

# Лабораторная работа №8

## Управление событиями

Перфилов Александр Константинович | Группа НПИбд-03-24

Российский университет дружбы народов, Москва, Россия

24 октября 2025

## Раздел 1

### Информация

# Докладчик

- Перфилов Александр Константинович
- Группа НПИбд-03-24
- Российский университет дружбы народов
- <https://github.com/AlexanderPErfilovKonstantinivich?tab=repositories>

## Цель работы

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

# Задание

- 1 Выполните задания по планированию задач с помощью `crond` (см. раздел 8.4.1).
- 2 Выполните задания по планированию задач с помощью `atd` (см. раздел 8.4.2).

## Раздел 2

### Выполнение лабораторной работы

## Планирование задач с помощью cron

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).

# Планирование задач с помощью cron

```
root@akpperfilov:~# systemctl status crond -l
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-10-24 15:22:52 MSK; 21min ago
  Invocation: 18d9bc5eb373415e862554a3032cb799
    Main PID: 1033 (crond)
      Tasks: 1 (limit: 48908)
     Memory: 1M (peak: 1.1M)
        CPU: 15ms
    CGroup: /system.slice/crond.service
            └─1033 /usr/sbin/crond -n

окт 24 15:22:52 akpperfilov.localdomain systemd[1]: Started crond.service - Command Sched
окт 24 15:22:52 akpperfilov.localdomain crond[1033]: (CRON) STARTUP (1.7.0)
окт 24 15:22:52 akpperfilov.localdomain crond[1033]: (CRON) INFO (Syslog will be used ins
окт 24 15:22:52 akpperfilov.localdomain crond[1033]: (CRON) INFO (RANDOM_DELAY will be sc
окт 24 15:22:52 akpperfilov.localdomain crond[1033]: (CRON) INFO (running with inotify su
root@akpperfilov:~# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

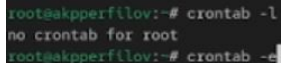
# For details see man 4 crontabs

# 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,fri,sat
```



## Планирование задач с помощью cron

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. 2).



```
root@akpperfilov:~# crontab -l
no crontab for root
root@akpperfilov:~# crontab -e
```

Рис. 2: Просмотр списка заданий в расписании, открытие файла расписания на редактирование.

## Планирование задач с помощью cron

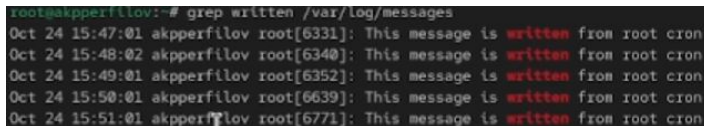
Предыдущая команда запустила интерфейс редактора (по умолчанию используется vi). Добавим следующую строку в файл расписания (запись сообщения в системный журнал), используя Ins для перехода в vi в режим ввода: `/1 * * * logger This message is written from root cron`. Закроем сеанс редактирования vi и сохраним изменения, используя команду vi: `Esc : wq`. (рис. 3).



Рис. 3: Открытие текстового редактора vi и добавление строки в файл расписания.

## Планирование задач с помощью cron

Посмотрим 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. 4).

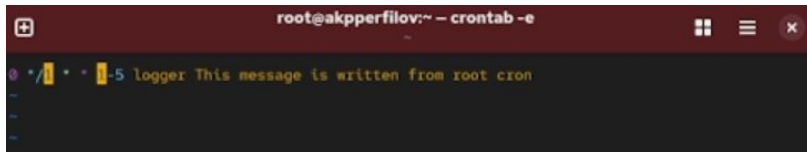


```
root@akpperfilov:~# grep written /var/log/messages
Oct 24 15:47:01 akpperfilov root[6331]: This message is written from root cron
Oct 24 15:48:02 akpperfilov root[6340]: This message is written from root cron
Oct 24 15:49:01 akpperfilov root[6352]: This message is written from root cron
Oct 24 15:50:01 akpperfilov root[6639]: This message is written from root cron
Oct 24 15:51:01 akpperfilov root[6771]: This message is written from root cron
```

Рис. 4: Просмотр списка заданий в расписании, просмотр журнала системных событий.

## Планирование задач с помощью cron

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. 5).



```
root@akpperfilov:~ - crontab -e
* /1 * * 1-5 logger This message is written from root cron
```

Рис. 5: Изменение записи в расписании crontab.

## Планирование задач с помощью cron

Теперь посмотрим список заданий в расписании: `crontab -l` (рис. 6).

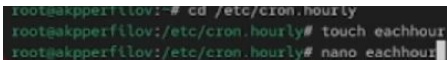


```
root@akpperfilov:~# crontab -l
0 */1 * * 1-5 logger This message is written from root cron
```

Рис. 6: Просмотр списка заданий в расписании.

## Планирование задач с помощью cron

Перейдём в каталог `/etc/cron.hourly` и создадим в нём файл сценария с именем `eachhour` (рис. 7).

A screenshot of a terminal window showing three commands being executed in sequence. The first command is `cd /etc/cron.hourly`, the second is `touch eachhour`, and the third is `nano eachhour`. The prompt is `root@akpperfilov:~#` for the first command and `root@akpperfilov:/etc/cron.hourly#` for the subsequent two.

