



## РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук  
Кафедра прикладной информатики и теории вероятностей

### ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 8

Планировщики событий (cron и at)

Студент: Эйвази Мани

Группа: НПИбд-03-24

Студенческий билет №: 1032245107

### **Цель работы**

Получение практических навыков работы с планировщиками событий cron и at. Освоение методов автоматизации регулярных и одноразовых задач в операционной системе Linux, включая настройку расписаний, создание скриптов и управление заданиями..

**Первый шаг:** Проверка состояния демона `crond` и изучение системного файла расписаний. Служба `crond` активна и запущена. Файл `/etc/crontab` содержит системное расписание и переменные окружения. Личное расписание пользователя `root` изначально пустое.

```
Jan 30 3:37 PM
manieyvazi@localhost:/etc/cron.d
/etc/cron.d

manieyvazi@localhost:~$ su
Password:
root@localhost:/home/manieyvazi# systemctl status crond -l
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; preset: enabled)
   Active: active (running) since Fri 2026-01-30 15:01:58 MSK; 12min ago
  Invocation: e395ae59c31f4a979ea71fae86423829
    Main PID: 1320 (crond)
      Tasks: 1 (limit: 48916)
     Memory: 1M (peak: 1.2M)
        CPU: 71ms
    CGroup: /system.slice/crond.service
            └─1320 /usr/sbin/crond -n

Jan 30 15:01:58 localhost.localdomain systemd[1]: Started crond.service - Command Scheduler.
Jan 30 15:01:58 localhost.localdomain crond[1320]: (CRON) STARTUP (1.7.0)
Jan 30 15:01:58 localhost.localdomain crond[1320]: (CRON) INFO (Syslog will be used instead of sendmail.)
Jan 30 15:01:58 localhost.localdomain crond[1320]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 56% if used.)
Jan 30 15:01:58 localhost.localdomain crond[1320]: (CRON) INFO (running with inotify support)
root@localhost:/home/manieyvazi# 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
# | | | | |
# * * * * * user-name  command to be executed
```

**Второй шаг:** Добавление задания, которое будет выполняться каждую минуту и записывать сообщение в системный журнал. После сохранения файла и проверки командой `crontab -l` задание отображается в списке. Через несколько минут команда `grep "written from root cron" /var/log/messages` подтверждает его выполнение – в логе появляются соответствующие записи.

```
root@localhost:/home/manieyvazi# crontab -e
no crontab for root - using an empty one
crontab: installing new crontab
"/tmp/crontab.fyRK1y":1: bad minute
Invalid crontab file, can't install.
Do you want to retry the same edit? (Y/N) y
crontab: installing new crontab
```

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

**Третий шаг:** Изменение задания для выполнения в 0 минут каждого часа, но только по рабочим дням (понедельник-пятница). После редактирования (`crontab -e`) и проверки (`crontab -l`) задание обновлено. Логи будут писаться только в указанные часы рабочих дней.

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

**Четвертый шаг:** Создание исполняемого скрипта, который будет запускаться ежечно. Создан скрипт `eachhour`. Демон `crond` автоматически выполняет все исполняемые скрипты из каталогов `cron.hourly`, `cron.daily` и т.д. в соответствии со своим расписанием (обычно каждый час, день и т.д.).

```
root@localhost:/home/manieyvazi# cd /etc/cron.hourly
root@localhost:/etc/cron.hourly# touch eachhour
root@localhost:/etc/cron.hourly# nano eachhour
root@localhost:/etc/cron.hourly# cat eachhour
#!/bin/sh
logger This message is written at $(date)
root@localhost:/etc/cron.hourly# chmod +x eachhour
```

**Пятый шаг:** Создание системного файла расписания для выполнения задачи в конкретную минуту каждого часа. Создан файл `eachhour`. Его синтаксис отличается от пользовательского `crontab`: после времени выполнения явно указывается пользователь (`root`), от имени которого будет запущена команда. Задание будет выполняться на 11-й минуте каждого часа. Через некоторое время запись в `/var/log/messages` подтверждает выполнение.

```
root@localhost:/etc/cron.d# cd /etc/cron.d
root@localhost:/etc/cron.d# touch eachhour
root@localhost:/etc/cron.d# ls
0hourly  eachhour
root@localhost:/etc/cron.d# nano eachhour
root@localhost:/etc/cron.d# cat eachhour
11 * * * * root logger This message is written from /etc/cron.d
```

- **Шестой шаг:** Убедиться, что демон для одноразовых заданий запущен.

