

Отчёт по лабораторной работе №8

Планировщики событий

Анна Саенко

Содержание

1	Цель работы	5
2	Ход выполнения работы	6
2.1	Планирование задач с помощью stop	6
2.2	Планирование заданий с помощью at	10
3	Контрольные вопросы	12
4	Заключение	15

Список иллюстраций

2.1	Просмотр файла /etc/crontab	7
2.2	Создание задания cron для root	8
2.3	Проверка выполнения cron-задания через журнал	8
2.4	Изменение расписания cron для root	9
2.5	Создание сценария eachhour	9
2.6	Создание cron-задания в /etc/cron.d	10
2.7	Проверка статуса службы atd	11

Список таблиц

1 Цель работы

Получение навыков работы с планировщиками событий cron и at.

2 Ход выполнения работы

2.1 Планирование задач с помощью cron

Сначала я проверила состояние службы `crond`, отвечающей за выполнение заданий по расписанию. Из вывода команды видно, что служба `crond` загружена (loaded), включена (enabled) и активно работает (active running).

Далее я просмотрела содержимое системного файла расписания `/etc/crontab`, который используется для определения глобальных задач, выполняемых по расписанию.

В файле указаны переменные среды, такие как `SHELL`, `PATH` и `MAILTO`, а также приведён пример формата задания.

```

root@aasaenko:/home/aasaenko# systemctl status crond -l
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-10-06 13:35:44 MSK; 1min 57s ago
 Invocation: cdf5be42e6c04298ac52bad2f4c1658f
    Main PID: 1207 (crond)
      Tasks: 1 (limit: 24779)
     Memory: 1M (peak: 1.1M)
        CPU: 6ms
    CGroup: /system.slice/crond.service
           └─1207 /usr/sbin/crond -n

Oct 06 13:35:44 aasaenko.localdomain systemd[1]: Started crond.service - Command Scheduler.
Oct 06 13:35:44 aasaenko.localdomain crond[1207]: (CRON) STARTUP (1.7.0)
Oct 06 13:35:44 aasaenko.localdomain crond[1207]: (CRON) INFO (Syslog will be used instead of sendmail.)
Oct 06 13:35:44 aasaenko.localdomain crond[1207]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 30% if used.)
Oct 06 13:35:44 aasaenko.localdomain crond[1207]: (CRON) INFO (running with inotify support)
root@aasaenko:/home/aasaenko# 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

root@aasaenko:/home/aasaenko#

```

Рис. 2.1: Просмотр файла /etc/crontab

Затем я открыла индивидуальное расписание пользователя root с помощью команды редактирования crontab.

В пустой файл добавила строку:

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

Эта запись означает, что команда logger будет выполняться каждую минуту.

Поля строки имеют следующий смысл:

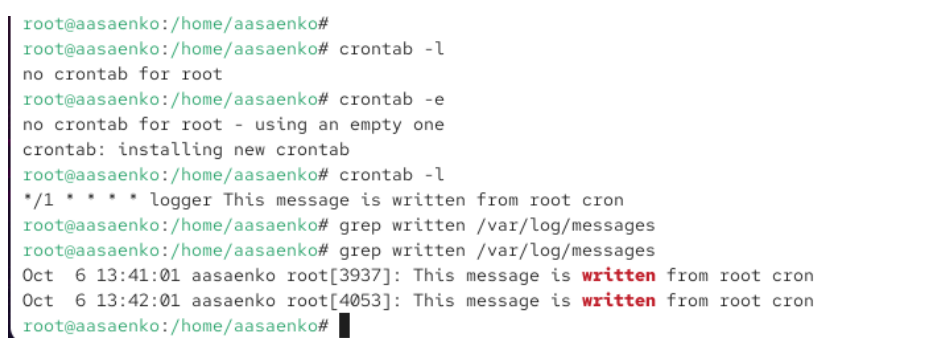
- /1 — интервал в одну минуту,
- — каждый час,
- * — каждый день месяца,
- * — каждый месяц,
- * — каждый день недели,
- далее следует команда, выполняемая по расписанию.

A terminal window titled 'aasaenko@aasaenko:/home/aasaenko - crontab -e'. The first line of the crontab file is '* /1 * * * * logger This message is written from root cron'. The rest of the file is empty, indicated by tilde symbols (~) on subsequent lines.

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

Рис. 2.2: Создание задания cron для root

После сохранения файла я проверила, что запись успешно добавлена. Через несколько минут просмотрела системный журнал /var/log/messages для проверки выполнения задания. В логе появились строки с сообщением «This message is written from root cron», что подтверждает корректную работу cron-задания.

