

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №8**

дисциплина: Основы администрирования операционных систем

Студент: Накова Амина Михайловна

Студ. билет № 1132232887

Группа: НПИбд-02-23

**МОСКВА**

2025 г.

## Цель работы:

Целью данной работы является получение навыков работы с планировщиками событий cron и at.

## Планирование задач с помощью cron:

### Мониторинг журнала системных событий в реальном времени:

Запустим терминал и получим полномочия администратора: **su -**.  
Просмотрим статус демона crond: **systemctl status crond -l** и содержимое файла конфигурации **/etc/crontab**: **cat /etc/crontab** (Рис. 1.1):

```
[nakova@localhost ~]$ su nakova
Пароль:
[nakova@localhost ~]$ systemctl status crond -l
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; preset: en>
   Active: active (running) since Mon 2025-02-17 03:17:34 MSK; 7min ago
     Main PID: 919 (crond)
       Tasks: 1 (limit: 23024)
      Memory: 1.0M
         CPU: 41ms
        CGroup: /system.slice/crond.service
                └─919 /usr/sbin/crond -n

фев 17 03:17:34 localhost.localdomain systemd[1]: Started Command Scheduler.
фев 17 03:17:34 localhost.localdomain crond[919]: (CRON) STARTUP (1.5.7)
фев 17 03:17:34 localhost.localdomain crond[919]: (CRON) INFO (Syslog will be u>
фев 17 03:17:34 localhost.localdomain crond[919]: (CRON) INFO (RANDOM_DELAY wil>
фев 17 03:17:34 localhost.localdomain crond[919]: (CRON) INFO (running with ino>
lines 1-15/15 (END)
```

**Рис. 1.1.** Запуск терминала и получение полномочий администратора, просмотр статуса демона crond и содержимого файла конфигурации **/etc/crontab**.

Теперь посмотрим список заданий в расписании: **crontab -l**. Ничего не отобразилось, так как расписание ещё не задано. Далее откроем файл расписания на редактирование: **crontab -e** (Рис. 1.2):

```

фев 17 03:17:34 localhost.localdomain crond[919]: (CRON) INFO (Syslog will be u>
фев 17 03:17:34 localhost.localdomain crond[919]: (CRON) INFO (RANDOM_DELAY wil>
фев 17 03:17:34 localhost.localdomain crond[919]: (CRON) INFO (running with ino>
[nakova@localhost ~]$ cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

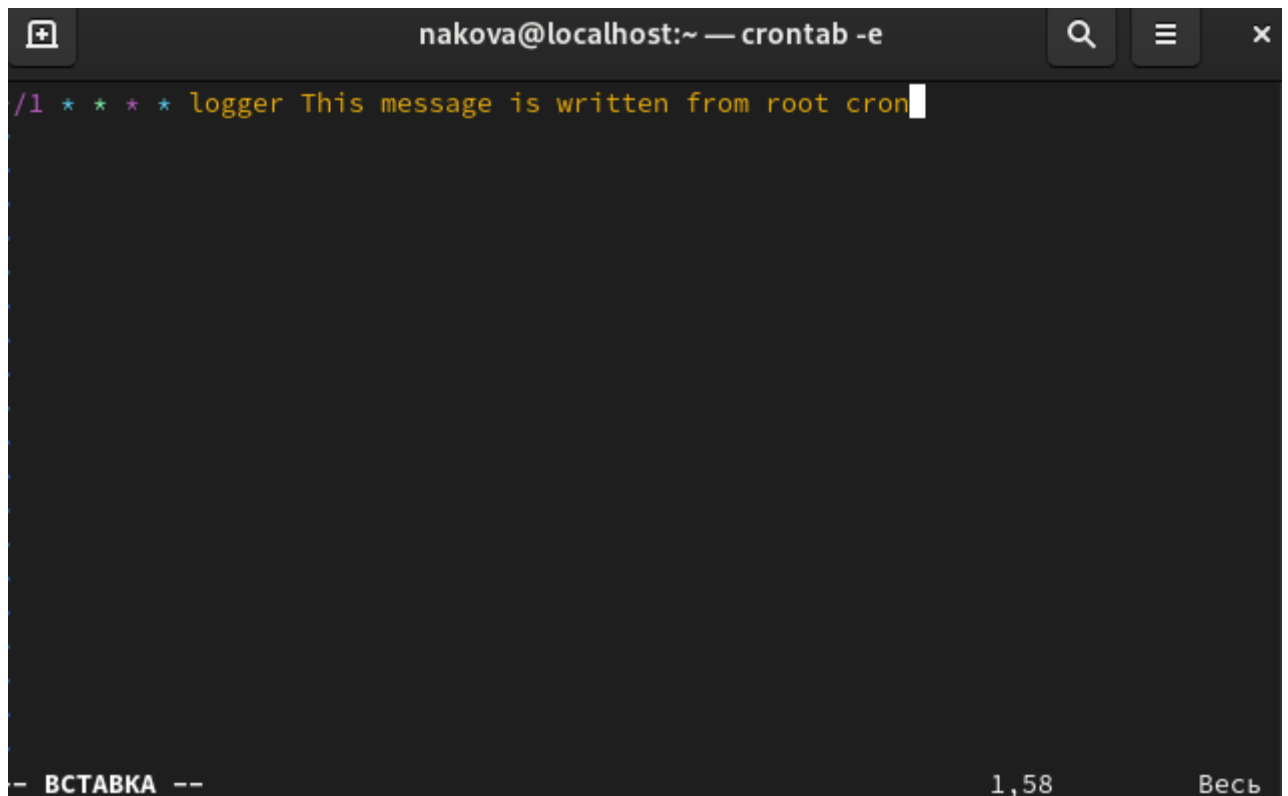
# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,f
ri,sat
# | | | | |
# * * * * * user-name  command to be executed

[nakova@localhost ~]$