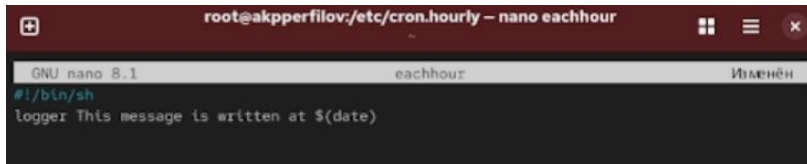
```
root@akpperfilov:~# cd /etc/cron.hourly
root@akpperfilov:/etc/cron.hourly# touch eachhour
root@akpperfilov:/etc/cron.hourly# nano eachhour
```

Рис. 7: Открытие каталога `/etc/cron.hourly` и создание в нём файла сценария с именем `eachhour`.

## Планирование задач с помощью cron

Далее откроем файл `eachhour` для редактирования и пропишем в нём следующий скрипт (запись сообщения в системный журнал) (рис. 8).

# Планирование задач с помощью cron



```
root@akpperfilov:/etc/cron.hourly - nano eachhour
GNU nano 8.1      eachhour      Изменён
#!/bin/sh
logger This message is written at $(date)
```

Рис. 8: Открытие файла `eachhour` для редактирования и прописывание в нём скрипта.



## Планирование задач с помощью cron

Сделаем файл сценария `eachhour` исполняемым. Теперь перейдём в каталог `/etc/cron.d` и создадим в нём файл с расписанием `eachhour` (рис. 9).

```
root@akpperfilov:/etc/cron.hourly# chmod +x eachhour
root@akpperfilov:/etc/cron.hourly# cd /etc/cron.d
root@akpperfilov:/etc/cron.d# touch eachhour
```

Рис. 9: Делаем файл сценария `eachhour` исполняемым, открытие каталога `/etc/cron.d` и создание в нём файла с расписанием `eachhour`, открытие файла на редактирование.

## Планирование задач с помощью cron

Откроем этот файл для редактирования и поместим в него следующее содержимое (рис. 10).

# Планирование задач с помощью cron

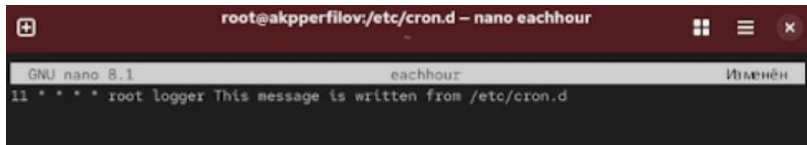


Рис. 10: Добавление содержимого в файл и сохранение изменений.

## Планирование задач с помощью cron

Не выключая систему, через некоторое время (2 часа) посмотрим журнал системных событий. По журналу видим, что запуск сценария `eachhour` в соответствии с заданным расписанием был осуществлён (рис. 11).

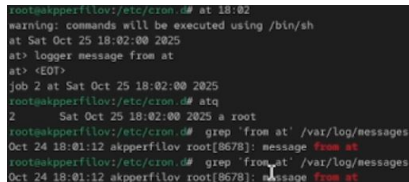
```
root@akpperfilov:/etc/cron.d# grep written /var/log/messages
Oct 24 15:47:01 akpperfilov root[6331]: This message is written from root cron
Oct 24 15:48:02 akpperfilov root[6340]: This message is written from root cron
Oct 24 15:49:01 akpperfilov root[6352]: This message is written from root cron
Oct 24 15:50:01 akpperfilov root[6639]: This message is written from root cron
Oct 24 15:51:01 akpperfilov root[6771]: This message is written from root cron
Oct 24 15:52:01 akpperfilov root[7007]: This message is written from root cron
Oct 24 16:00:01 akpperfilov root[7612]: This message is written from root cron
Oct 24 16:01:01 akpperfilov root[7645]: This message is written at Пн 24 окт 2025 16:01:01
MSK
Oct 24 16:11:01 akpperfilov root[7703]: This message is written from /etc/cron.d
Oct 24 17:00:01 akpperfilov root[7977]: This message is written from root cron
Oct 24 17:01:01 akpperfilov root[8000]: This message is written at Пн 24 окт 2025 17:01:01
MSK
Oct 24 17:11:02 akpperfilov root[8041]: This message is written from /etc/cron.d
```

Рис. 11: Просмотр журнала системных событий через 2 часа.

## 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 17:00. Для этого введём `at 17:00`, 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) (рис. 12).

# Планирование заданий с помощью at



```
root@akpperfilov:/etc/cron.d# at 18:02
warning: commands will be executed using /bin/sh
at Sat Oct 25 18:02:00 2025
at> logger message from at
at> <EOT>
job 2 at Sat Oct 25 18:02:00 2025
root@akpperfilov:/etc/cron.d# atq
2          Sat Oct 25 18:02:00 2025 a root
root@akpperfilov:/etc/cron.d# grep 'from at' /var/log/messages
Oct 24 18:01:12 akpperfilov root[8678]: message from at
root@akpperfilov:/etc/cron.d# grep 'from at' /var/log/messages
Oct 24 18:01:12 akpperfilov root[8678]: message from at
```

Рис. 12: Запуск терминала и получение полномочий администратора, проверка загрузки и включения службы atd, создание задачи выполнения команды logger message from at в 17:00, закрытие оболочки. Проверка планировки задачи и выполнения её в указанное время.

## Выводы

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