Лабораторная работа №8

Планировщики событий

Комягин Андрей Николаевич

Содержание

# 1 Цель работы

Получение навыков работы с планировщиками событий cron и at.

# 2 Выполнение лабораторной работы

## 2.1 Планирование задач с помощью cron

Посмотриv статус демона crond, посмотрим содержимое файла конфигурации /etc/crontab, осмотрим список заданий в расписании (рис. 1).

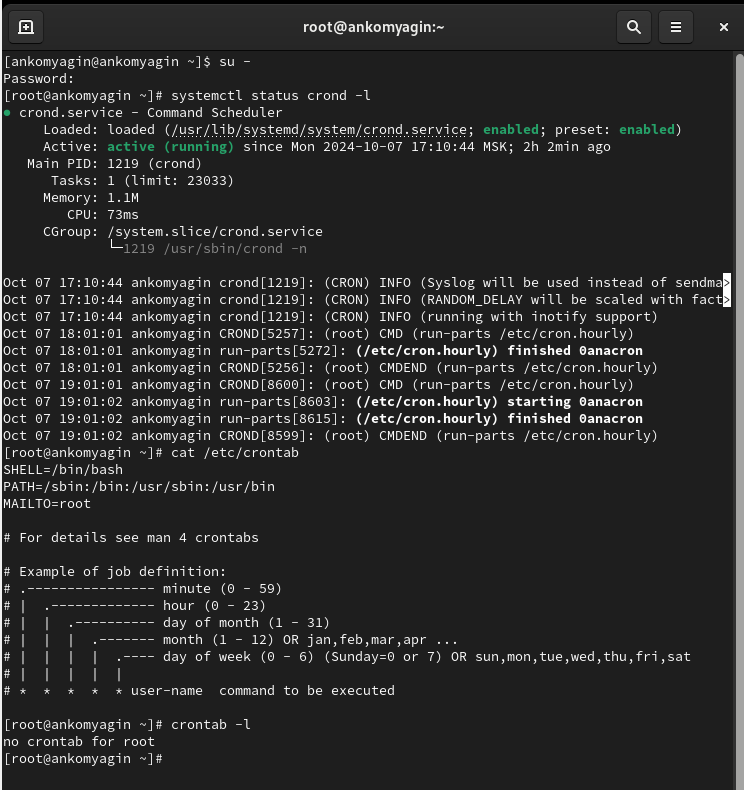


Рис. 1: статус crond задания в расписании

Добавим строку в файл расписания. Посмотрим список заданий в расписании. Просмотрим журнал системных событий(рис. 2).

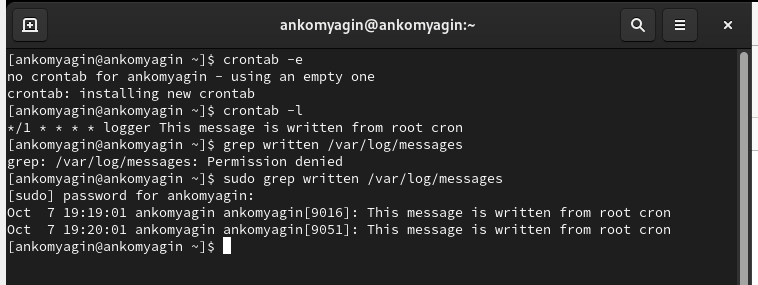


Рис. 2: файл расписания, журнал событий

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

Синтаксис: (по порядку звездочек)

* */1 — это поле минут. Оно означает “каждую минуту”. Знак*  означает “каждое значение”, а /1 указывает, что это происходит каждые 1 минуту.
  + — поле часов. Здесь также указано “каждый час”.
  + — поле дней месяца. “Каждый день месяца”.
  + — поле месяцев. “Каждый месяц”.
  + — поле дней недели. “Каждый день недели”.

Эта запись означает, что команда **logger This message is written from root cron** будет выполняться каждую минуту.

Изменим запись в расписании crontab на следующую:

**0 */1*  \* 1-5 logger This message is written from root cron** (рис. 3).

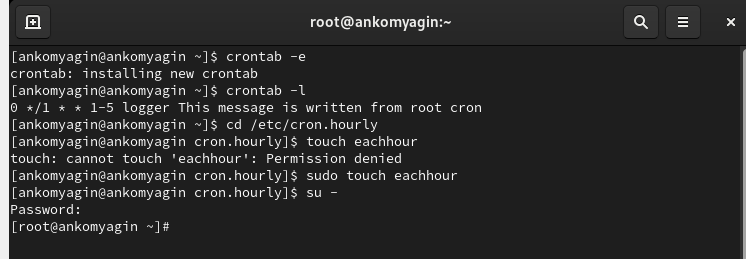


Рис. 3: crontab

Перейдём в каталог /etc/cron.hourly и создадим в нём файл сценария с именем **eachhour**. Откроем файл eachhour и пропишем в нём скрипт. Сделаем файл сценария **eachhour** исполняемым. Перейдём в каталог /etc/crond.d и создадим в нём файл с расписанием **eachhour** (рис. 4).

Синтаксис:

* 0 — это поле минут. Команда будет выполняться в начале часа (0 минут).
* \*/1 — поле часов. Это означает “каждый час”.
  + — поле дней месяца. “Каждый день месяца”.
  + — поле месяцев. “Каждый месяц”.
* 1-5 — поле дней недели. Это означает, что команда будет выполняться только с понедельника по пятницу.

