Отчёт по лабораторной работе №8

Планировщики событий

Лабси Мохаммед

Содержание

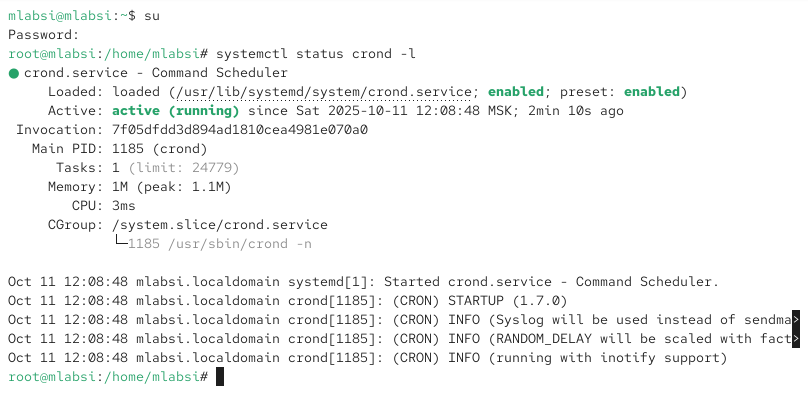
# 1 Цель работы

Получение навыков работы с планировщиками событий cron и at.

# 2 Ход выполнения

## 2.1 Планирование задач с помощью cron

1. Получены права администратора с помощью команды **su -**.  
   После ввода пароля открыт сеанс суперпользователя **root**.
2. Проверен статус службы **crond**, отвечающей за выполнение заданий по расписанию.  
   Команда **systemctl status crond -l** показала, что служба активна (**active (running)**).

* 
* Рис. 1: Проверка статуса службы crond

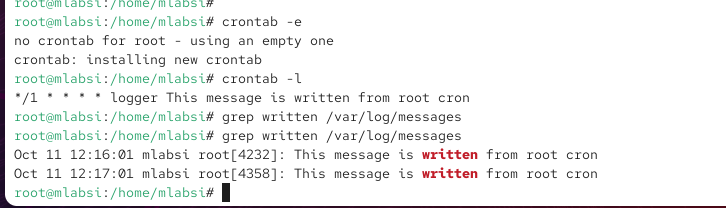
1. Просмотрено содержимое системного файла расписаний **/etc/crontab** с помощью команды **cat /etc/crontab**.  
   В файле заданы переменные среды и пример синтаксиса задания cron.

* 
* Рис. 2: Просмотр содержимого файла /etc/crontab

1. Проверено текущее расписание пользователя **root** командой **crontab -l** — расписание отсутствует.
2. Открыт редактор для создания нового расписания: **crontab -e**.  
   В файл добавлена строка:  
   ***/1*  \* \* \* logger This message is written from root cron**\*  
   Эта запись означает: выполнять команду **logger** каждую минуту, каждый час, каждый день, каждый месяц, независимо от дня недели.  
   Команда **logger** отправляет сообщение в системный журнал.

* 
* Рис. 3: Создание записи в crontab для root

1. После сохранения файла расписание просмотрено повторно (**crontab -l**) — добавленная строка отобразилась корректно.
2. Через несколько минут проверен системный журнал командой **grep written /var/log/messages**.  
   В логе зафиксированы сообщения от **root cron**, что подтверждает успешное выполнение задания.

* 
* Рис. 4: Проверка выполнения задания cron через системный журнал

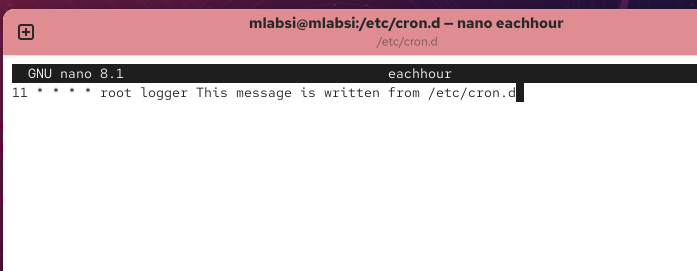
1. Изменено расписание cron на выполнение команды каждый час по будним дням (с понедельника по пятницу):  
   **0 */1*  \* 1-5 logger This message is written from root cron**  
   Синтаксис означает: запускать в начале каждого часа (минуты = 0), каждый час, каждый день месяца, каждый месяц, только по дням недели **1–5 (понедельник–пятница)**.

* 
* Рис. 5: Изменение расписания cron

1. В каталоге **/etc/cron.hourly** создан сценарий **eachhour** с содержимым:  
   **#!/bin/sh**  
   **logger This message is written at $(date)**  
   Этот скрипт будет автоматически выполняться системой каждый час.

