Отчёт по лабораторной работе №8

Планировщики событий

Наурузова Айшат Магометовна

Содержание

# 1 Цель работы

Получение навыков работы с планировщиками событий cron и at.

# 2 Ход выполнения

## 2.1 Планирование задач с помощью cron

После входа в систему были получены права администратора:

su -

### 2.1.1 Проверка статуса службы cron

Для проверки состояния службы crond была использована команда:

systemctl status crond -l

Вывод показывает, что служба crond загружена и активно работает:

* Loaded: loaded
* Active: active (running)

Это подтверждает, что планировщик заданий запущен и готов к работе.

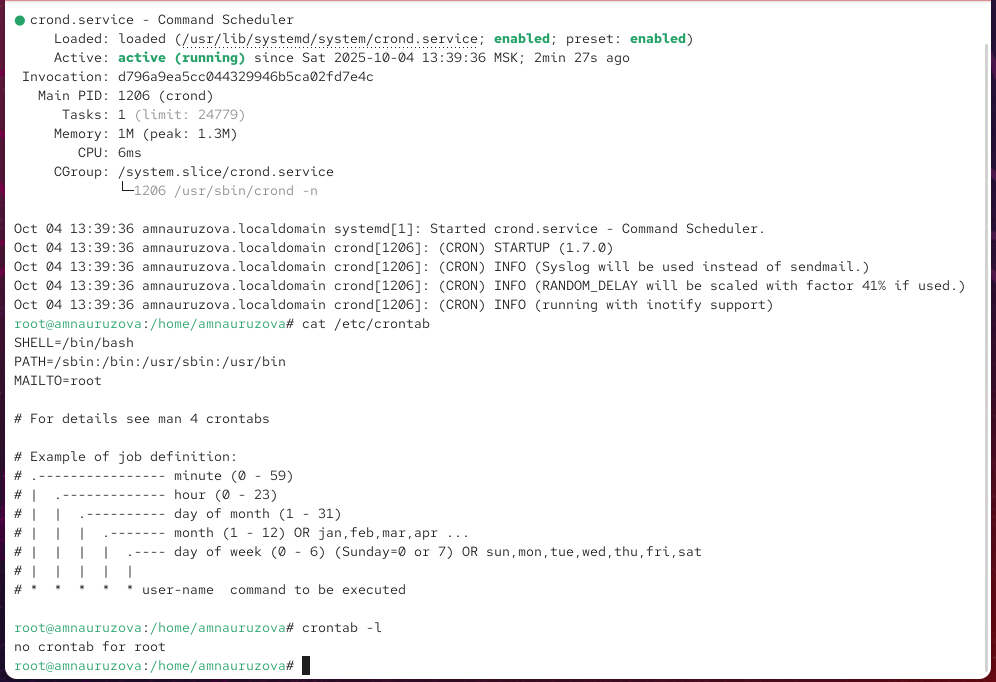


Рис. 1: Проверка статуса службы crond

### 2.1.2 Просмотр конфигурации и текущего расписания

Далее было просмотрено содержимое системного файла конфигурации /etc/crontab:

cat /etc/crontab

Файл содержит стандартные переменные окружения (SHELL, PATH, MAILTO) и пример шаблона задания.

Также проверен список текущих заданий пользователя root:

crontab -l

Ответ системы — “no crontab for root”, что означает отсутствие индивидуальных задач.

### 2.1.3 Добавление нового задания

Для создания нового расписания была выполнена команда:

crontab -e

В открывшемся редакторе добавлена строка:

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

Эта запись означает, что команда logger будет выполняться каждую минуту, добавляя сообщение в системный журнал.

**Расшифровка синтаксиса:**

| Поле | Значение | Описание |
| --- | --- | --- |
| \*/1 | минута | выполнение каждые 1 минуту |
| \* | час | каждый час |
| \* | день месяца | каждый день |
| \* | месяц | каждый месяц |
| \* | день недели | любой день недели |