```

**Рис. 1.2.** Просмотр списка заданий в расписании, открытие файла расписания на редактирование.

Предыдущая команда запустила интерфейс редактора (по умолчанию используется vi). Добавим следующую строку в файл расписания (запись сообщения в системный журнал), используя **Ins** для перехода в vi в режим ввода: **\*/1 \* \* \* \* logger This message is written from root cron.** Закроем сеанс редактирования vi и сохраним изменения, используя команду vi: **Esc : wq**. (Рис. 1.3):



**Рис. 1.3.** Открытие текстового редактора vi и добавление строки в файл расписания.

Просмотрим список заданий в расписании: **crontab -l** (в расписании появилась запись о запланированном событии). Не выключая систему, через некоторое время (2–3 минуты) посмотрим журнал системных событий: **grep written /var/log/messages** (Рис. 1.4):

```
crontab: installing new crontab
[nakova@localhost ~]$ crontab -l
*/1 * * * * logger This message is written from root cron
[nakova@localhost ~]$
```

**Рис. 1.4.** Просмотр списка заданий в расписании, просмотр журнала системных событий.

Вернёмся в текстовый редактор vi и изменим запись в расписании crontab на следующую: **0 \*/1 \* \* 1-5 logger This message is written from root cron** (Рис. 1.5):

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

**Рис. 1.5.** Изменение записи в расписании crontab.

Теперь посмотрим список заданий в расписании: **crontab -l** (Рис. 1.6).

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

**Рис. 1.6.** Просмотр списка заданий в расписании.

Перейдём в каталог `/etc/cron.hourly` и создадим в нём файл сценария с именем `eachhour` (Рис. 1.7):