Команда: `systemctl status atd`

Служба `atd` активна и работает.

**Седьмой шаг:** Запланировать выполнение команды на конкретное время (например, через 2 минуты от текущего). Команда `atq` показывает запланированное задание с его номером (job id) и временем выполнения. В указанное время (`grep 'message from at' /var/log/messages`) в системном журнале появляется соответствующая запись, подтверждающая успешное выполнение.

```
[1]+  Stopped                  systemctl status atd
root@localhost:/home/manieyvazi# at 9:30
warning: commands will be executed using /bin/sh
at Sat Jan 31 09:30:00 2026
at> logger message from at
at> <EOT>
job 1 at Sat Jan 31 09:30:00 2026
root@localhost:/home/manieyvazi# atq
1          Sat Jan 31 09:30:00 2026 a root
root@localhost:/home/manieyvazi# at 15:32
warning: commands will be executed using /bin/sh
at Fri Jan 30 15:32:00 2026
at> logger message test
at> <EOT>
job 2 at Fri Jan 30 15:32:00 2026
root@localhost:/home/manieyvazi# atq
1          Sat Jan 31 09:30:00 2026 a root
2          Fri Jan 30 15:32:00 2026 a root
root@localhost:/home/manieyvazi#
```

## заключение

- **Личные задания:** создание через crontab-e, освоен синтаксис полей времени и шаблоны (\*, \*/n, диапазоны).
  - **Системные задания через каталоги:** использование /etc/cron.hourly/, /etc/cron.daily/ и др. для исполняемых скриптов с фиксированной периодичностью.
- **Системные задания через /etc/cron.d/:** централизованное управление заданиями с указанием пользователя для выполнения.

### Планировщик at (разовые задачи):

- Планирование команд на указанное время командой at <время>.
- Управление: просмотр очереди (atq), удаление заданий (atrm <номер>).

### Верификация выполнения:

- Проверка факта выполнения через анализ системного журнала /var/log/messages (с использованием grep).
- Генерация тестовых записей с помощью утилиты logger.

**Вывод:** Грамотное применение cron и at позволяет автоматизировать рутинные административные задачи (резервное копирование, очистка, отчёты), повышая эффективность управления системой.

1. Как настроить задание cron, чтобы оно выполнялось раз в 2 недели?

Один из способов: `0 0 */14 * * <команда>`. Это задание будет выполняться в 0:00 каждые 14 дней месяца. Для точного учета недель можно использовать поле "день недели" в комбинации с другими, но проще создать скрипт, который проверяет дату, и запускать его ежедневно.

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>`

3. Как настроить задание cron, чтобы оно выполнялось каждые 2 минуты каждый день?

`*/2 * * * * <команда>`

4. Как настроить задание cron, чтобы оно выполнялось 19 сентября ежегодно?

`0 0 19 9 * <команда>`

5. Как настроить задание cron, чтобы оно выполнялось каждый четверг сентября ежегодно?

`0 0 * 9 4 <команда>` (поле "день недели": 4 = четверг, поле "день месяца": \* = любой).

6. Какая команда позволяет вам назначить задание cron для пользователя alice? Приведите подтверждающий пример.

`crontab -u alice -e` (для редактирования). Или можно создать файл и загрузить его: `crontab -u alice <файл_с_расписанием>`.

7. Как указать, что пользователю bob никогда не разрешено назначать задания через cron? Приведите подтверждающий пример.

Добавить имя пользователя в файл `/etc/cron.deny`. Пример:

`bash`

`echo "bob" >> /etc/cron.deny`



8. Вам нужно убедиться, что задание выполняется каждый день, даже если сервер во время выполнения временно недоступен. Как это сделать?

Использовать anacron. Задания anacron (обычно размещаются в /etc/cron.daily/, /etc/cron.weekly/, /etc/cron.monthly/) выполняются при следующем включении системы, если они были пропущены по расписанию.

9. Какая команда позволяет пользователю посмотреть, какие задания назначены на выполнение планировщиком atd?

atq (или at-l).