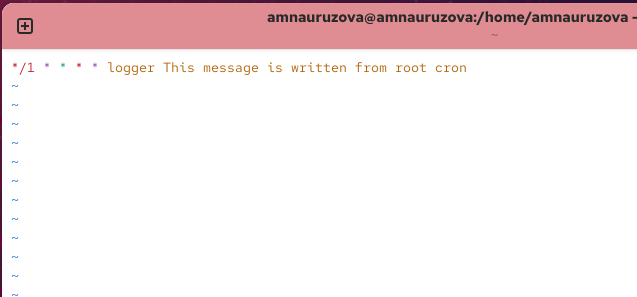


Рис. 2: Создание задания в crontab

После сохранения проверка через crontab -l показывает добавленную строку.  
Через несколько минут в системном журнале появились записи с сообщением, что подтверждает выполнение задания:

grep written /var/log/messages

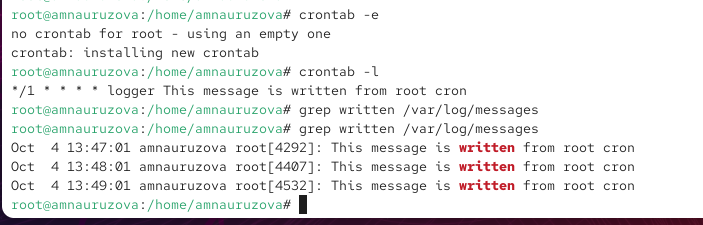


Рис. 3: Проверка выполнения задания cron

### 2.1.4 Изменение расписания cron

Далее задание было отредактировано командой:

crontab -e

Строка изменена на:

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

**Пояснение синтаксиса:**

| Поле | Значение | Описание |
| --- | --- | --- |
| 0 | минута | запуск в начале часа |
| \*/1 | час | каждый час |
| \* | день месяца | каждый день |
| \* | месяц | каждый месяц |
| 1-5 | день недели | с понедельника по пятницу |

Таким образом, команда logger будет выполняться ежечасно в будние дни.

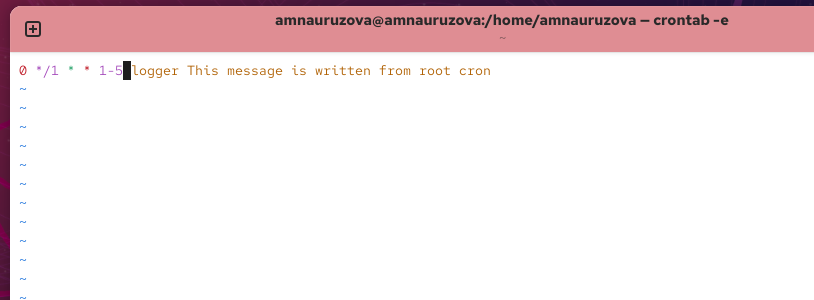


Рис. 4: Редактирование расписания cron

### 2.1.5 Использование каталогов cron.hourly и cron.d

В каталоге /etc/cron.hourly создан сценарий eachhour:

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

Скрипт отправляет в системный журнал сообщение с текущей датой и временем.  
Для корректного выполнения скрипт был сделан исполняемым:

chmod +x eachhour



Рис. 5: Создание сценария eachhour в cron.hourly

Затем в каталоге /etc/cron.d создан одноимённый файл с расписанием eachhour со следующим содержимым:

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

**Пояснение синтаксиса:**

| Поле | Значение | Описание |
| --- | --- | --- |
| 11 | минута | выполнение в 11-й минуте каждого часа |
| \* \* \* \* | остальные поля | любой час, день, месяц, день недели |
| root | пользователь | запуск от имени root |
| logger … | команда | запись сообщения в журнал |



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

После ожидания указанного времени соответствующие записи появились в системном журнале, что подтверждает работу задания.

## 2.2 Планирование заданий с помощью at

Для разового выполнения задач используется служба atd.

Сначала проверен её статус:

systemctl status atd

Служба активна (active (running)), что видно из вывода команды.