**`cd /etc/cron.hourly`**

**`touch eachhour`**

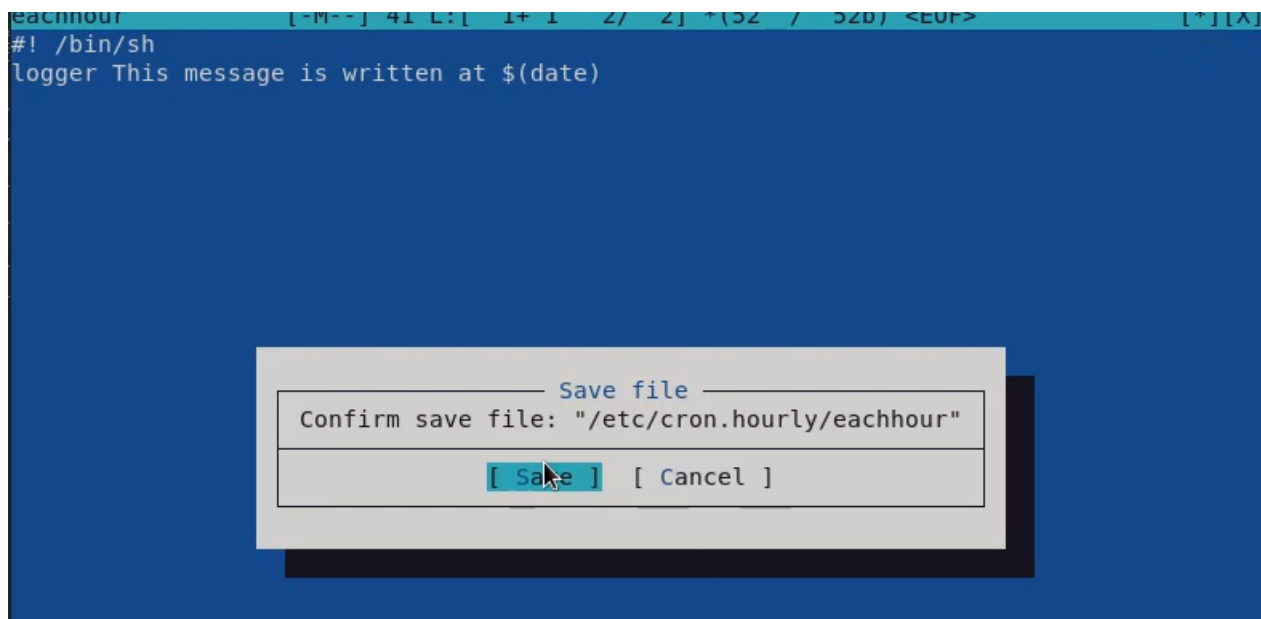
Далее откроем файл `eachhour` для редактирования и пропишем в нём следующий скрипт (запись сообщения в системный журнал) (Рис. 1.8):

**`#!/bin/sh`**

**`logger This message is written at $(date)`**

```
touch eachhour  
mcedit eachhour
```

**Рис. 1.7.** Открытие каталога `/etc/cron.hourly` и создание в нём файла сценария с именем `eachhour`.



**Рис. 1.8.** Открытие файла `eachhour` для редактирования и прописывание в нём скрипта.

Сделаем файл сценария `eachhour` исполняемым:

**`chmod +x eachhour`**

Теперь перейдём в каталог `/etc/crond.d` и создадим в нём файл с расписанием `eachhour` (Рис. 1.9):