A terminal window showing a series of commands and their outputs to verify the cron job. The commands include 'crontab -l', 'crontab -e', 'crontab -l', and 'grep' searches in the system log. The log entries show the message being written at specific times.

```
root@aasaenko:/home/aasaenko#
root@aasaenko:/home/aasaenko# crontab -l
no crontab for root
root@aasaenko:/home/aasaenko# crontab -e
no crontab for root - using an empty one
crontab: installing new crontab
root@aasaenko:/home/aasaenko# crontab -l
*/1 * * * * logger This message is written from root cron
root@aasaenko:/home/aasaenko# grep written /var/log/messages
root@aasaenko:/home/aasaenko# grep written /var/log/messages
Oct  6 13:41:01 aasaenko root[3937]: This message is written from root cron
Oct  6 13:42:01 aasaenko root[4053]: This message is written from root cron
root@aasaenko:/home/aasaenko#
```

Рис. 2.3: Проверка выполнения cron-задания через журнал

Далее я изменила расписание, чтобы сообщение записывалось в начале каждого часа с понедельника по пятницу.

Для этого отредактировала crontab и добавила строку:

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

Расшифровка записи:

- 0 — выполнение в начале часа,
- /1 — *каждый час*,
- — каждый день месяца,
- * — каждый месяц,
- 1-5 — с понедельника по пятницу,
- команда — logger This message is written from root cron.

```
root@aasaenko:/home/aasaenko#  
root@aasaenko:/home/aasaenko# crontab -e  
crontab: installing new crontab  
Backup of root's previous crontab saved to /root/.cache/crontab/crontab.bak  
root@aasaenko:/home/aasaenko# crontab -l  
0 */1 * * 1-5 logger This message is written from root cron  
root@aasaenko:/home/aasaenko#
```

Рис. 2.4: Изменение расписания cron для root

Затем я создала сценарий, который будет выполняться каждый час.

Для этого в каталоге /etc/cron.hourly был создан файл eachhour.

Этот сценарий записывает текущее время в системный журнал при каждом запуске.

```
aasaenko@aasaenko:/etc/cron.hourly - n  
/etc/cron.hourly  
eachhour [----] 41 L:[ 1+ 1 2/ 2] *(51 / 51b) <EOF>  
#!/bin/sh  
logger This message is written at $(date)
```

Рис. 2.5: Создание сценария eachhour

Далее я создала задание cron в каталоге /etc/cron.d.

Для этого был создан файл eachhour, в который добавлена строка:

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

Эта запись означает, что каждые сутки, в 11-й минуте каждого часа, пользователь root будет выполнять команду logger.

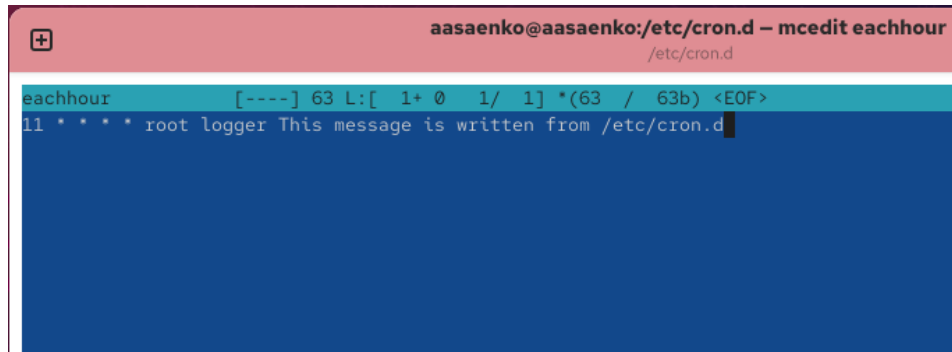


Рис. 2.6: Создание cron-задания в /etc/cron.d

Через некоторое время можно просмотреть журнал системных событий /var/log/messages, чтобы убедиться, что задания выполняются корректно.

В журнале должны появиться записи, подтверждающие запуск скрипта и выполнение команд по расписанию.

2.2 Планирование заданий с помощью at

Сначала я проверила состояние службы atd, отвечающей за выполнение однократных заданий по расписанию.