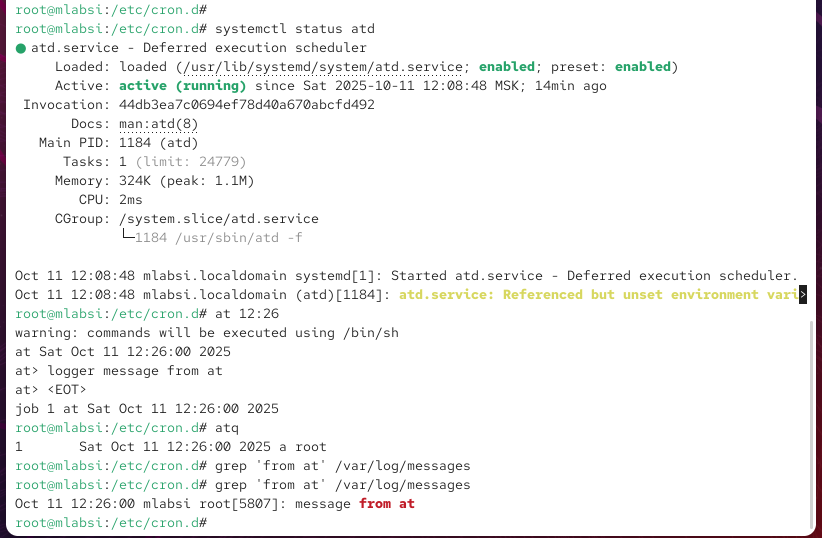
* 
* Рис. 6: Создание скрипта eachhour в /etc/cron.hourly

1. В каталоге **/etc/cron.d** создан файл **eachhour** со следующим содержимым:  
   **11 \* \* \* \* root logger This message is written from /etc/cron.d**  
   Синтаксис: выполнение команды от имени **root** каждый час в **11-й минуте** каждого часа, любого дня, месяца и дня недели.

* 
* Рис. 7: Создание расписания в /etc/cron.d

## 2.2 Планирование заданий с помощью at

1. Проверен статус службы **atd**, управляющей выполнением разовых заданий.  
   Команда **systemctl status atd** показала, что служба активна (**active (running)**).
2. Создано однократное задание с помощью команды **at 12:26**.  
   В интерактивном режиме введена команда **logger message from at**.  
   Завершение ввода выполнено комбинацией **Ctrl + D**.
3. Проверено наличие задания командой **atq** — задание отобразилось в очереди.
4. Через некоторое время проверен системный журнал командой **grep ‘from at’ /var/log/messages**.  
   В логах появилось сообщение, подтверждающее успешное выполнение команды **logger message from at**.

* 
* Рис. 8: Создание и выполнение задания через службу atd

# 3 Контрольные вопросы

1. **Как настроить задание cron, чтобы оно выполнялось раз в 2 недели?**  
   В стандартном синтаксисе cron отсутствует прямой параметр для двухнедельного интервала.  
   Однако можно указать выполнение каждое воскресенье, используя условие по номеру недели:  
   **0 0 \* \* 0 [ (date +%U % 2)) -eq 0 ) ] && команда**  
   Альтернативно можно использовать системный каталог **/etc/cron.d** с более гибким скриптом, который проверяет чётность недели.
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**  
   Синтаксис означает: запуск в 02:00, 1-го и 15-го числа любого месяца, независимо от дня недели.
3. **Как настроить задание cron, чтобы оно выполнялось каждые 2 минуты каждый день?**  
   ***/2*  \* \* \* команда**  
   Синтаксис означает: выполнять команду каждые две минуты, каждый час, каждый день, каждый месяц.
4. **Как настроить задание cron, чтобы оно выполнялось 19 сентября ежегодно?**  
   **0 0 19 9 \* команда**  
   Синтаксис означает: выполнение в 00:00, 19-го числа 9-го месяца (сентября) любого года.
5. **Как настроить задание cron, чтобы оно выполнялось каждый четверг сентября ежегодно?**  
   **0 0 \* 9 4 команда**  
   Синтаксис означает: запуск каждый четверг (день недели 4), в сентябре (месяц 9), в 00:00.
6. **Какая команда позволяет вам назначить задание cron для пользователя alice? Приведите подтверждающий пример.**  
   Команда **crontab -u alice -e** открывает редактор для создания расписания пользователя **alice**.  
   Например:  
   **crontab -u alice -e**  
   и добавляем строку  
   **0 9 \* \* \* logger “Ежедневное сообщение от пользователя alice”**
7. **Как указать, что пользователю bob никогда не разрешено назначать задания через cron? Приведите подтверждающий пример.**  
   В файл **/etc/cron.deny** добавляется строка с именем пользователя:  
   **echo “bob” >> /etc/cron.deny**  
   После этого пользователь **bob** не сможет использовать **crontab** 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 используется утилита **anacron**, предназначенная для выполнения пропущенных заданий.  
   Задания настраиваются в файле **/etc/anacrontab**, где указываются интервалы выполнения и команды.
9. **Какая команда позволяет узнать, запланированы ли какие-либо задания на выполнение планировщиком atd?**  
   Для просмотра очереди заданий используется команда **atq**.  
   Она выводит список всех отложенных заданий с указанием их идентификаторов и времени выполнения.

# 4 Заключение

В ходе работы были изучены и применены инструменты планирования задач в Linux с использованием утилит **cron** и **at**, что позволило автоматизировать выполнение команд по расписанию и в заданное время.