

Отчёт по лабораторной работе №8

Планировщики событий

Максат Хемраев

Содержание

1	Цель работы	5
2	Отчёт по выполнению работы	6
2.1	Планирование задач с помощью <i>cron</i> и <i>at</i>	6
2.1.1	Использование службы <i>cron</i>	6
2.1.2	Планирование заданий с помощью <i>at</i>	10
2.1.3	1. Как настроить задание <i>cron</i> , чтобы оно выполнялось раз в 2 недели?	11
2.1.4	2. Как настроить задание <i>cron</i> , чтобы оно выполнялось 1-го и 15-го числа каждого месяца в 2 часа ночи?	12
2.1.5	3. Как настроить задание <i>cron</i> , чтобы оно выполнялось каждые 2 минуты каждый день?	12
2.1.6	4. Как настроить задание <i>cron</i> , чтобы оно выполнялось 19 сентября ежегодно?	12
2.1.7	5. Как настроить задание <i>cron</i> , чтобы оно выполнялось каждый четверг сентября ежегодно?	13
2.1.8	6. Какая команда позволяет вам назначить задание <i>cron</i> для пользователя <i>alice</i> ? Приведите подтверждающий пример.	13
2.1.9	7. Как указать, что пользователю <i>bob</i> никогда не разрешено назначать задания через <i>cron</i> ? Приведите подтверждающий пример.	13
2.1.10	8. Вам нужно убедиться, что задание выполняется каждый день, даже если сервер во время выполнения временно недоступен. Как это сделать?	14
2.1.11	9. Какая команда позволяет узнать, запланированы ли какие-либо задания на выполнение планировщиком <i>atd</i> ?	14
3	Заключение	15

Список иллюстраций

2.1	Проверка состояния службы crond	6
2.2	Добавление нового задания cron	8
2.3	Проверка добавленного задания	8
2.4	Изменение расписания задания cron	9
2.5	Создание сценария eachhour в cron.hourly	9
2.6	Создание файла eachhour в cron.d	10
2.7	Проверка состояния службы atd	11

Список таблиц

1 Цель работы

Получение навыков работы с планировщиками событий cron и at.

2 Отчёт по выполнению работы

2.1 Планирование задач с помощью *cron* и *at*

2.1.1 Использование службы *cron*

1. В начале работы были получены административные привилегии с помощью команды `su -`.

Затем был проверен статус демона `crond`, отвечающего за выполнение запланированных задач.

Вывод показал, что служба активна и работает корректно.

```
root@mhemraev:~# 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 11:20:59 MSK; 59s ago
 Invocation: 95459b23fa33413a888f4533abe880bb
    Main PID: 1162 (crond)
      Tasks: 1 (limit: 12313)
     Memory: 1M (peak: 1.1M)
        CPU: 3ms
    CGroup: /system.slice/crond.service
            └─1162 /usr/sbin/crond -n

Oct 06 11:20:59 mhemraev.localdomain systemd[1]: Started crond.service - Command Scheduler.
Oct 06 11:20:59 mhemraev.localdomain crond[1162]: (CRON) STARTUP (1.7.0)
Oct 06 11:20:59 mhemraev.localdomain crond[1162]: (CRON) INFO (Syslog will be used instead of sendmail.)
Oct 06 11:20:59 mhemraev.localdomain crond[1162]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 63% if used.)
Oct 06 11:20:59 mhemraev.localdomain crond[1162]: (CRON) INFO (running with inotify support)
root@mhemraev:~#
root@mhemraev:~# 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@mhemraev:~# crontab -l
no crontab for root
root@mhemraev:~#
```

Рис. 2.1: Проверка состояния службы `crond`

2. Далее было изучено содержимое файла `/etc/crontab`, где хранятся системные задания, выполняемые от различных пользователей.

В нём указаны переменные среды, а также пример формата записи задания:

минута (0–59)

час (0–23)

день месяца (1–31)

месяц (1–12)

день недели (0–6)

пользователь команда

Такой синтаксис позволяет точно определять периодичность выполнения заданий.

3. Проверка индивидуального расписания пользователя с помощью `crontab -l` показала, что заданий пока нет.

4. Для добавления нового задания был открыт редактор `crontab` командой `crontab -e`.

В файл добавлена строка:

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

Эта запись означает, что команда `logger` будет выполняться каждую минуту, отправляя сообщение в системный журнал.