Эта запись означает, что команда будет выполняться каждый час в начале часа (0 минут) только с понедельника по пятницу.

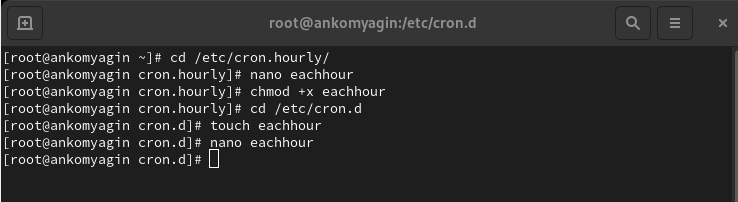


Рис. 4: eachhour

Откройем этот файл для редактирования и поместим в него следующее содержимое:

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

Синтаксис:

* 11 — это поле минут. Команда будет выполняться в 11-й минуте каждого часа.
  + — поле часов. “Каждый час”.
  + — поле дней месяца. “Каждый день месяца”.
  + — поле месяцев. “Каждый месяц”.
  + — поле дней недели. “Каждый день недели”.
* root — это поле пользователя, указывающее, что команда будет выполняться от имени пользователя root.

Эта запись означает, что команда **logger This message is written from /etc/cron.d** будет выполняться каждый час в 11-й минуте от имени пользователя root.

## 2.2 Планирование заданий с помощью at

Проверим, что служба atd загружена и включена. Зададим выполнение команды logger message from at в 19:29. Убедимся, что задание действительно запланировано(рис. 5).

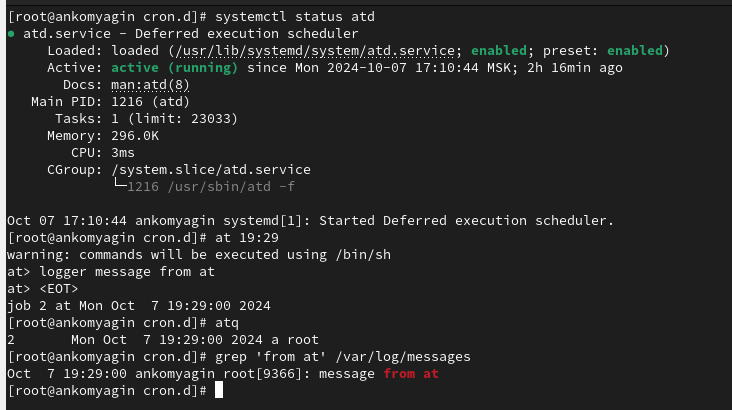


Рис. 5: служба atd

# 3 Контрольные вопросы

1. Настройка задания cron для выполнения раз в 2 недели

Чтобы задать выполнение задания раз в 2 недели, можно использовать следующие записи:

**0 0 \* \* 0,14 command**

Эта запись будет выполнять команду в полночь (00:00) каждое воскресенье и 14-го числа каждого месяца. Однако, если вам нужно строгое выполнение каждые 14 дней, это потребует более сложной логики, так как cron не поддерживает это напрямую.

1. Настройка задания cron для выполнения 1-го и 15-го числа каждого месяца в 2 часа ночи

**0 2 1,15 \* \* command**

Эта запись будет выполнять команду в 2:00 ночи 1-го и 15-го числа каждого месяца.

1. Настройка задания cron для выполнения каждые 2 минуты каждый день

**\*/2 \* \* \* \* command**

Эта запись будет выполнять команду каждые 2 минуты каждый день.

1. Настройка задания cron для выполнения 19 сентября ежегодно

**0 0 19 9 \* command**

Эта запись будет выполнять команду в полночь (00:00) 19 сентября каждого года.

1. Настройка задания cron для выполнения каждый четверг сентября ежегодно

**0 0 \* 9 4 command**

Эта запись будет выполнять команду в полночь (00:00) каждую пятницу (4 - четверг) в сентябре.

1. Команда для назначения задания cron для пользователя alice Вы можете использовать команду crontab с параметром -u:

**sudo crontab -u alice -e**

Это откроет crontab для пользователя alice, где вы можете добавить свои задания.

1. Запрет назначения заданий через cron для пользователя bob

Для этого можно отредактировать файл /etc/cron.d/cron.allow и /etc/cron.d/cron.deny. Если вы добавите пользователя bob в файл cron.deny, он не сможет назначать задания через cron.

**echo “bob” >> /etc/cron.d/cron.deny**

1. Убедиться, что задание выполняется каждый день, даже если сервер временно недоступен

Вы можете использовать systemd или другой менеджер задач, который поддерживает повторные попытки выполнения. Кроме того, можно добавить логику в сам скрипт, чтобы проверять, был ли он выполнен в предыдущие дни, например, записывать дату последнего выполнения в файл и проверять его наличие.

1. Команда для проверки запланированных заданий на выполнение с помощью atd

**atq**

Эта команда покажет список всех заданий, запланированных с помощью at.

# 4 Вывод

В ходе выполнения лабораторной работы я получил навыки работы с планировщиками событий cron и at.

# Список литературы

[Туис, курс Администрирование операционных систем](https://esystem.rudn.ru/course/view.php?id=5946)