

# **Лабораторная работа №8**

**Отчет**

Славинский Владислав Вадимович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выводы</b>	<b>16</b>
<b>4</b>	<b>Ответы на контрольные вопросы</b>	<b>17</b>

# Список иллюстраций

2.1	Переход в режим суперпользователя . . . . .	6
2.2	Проверка статуса <code>crond</code> . . . . .	7
2.3	Содержимое файла конфигурации . . . . .	7
2.4	Список заданий . . . . .	8
2.5	Редактирование файла расписания . . . . .	9
2.6	Проверка списка заданий в расписании . . . . .	9
2.7	Журнал системных событий . . . . .	9
2.8	Новая запись . . . . .	10
2.9	Просмотр списка заданий в расписании . . . . .	10
2.10	Создание сценария <code>eachhour</code> . . . . .	11
2.11	Редактирование <code>eachhour</code> . . . . .	11
2.12	Становление файла <code>eachhour</code> исполняемым . . . . .	11
2.13	Создание файла с расписанием в каталоге <code>/etc/crond.d</code> . . . . .	12
2.14	Редактирование <code>eachhour</code> в <code>crond</code> . . . . .	12
2.15	Журнал системных событий . . . . .	13
2.16	Проверка службы <code>atd</code> . . . . .	13
2.17	Настройка выполнения команды . . . . .	14
2.18	Проверка запланировано ли задание . . . . .	14
2.19	Проверка сообщения . . . . .	15
4.1	Подтверждающий пример . . . . .	17

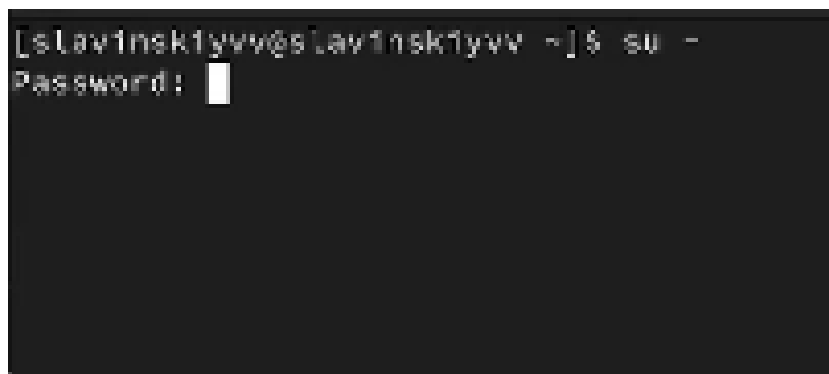
## **Список таблиц**

# 1 Цель работы

Получить навыки работы с планировщиками событий cron и at.

## 2 Выполнение лабораторной работы

Запустим терминал и получим полномочия администратора. (рис. 2.1)

A terminal window with a dark background. The prompt is [slavinskiyvv@slavinskiyvv ~]\$. The user has entered 'su -'. Below this, there is a 'Password:' prompt followed by a small white rectangle representing a masked password.

```
[slavinskiyvv@slavinskiyvv ~]$ su -  
Password: 
```

Рис. 2.1: Переход в режим суперпользователя

Посмотрим статус демона crond: `systemctl status crond -l`. Как видим, служба активна. (рис. 2.2)

```
root@slavinskiyvv:~# su -
Password:
[root@slavinskiyvv ~]# systemctl status crond -l
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; preset: ena
   Active: active (running) since Fri 2025-10-24 21:04:02 MSK; 30min ago
     Main PID: 1188 (crond)
       Tasks: 1 (Limit: 48747)
      Memory: 1.0M
         CPU: 8ms
    CGroup: /system.slice/crond.service
            └─1188 /usr/sbin/crond -n

Oct 24 21:04:02 slavinskiyvv systemd[1]: Started Command Scheduler.
Oct 24 21:04:02 slavinskiyvv crond[1188]: (CRON) STARTUP (1.5.7)
Oct 24 21:04:02 slavinskiyvv crond[1188]: (CRON) INFO (Syslog will be used insto
Oct 24 21:04:02 slavinskiyvv crond[1188]: (CRON) INFO (RANDOM_DELAY will be sca
Oct 24 21:04:02 slavinskiyvv crond[1188]: (CRON) INFO (running with inotify sup
lines 1-15/15 (END)
```

