Отчёт по лабораторной работе №8

Планировщики событий

Анастасия Новикова

Содержание

# 1 Цель работы

Получение навыков работы с планировщиками событий cron и at.

# 2 Ход выполнения

## 2.1 Планирование задач с помощью cron

Для начала был запущен терминал и выполнен переход в режим администратора с помощью команды su -.  
Далее проверен статус демона crond, отвечающего за выполнение задач по расписанию. Команда systemctl status crond -l показала, что служба активна и включена при загрузке системы (active (running) и enabled).

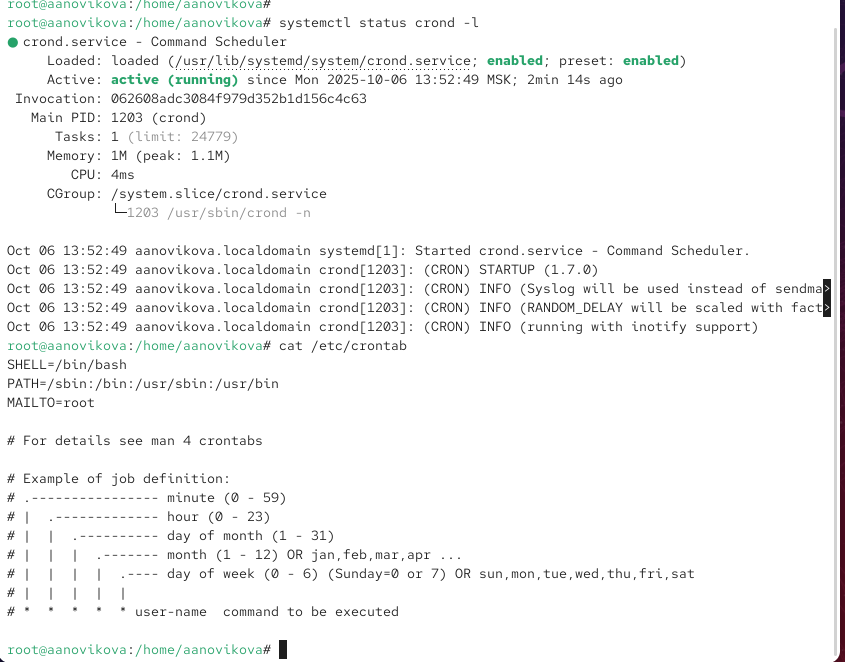


Рис. 1: Статус службы crond

После этого было просмотрено содержимое файла /etc/crontab, в котором заданы общие параметры среды и приведён пример синтаксиса записи расписания заданий.

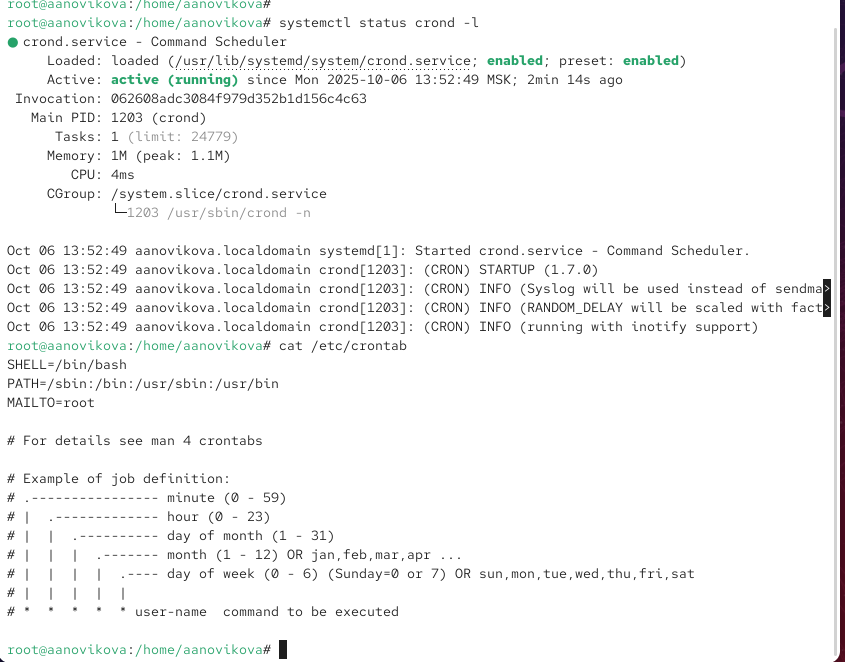


Рис. 2: Просмотр содержимого /etc/crontab

Далее была проверена текущая таблица заданий командой crontab -l, которая сообщила об отсутствии задач для пользователя root.  
Для добавления новой записи был выполнен crontab -e, после чего в открывшемся редакторе добавлена строка:

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

Эта запись означает выполнение команды каждую минуту (\*/1) без ограничений по часам, дням, месяцам и дням недели.  
Команда logger добавляет сообщение в системный журнал.

