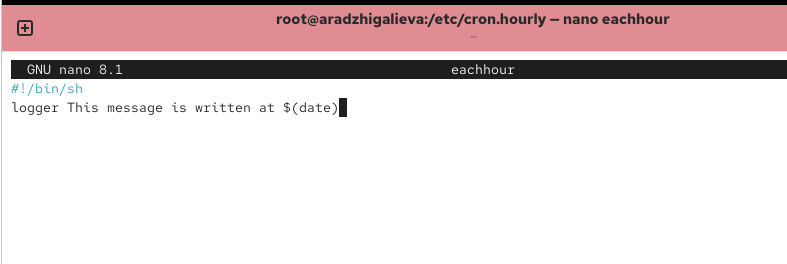


Рис. 5: Создание и редактирование скрипта eachhour

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.



Рис. 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.

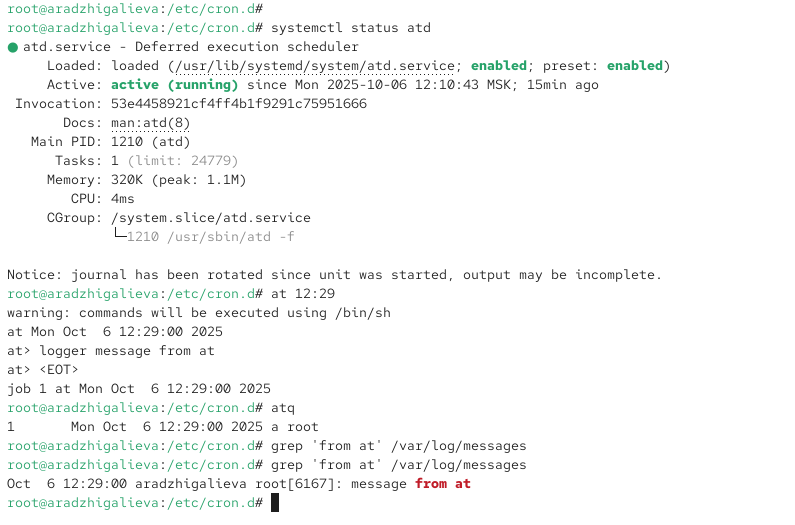


Рис. 7: Планирование и выполнение одноразового задания с помощью at

# 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.