Рис. 2.2: Проверка статуса crond

Посмотрим содержимое файла конфигурации /etc/crontab: cat /etc/crontab. В файле нам показаны несколько переменных окружения такие как SHELL, PATH, MAILTO, и приведены примеры расписаний для задач, которые выполняется ежечасно, ежедневно, еженедельно, ежемесячно. (рис. 2.3)

```
Oct 24 21:04:02 slavinskiyvv systemd[1]: Started Command Scheduler.
Oct 24 21:04:02 slavinskiyvv crond[1188]: (CRON) STARTUP (1.5.7)
Oct 24 21:04:02 slavinskiyvv crond[1188]: (CRON) INFO (Syslog will be used insto
Oct 24 21:04:02 slavinskiyvv crond[1188]: (CRON) INFO (RANDOM_DELAY will be sca
Oct 24 21:04:02 slavinskiyvv crond[1188]: (CRON) INFO (running with inotify sup
[root@slavinskiyvv ~]# 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
# | | | | |
# * * * * * user-name  command to be executed

[root@slavinskiyvv ~]#
```

Рис. 2.3: Содержимое файла конфигурации

Посмотрим список заданий в расписании: crontab -l. И замечаем, что у нас пусто, так как расписание еще не задано. (рис. 2.4)

```

Oct 24 21:04:12 slavinskiyvv libana[1188]: (CRON); INFO (running)
Oct 24 21:04:12 slavinskiyvv crond[1188]: (CRON); INFO (running)
[root@slavinskiyvv ~]# 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,
# | | | | |
# * * * * * user-name    command to be executed

[root@slavinskiyvv ~]# crontab -l
no crontab for root
[root@slavinskiyvv ~]#

```

Рис. 2.4: Список заданий

Откроем файл расписания на редактирование (`crontab -e`) и введем в нем следующую команду: `/1 * * * logger This message is written from root cron`. Синтаксис записи в `crontab` у нас таков: минута, час, день месяца, месяц, день недели. В нашем случае `/1` - это выполнение каждую минуту, а просто каждый, т.е. каждый час, каждый день и т.д. (рис. 2.5)



```
root@slavinskiyv ~# crontab -l
*/1 * * * * logger 'This message is written from root cron'
-- INSERT --
```

Рис. 2.5: Редактирование файла расписания

Посмотрим список заданий в расписании: `crontab -l`. В расписании у нас появилась запись о запланированном событии. (рис. 2.6)

```
[root@slavinskiyv ~]# crontab -l
no crontab for root
[root@slavinskiyv ~]# crontab -e
no crontab for root - using an empty one
crontab: installing new crontab
[root@slavinskiyv ~]# crontab -l
*/1 * * * * logger 'This message is written from root cron'
[root@slavinskiyv ~]#
```

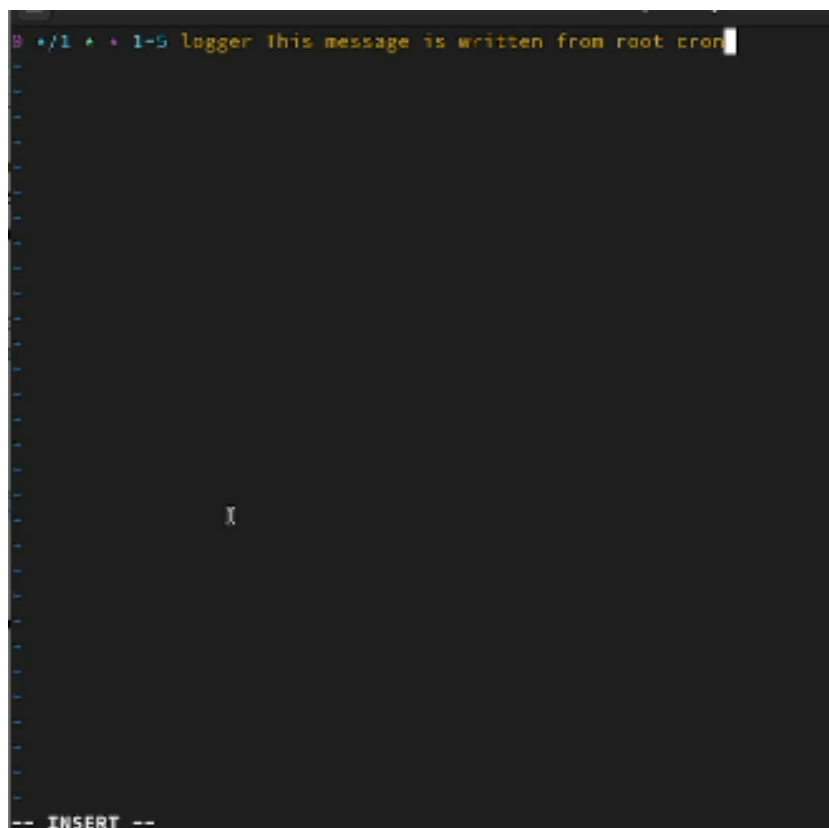
Рис. 2.6: Проверка списка заданий в расписании