После сохранения изменений и проверки содержимого (crontab -l) запись успешно отобразилась.

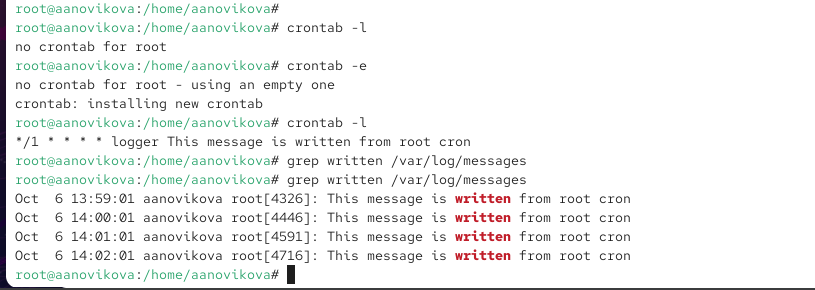


Рис. 3: Создание и проверка задания cron

Через несколько минут команда grep written /var/log/messages подтвердила срабатывание расписания — в журнале появились строки с сообщением, записанным от имени root.

Затем запись была изменена на следующую:

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

Эта конфигурация означает запуск задания в начале каждого часа (минуты 0), каждый час (\*/1), по будним дням (с понедельника по пятницу — 1-5).

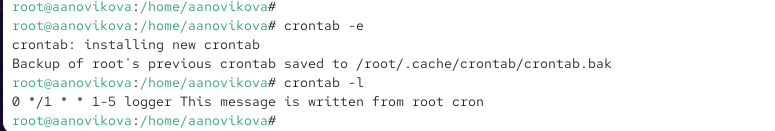


Рис. 4: Редактирование расписания root cron

## 2.2 Использование каталогов /etc/cron.hourly и /etc/cron.d

В каталоге /etc/cron.hourly был создан файл сценария eachhour.  
Он открывался на редактирование и получил следующий скрипт:

#!/bin/sh  
logger This message is written at $(date)

Данный скрипт записывает сообщение в системный журнал с указанием текущей даты.



Рис. 5: Создание скрипта eachhour в /etc/cron.hourly

После сохранения файл был сделан исполняемым с помощью chmod +x eachhour.

Затем в каталоге /etc/cron.d был создан файл с тем же именем eachhour, в который добавлена строка:

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

Это означает, что команда будет выполняться каждый час, на 11-й минуте, от имени пользователя root.

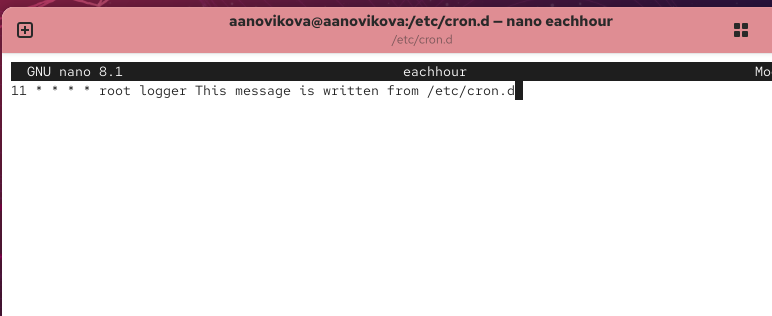


Рис. 6: Создание задания eachhour в /etc/cron.d

## 2.3 Планирование заданий с помощью at

Для однократного выполнения задач используется служба atd.  
Её состояние было проверено командой systemctl status atd, которая показала, что служба активна (active (running)).