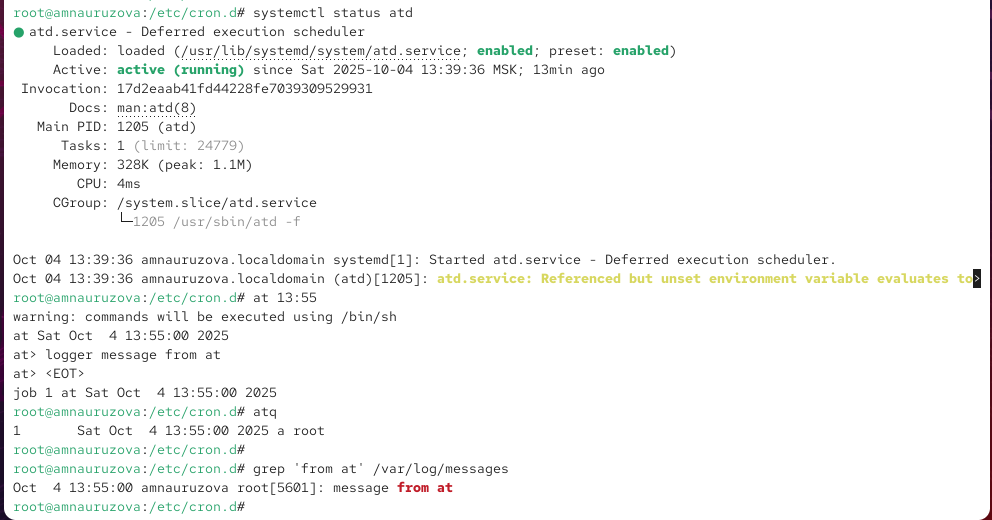


Рис. 7: Проверка службы atd

### 2.2.1 Создание задания at

Далее было запланировано однократное выполнение команды logger message from at в определённое время:

at 13:55  
logger message from at  
Ctrl + D

Просмотр очереди заданий:

atq

Показывает одно запланированное задание. После выполнения проверка через:

grep ‘from at’ /var/log/messages

показывает запись в журнале, подтверждающую выполнение команды.

# 3 Контрольные вопросы

**1. Как настроить задание cron, чтобы оно выполнялось раз в 2 недели?**  
Планировщик cron не поддерживает прямое указание интервала в 2 недели.  
Для этого можно использовать день недели и проверку недели через условие в скрипте, либо запускать раз в 14 дней через anacron.  
Пример с использованием условия:  
0 0 \* \* 0 [ $(($(date +%U) % 2)) -eq 0 ] && /path/to/script.sh  
Здесь задание выполняется по воскресеньям, но только на чётных неделях года.

**2. Как настроить задание cron, чтобы оно выполнялось 1-го и 15-го числа каждого месяца в 2 часа ночи?**  
Используется следующая запись:  
0 2 1,15 \* \* /path/to/script.sh  
Это означает: в 2:00, 1-го и 15-го числа каждого месяца.

**3. Как настроить задание cron, чтобы оно выполнялось каждые 2 минуты каждый день?**  
Для этого используется запись:  
\*/2 \* \* \* \* /path/to/script.sh  
Задание будет выполняться каждые две минуты ежедневно.

**4. Как настроить задание cron, чтобы оно выполнялось 19 сентября ежегодно?**  
Запись имеет вид:  
0 0 19 9 \* /path/to/script.sh  
То есть выполнение в полночь 19 сентября каждого года.

**5. Как настроить задание cron, чтобы оно выполнялось каждый четверг сентября ежегодно?**  
Используется следующая запись:  
0 0 \* 9 4 /path/to/script.sh  
Это означает: запускать в полночь по четвергам (день недели 4) только в сентябре (месяц 9).

**6. Какая команда позволяет вам назначить задание cron для пользователя alice? Приведите подтверждающий пример.**  
Задания можно создать для конкретного пользователя с помощью параметра -u:  
crontab -u alice -e  
Эта команда откроет расписание cron для пользователя **alice**, где можно добавить, например:  
0 8 \* \* \* logger "Cron job for 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:  
echo "bob" >> /etc/cron.deny  
После этого пользователь **bob** не сможет использовать crontab -e для создания заданий.

**8. Вам нужно убедиться, что задание выполняется каждый день, даже если сервер во время выполнения временно недоступен. Как это сделать?**  
Для этого используют планировщик **anacron**, который выполняет пропущенные задания после восстановления системы.  
Пример: добавить задачу в /etc/anacrontab  
1 5 script.daily /path/to/script.sh  
Это обеспечит выполнение задания один раз в день, даже если сервер был выключен.

**9. Какая команда позволяет узнать, запланированы ли какие-либо задания на выполнение планировщиком atd?**  
Для просмотра очереди заданий используется команда:  
atq  
Она выведет список всех отложенных заданий, ожидающих выполнения службой **atd**.

# 4 Заключение

В ходе работы были изучены инструменты планирования задач в Linux с использованием служб **cron** и **at**.  
На практике были созданы и протестированы периодические задания с различными параметрами времени, а также разовые задачи, выполняемые по расписанию.  
Изучен синтаксис записей в файлах crontab и каталоге /etc/cron.d, а также способы управления доступом пользователей к планировщику.