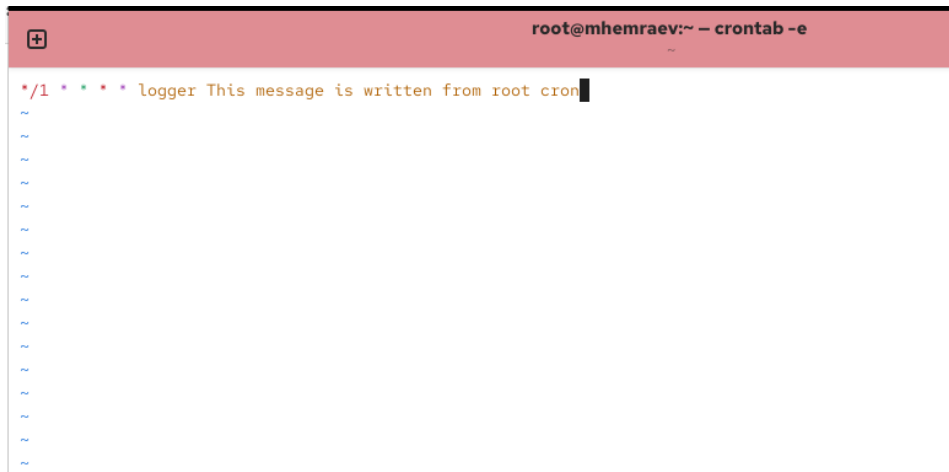


Рис. 2.2: Добавление нового задания cron

- После сохранения изменений список заданий был проверен снова. Новое задание появилось в расписании.

```
crontab: installing new crontab
root@mhemraev:~# crontab -l
*/1 * * * * logger This message is written from root cron
root@mhemraev:~# grep written /var/log/messages
Oct  6 11:25:01 mhemraev root[3791]: This message is written from root cron
root@mhemraev:~# grep written /var/log/messages
Oct  6 11:25:01 mhemraev root[3791]: This message is written from root cron
Oct  6 11:26:01 mhemraev root[3924]: This message is written from root cron
Oct  6 11:27:01 mhemraev root[4052]: This message is written from root cron
root@mhemraev:~#
```

Рис. 2.3: Проверка добавленного задания

- Через несколько минут были просмотрены системные журналы с помощью команды `grep written /var/log/messages`.

В выводе зафиксированы строки с сообщением “This message is written from root cron”, что подтверждает успешное выполнение задания по расписанию.

- Далее запись в crontab была изменена на следующую:

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

Этот синтаксис задаёт выполнение команды в начале каждого часа с понедельника по пятницу.

- 0 — выполнение в 0-й минуте
- */1 — каждый час
- 1-5 — рабочие дни недели (понедельник–пятница)

```

root@mhemraev:~ - crontab -e
0 */1 * * 1-5 logger This message is written from root cron
~
~
~
~
~
~
~

```

Рис. 2.4: Изменение расписания задания cron

8. Для автоматизации выполнения заданий был создан сценарий eachhour в каталоге /etc/cron.hourly.

В нём размещена команда:

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

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

Этот скрипт записывает текущее время в системный журнал при каждом запуске ежечасного расписания.

```

root@mhemraev:/etc/cron.hourly - mcedit
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 в cron.hourly

9. Скрипт был сделан исполняемым командой `chmod +x eachhour`.
10. Затем в каталоге `/etc/cron.d` был создан отдельный файл `eachhour` с содержанием:
11 * * * * root logger This message is written from /etc/cron.d
- Эта запись означает, что команда будет выполняться каждый час в 11-й минуте от имени пользователя `root`.

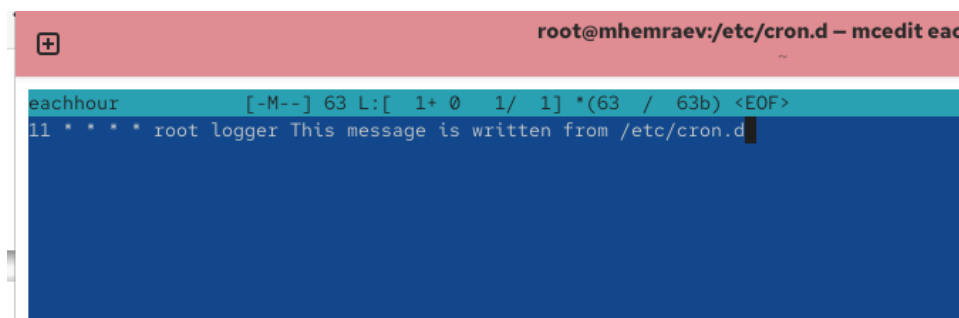


Рис. 2.6: Создание файла `eachhour` в `cron.d`

11. После сохранения файла система начала выполнять задания в соответствии с указанными интервалами, что подтверждается появлением сообщений в журнале `/var/log/messages`.

2.1.2 Планирование заданий с помощью *at*

1. Для разового выполнения задач была использована служба `atd`, позволяющая запускать команды в определённое время.
Проверка статуса показала, что служба активна и работает.