Из вывода команды видно, что служба atd загружена (loaded), включена (enabled) и активно работает (active running).

```
root@aasaenko:/etc/cron.d#  
root@aasaenko:/etc/cron.d# systemctl status atd  
● atd.service - Deferred execution scheduler  
   Loaded: loaded (/usr/lib/systemd/system/atd.service; enabled; preset: enabled)  
   Active: active (running) since Mon 2025-10-06 13:35:44 MSK; 12min ago  
  Invocation: 487f3feec6864e79bbc6e6296fc52dc8  
     Docs: man:atd(8)  
  Main PID: 1202 (atd)  
    Tasks: 1 (limit: 24779)  
  Memory: 320K (peak: 1.1M)  
     CPU: 3ms  
   CGroup: /system.slice/atd.service  
           └─1202 /usr/sbin/atd -f  
  
Oct 06 13:35:44 aasaenko.localdomain systemd[1]: Started atd.service - Deferred execution scheduler.  
Oct 06 13:35:44 aasaenko.localdomain (atd)[1202]: atd.service: Referenced but unset environment variable evaluates to s  
root@aasaenko:/etc/cron.d# at 13:50  
warning: commands will be executed using /bin/sh  
at Mon Oct 6 13:50:00 2025  
at> logger message from at  
at> <EOT>  
job 1 at Mon Oct 6 13:50:00 2025  
root@aasaenko:/etc/cron.d# atq  
1      Mon Oct 6 13:50:00 2025 a root  
root@aasaenko:/etc/cron.d# grep 'from at' /var/log/messages  
root@aasaenko:/etc/cron.d# grep 'from at' /var/log/messages  
Oct 6 13:50:00 aasaenko root[5462]: message from at  
root@aasaenko:/etc/cron.d#
```

Рис. 2.7: Проверка статуса службы atd

Далее я запланировала выполнение команды logger для записи сообщения в системный журнал в указанное время.

Для этого была выполнена команда планирования, после чего в интерактивном режиме я ввела строку:

```
logger message from at
```

После закрытия оболочки комбинацией клавиш Ctrl + D задание было добавлено в очередь выполнения.

Чтобы убедиться, что задание действительно запланировано, я просмотрела список задач командой atq.

В списке появилась строка с указанием времени запуска и идентификатором задания.

Через некоторое время я проверила системный журнал с помощью команды grep 'from at' /var/log/messages.

В логе появилось сообщение «message from at», что подтверждает успешное выполнение запланированной задачи.

3 Контрольные вопросы

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

В cron нет прямого указания для интервала «раз в 2 недели», но можно задать выполнение, например, **по понедельникам каждой второй недели** с помощью комбинации дней и недель:

`0 0 */14 * * команда`

Эта запись означает выполнение задания каждые 14 дней в полночь.

Альтернативно можно использовать `0 0 * * 1 [$(date +%U) -eq 0] &&` команда, чтобы запуск происходил в определённую неделю.

2. **Как настроить задание cron, чтобы оно выполнялось 1-го и 15-го числа каждого месяца в 2 часа ночи?**

Для этого используется следующая запись:

`0 2 1,15 * * команда`

Здесь:

- 0 — минуты,
- 2 — часы,
- 1,15 — числа месяца,
- * * — любой месяц и день недели.

3. **Как настроить задание cron, чтобы оно выполнялось каждые 2 минуты**

каждый день?

Для выполнения задания каждые две минуты используется запись:

`*/2 * * * * команда`

Символ `*/2` в поле минут означает интервал в две минуты.

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

Для этого указываем конкретный день и месяц:

`0 0 19 9 * команда`

Команда будет запускаться каждый год 19 сентября в полночь.

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

Для этого используется следующая запись:

`0 0 * 9 4 команда`

Здесь:

- `0 0` — выполнение в полночь,
- `*` — любой день месяца,
- `9` — сентябрь,
- `4` — четверг (в cron `0` — воскресенье, `1` — понедельник и т.д.).

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

Для назначения задания пользователю `alice` используется ключ `-u`:

`crontab -u alice -e`

Пример:

если в расписание `alice` добавить строку `*/10 * * * * logger Test message`, то она будет выполняться каждые 10 минут от имени пользователя `alice`.

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

Для этого используется файл `/etc/cron.deny`.

Нужно добавить имя пользователя bob в этот файл:

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

После этого попытка bob запустить `crontab` -е завершится ошибкой доступа.

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

Для гарантированного выполнения можно использовать каталог `/etc/cron.daily`, куда помещают сценарии, выполняемые раз в сутки.

Кроме того, можно установить пакет **anacron**, который выполняет пропущенные задачи при следующем включении системы.

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

Для просмотра запланированных заданий используется команда:

```
atq
```

Она выводит список всех заданий, ожидающих выполнения службой atd, с указанием их идентификаторов и времени запуска.

4 Заключение

В ходе выполнения лабораторной работы я изучила принципы работы планировщиков задач в Linux — **cron** и **at**.

Были выполнены следующие действия:

- проверка состояния служб `crond` и `atd`;
- создание и редактирование расписания заданий с помощью `crontab`;
- настройка периодического выполнения команд через файлы `/etc/crontab`, `/etc/cron.hourly` и `/etc/cron.d`;
- добавление одноразового задания с использованием планировщика `at`;
- проверка выполнения задач через системные журналы `/var/log/messages`.

В процессе работы я освоила синтаксис расписаний `cron`, научилась задавать интервалы и конкретные даты для выполнения команд, а также убедилась в правильной работе одноразовых заданий через службу `atd`.