**`cd /etc/cron.d`**

**`touch eachhour`**

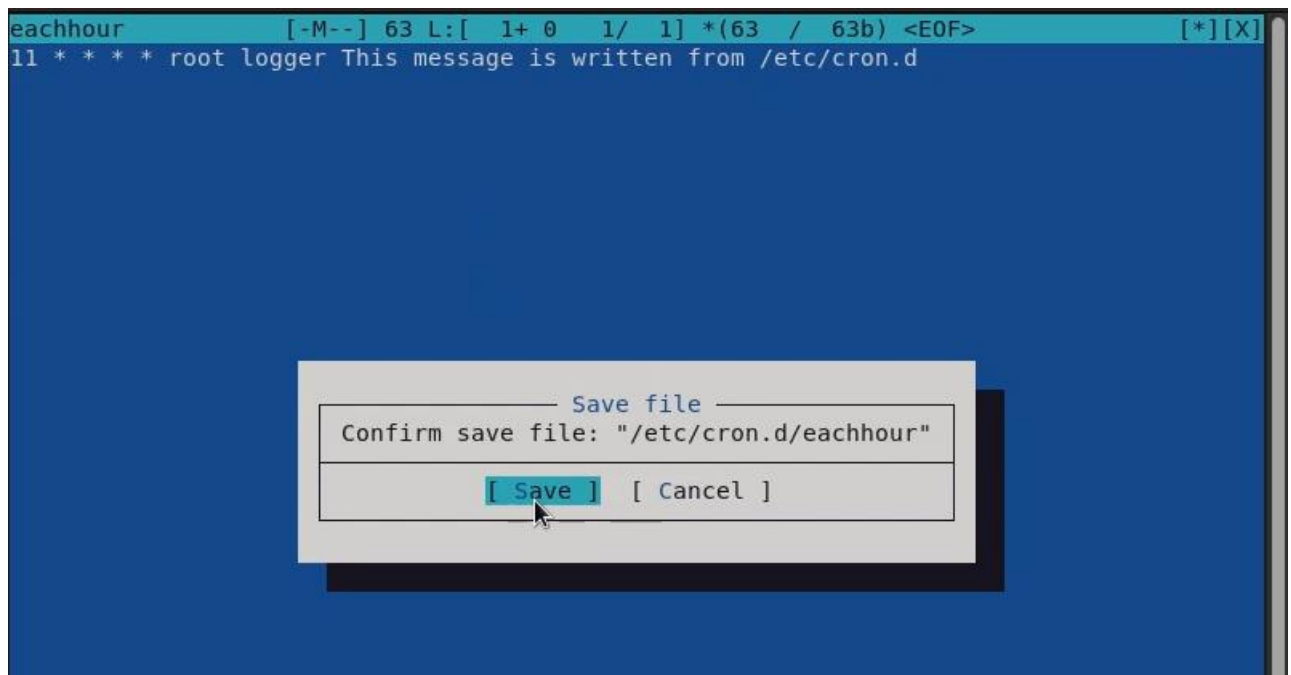
Откроем этот файл для редактирования и поместим в него следующее содержимое:

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

Сохраним изменения (Рис. 1.10).

```
chmod +x eachhour  
cd /etc/cron.d
```

**Рис. 1.9.** Делаем файл сценария `eachhour` исполняемым, открытие каталога `/etc/cron.d` и создание в нём файла с расписанием `eachhour`, открытие файла на редактирование.



**Рис. 1.10.** Добавление содержимого в файл и сохранение изменений.

Не выключая систему, через некоторое время (2 часа) посмотрим журнал системных событий:

```
grep written /var/log/messages
```

По журналу видим, что запуск сценария `eachhour` в соответствии с заданным расписанием был осуществлён

## Планирование заданий с помощью at:

Запустим терминал и получим полномочия администратора: **su -**. Проверим, что служба atd загружена и включена: **systemctl status atd**. Зададим выполнение команды **logger message from at** в 12:50. Для этого введём **at 12:50**, а затем **logger message from at**. Используем **Ctrl + d**, чтобы закрыть оболочку. Убедимся, что задание действительно запланировано: **atq**. С помощью команды **grep from at /var/log/messages** посмотрим, появилось ли соответствующее сообщение в лог-файле в указанное вами время (сообщение появилось)

## Ответы на контрольные вопросы:

1. Как настроить задание cron, чтобы оно выполнялось раз в 2 недели? **00 00 1,15 \* \* logger task**
2. Как настроить задание cron, чтобы оно выполнялось 1-го и 15-го числа каждого месяца в 2 часа ночи? **00 02 1,15 \* \* logger task**
3. Как настроить задание cron, чтобы оно выполнялось каждые 2 минуты каждый день? **\*/2 \* \* \* \* logger task**
4. Как настроить задание cron, чтобы оно выполнялось 19 сентября ежегодно? **\* \* 19 9 logger task**
5. Как настроить задание cron, чтобы оно выполнялось каждый четверг сентября ежегодно? **\* \* \* \* 4 logger task**
6. Какая команда позволяет вам назначить задание cron для пользователя alice? Приведите подтверждающий пример. **\* \* \* \* alice logger task**
7. Как указать, что пользователю bob никогда не разрешено назначать задания через cron? Приведите подтверждающий пример. **записать его в /etc/cron.deny**



8. Вам нужно убедиться, что задание выполняется каждый день, даже если сервер во время выполнения временно недоступен. Как это сделать? **Найти задание в логах `grep cron /var/log/messages`**

9. Какая команда позволяет узнать, запланированы ли какие-либо задания на выполнение планировщиком atd? **`atq`**

### **Вывод:**

В ходе выполнения лабораторной работы были получены навыки работы с планировщиками событий cron и at.