```
root@mhemraev:/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 11:20:59 MSK; 9min ago
     Invocation: 548e98349f114265b14bdadc8392ba61
       Docs: man:atd(8)
    Main PID: 1160 (atd)
      Tasks: 1 (limit: 12313)
     Memory: 316K (peak: 1.1M)
        CPU: 2ms
      CGroup: /system.slice/atd.service
             └─1160 /usr/sbin/atd -f

Oct 06 11:20:59 mhemraev.localdomain systemd[1]: Started atd.service - Deferred execution scheduler.
Oct 06 11:20:59 mhemraev.localdomain (atd)[1160]: atd.service: Referenced but unset environment variable evaluates to an empty string
root@mhemraev:/etc/cron.d#
root@mhemraev:/etc/cron.d# at 11:33
warning: commands will be executed using /bin/sh
at Mon Oct 6 11:33:00 2025
at> logger message from at
at> <EOT>
job 1 at Mon Oct 6 11:33:00 2025
root@mhemraev:/etc/cron.d# atq
1      Mon Oct 6 11:33:00 2025 a root
root@mhemraev:/etc/cron.d# grep 'from at' /var/log/messages
root@mhemraev:/etc/cron.d# grep 'from at' /var/log/messages
Oct 6 11:33:00 mhemraev root[4865]: message from at
root@mhemraev:/etc/cron.d# atq
root@mhemraev:/etc/cron.d#
```

Рис. 2.7: Проверка состояния службы atd

2. Команда для планирования задания: at 11:33.

После ввода открылась интерактивная оболочка, в которую была добавлена команда `logger message from at`.

Завершение ввода осуществлено комбинацией клавиш `Ctrl + D`.

3. Проверка очереди заданий командой `atq` показала, что задание успешно добавлено.

После выполнения команды в указанное время в журнале `/var/log/messages` появилась запись:

```
message from at
```

Это подтверждает корректную работу службы at и успешное выполнение отложенного задания.

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

- В стандартном синтаксисе cron нет прямого параметра «раз в две недели». Однако можно задать выполнение каждое воскресенье и ограничить выполнение с помощью условия в команде.

Например:

`0 0 */14 * * команда` — запуск каждые 14 дней.

Альтернативно можно использовать `anacron` для заданий с интервалом более недели.

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

- Запись в `crontab` будет следующей:

`0 2 1,15 * * команда`

Это означает: выполнять в 2:00 (0 минут, 2 часа) 1-го и 15-го числа каждого месяца.

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

- Используется следующий формат:

`*/2 * * * * команда`

Здесь `*/2` указывает на выполнение каждые две минуты независимо от даты и времени суток.

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

- Следующая запись обеспечивает выполнение один раз в год:

`0 0 19 9 * команда`

Это означает запуск в полночь 19-го числа 9-го месяца (сентября).

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

- Для этого используется комбинация дня недели и месяца:

`0 0 * 9 4` команда

Где 4 обозначает четверг, а 9 — сентябрь.

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

- Команда для редактирования расписания другого пользователя:

`crontab -u alice -e`

Пример записи:

`0 8 * * * /home/alice/backup.sh` — выполнение скрипта резервного копирования от имени пользователя **alice** каждый день в 8:00.

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

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

Достаточно добавить строку с именем пользователя:

`bob`

После этого пользователь **bob** не сможет создавать или редактировать задания cron.

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

- Следует использовать планировщик **anacron**, который гарантирует выполнение пропущенных заданий после включения системы.

Например, запись в `/etc/anacrontab`:

```
1 5 daily_job /usr/local/bin/task.sh
```

Здесь 1 — выполнение раз в день, 5 — задержка в минутах после запуска системы.

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

- Для просмотра очереди заданий используется команда:

```
atq
```

Она выводит список всех отложенных задач, запланированных службой **atd**.

3 Заключение

В ходе работы были изучены и practically освоены средства планирования задач в Linux с использованием служб **cron** и **at**, обеспечивающих автоматизацию и гибкое управление выполнением команд по расписанию.