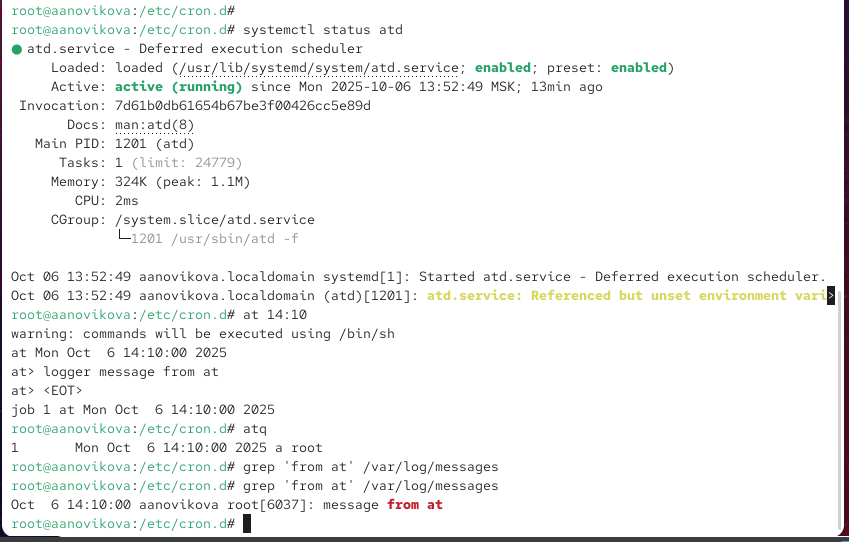


Рис. 7: Проверка статуса службы atd

Далее было задано выполнение команды logger message from at в определённое время (например, 14:10) при помощи команды:

at 14:10  
logger message from at

Задание успешно добавлено в очередь (atq), а через указанное время в системном журнале появилось сообщение о его выполнении.  
Проверка командой grep ‘from at’ /var/log/messages подтвердила выполнение задания.

# 3 Контрольные вопросы

**1. Как настроить задание cron, чтобы оно выполнялось раз в 2 недели?**  
В стандартном cron отсутствует прямой параметр для запуска раз в две недели.  
Реализуется через указание определённого дня недели и использование условий в скрипте или через ключ \*/14 в поле дней месяца:  
0 0 \*/14 \* \* команда — выполнение каждые 14 дней в полночь.

**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  
Задание будет выполняться дважды в месяц — 1-го и 15-го числа в 02:00.

**3. Как настроить задание cron, чтобы оно выполнялось каждые 2 минуты каждый день?**  
\*/2 \* \* \* \* команда  
Поля означают: каждые 2 минуты, каждый час, каждый день, каждый месяц.

**4. Как настроить задание cron, чтобы оно выполнялось 19 сентября ежегодно?**  
0 0 19 9 \* команда  
Выполнение один раз в год — 19 сентября в полночь.

**5. Как настроить задание cron, чтобы оно выполнялось каждый четверг сентября ежегодно?**  
0 0 \* 9 4 команда  
Поле месяца — 9 (сентябрь), день недели — 4 (четверг).

**6. Какая команда позволяет вам назначить задание cron для пользователя alice? Приведите подтверждающий пример.**  
Команда выполняется от имени администратора:  
crontab -u alice -e  
Она открывает таблицу заданий для пользователя alice.  
Пример записи:  
30 9 \* \* \* logger "Запуск cron для пользователя alice"

**7. Как указать, что пользователю bob никогда не разрешено назначать задания через cron? Приведите подтверждающий пример.**  
Необходимо добавить имя пользователя bob в файл /etc/cron.deny.  
Пример содержимого:  
bob  
Это полностью запрещает пользователю bob использовать crontab.

**8. Вам нужно убедиться, что задание выполняется каждый день, даже если сервер во время выполнения временно недоступен. Как это сделать?**  
Следует использовать пакет **anacron**, который запускает пропущенные задания после восстановления работы системы.  
Файлы заданий располагаются в /etc/cron.daily, /etc/cron.weekly, /etc/cron.monthly.

**9. Какая команда позволяет узнать, запланированы ли какие-либо задания на выполнение планировщиком atd?**  
Команда atq выводит список всех ожидающих выполнения заданий, запланированных с помощью at.

# 4 Заключение

В ходе выполнения лабораторной работы была изучена и отработана практика использования планировщиков задач в Linux — **cron** и **at**.  
Были рассмотрены механизмы запуска периодических заданий, редактирование расписаний через crontab, настройка системных каталогов /etc/cron.hourly и /etc/cron.d, а также планирование однократных задач с помощью службы **atd**.