Теперь оставим нашу систему на 2-3 минуты и потом посмотрим журнал системных событий: `grep written /var/log/messages`. И как видим, у нас повторяются записи в журнале с интервалом в 1 минуту. (рис. 2.7)

```
[root@slavinskiyv ~]# crontab -l
*/1 * * * * logger 'This message is written from root cron'
[root@slavinskiyv ~]# grep written /var/log/messages
Oct 24 21:37:02 slavinskiyv root[5832]: This message is written from root cron
Oct 24 21:38:01 slavinskiyv root[5861]: This message is written from root cron
Oct 24 21:39:01 slavinskiyv root[5887]: This message is written from root cron
Oct 24 21:40:01 slavinskiyv root[5115]: This message is written from root cron
[root@slavinskiyv ~]#
```

Рис. 2.7: Журнал системных событий

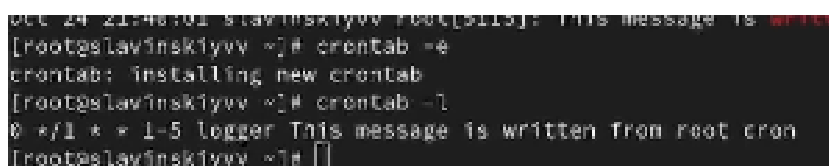
Изменим запись в расписании crontab на следующую: 0 /1 \* \* 1-5 logger This message is written from root cron. В этом же случае при 0 будет происходить выполнение в нулевую минуту каждого часа, \*/1 - это выполнение каждый час, а 1-5 это выполнение с понедельника по пятницу.(рис. 2.8)



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

Рис. 2.8: Новая запись

Посмотрим список заданий в расписании: crontab -l.(рис. 2.9)



```
Oct 24 21:48:01 slavinskiyvv root[6115]: This message is written
[root@slavinskiyvv ~]# crontab -e
crontab: installing new crontab
[root@slavinskiyvv ~]# crontab -l
0 */1 * * 1-5 logger This message is written from root cron
[root@slavinskiyvv ~]#
```

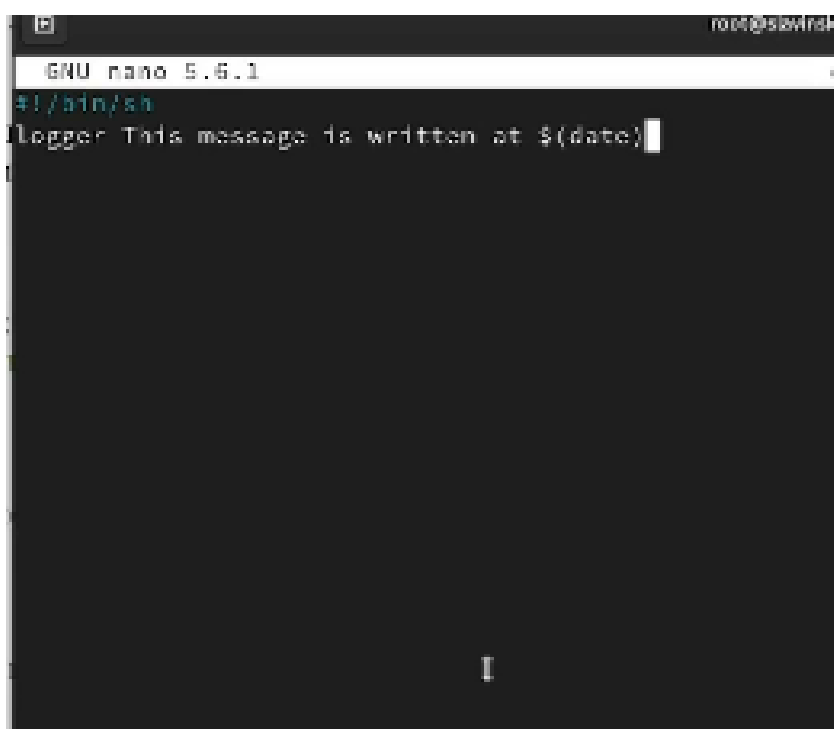
Рис. 2.9: Просмотр списка заданий в расписании

Перейдем в каталог /etc/cron.hourly и создадим в нём файл сценария с именем eachhour: cd /etc/cron.hourly, touch eachhour. (рис. 2.10)

```
Oct 24 21:48:51 slavinskiyvv root[5115]: This message is written
[root@slavinskiyvv ~]# crontab -e
crontab: installing new crontab
[root@slavinskiyvv ~]# crontab -l
8 */1 * * 1-5 logger This message is written from root cron
[root@slavinskiyvv ~]# cd /etc/cron.hourly
[root@slavinskiyvv cron.hourly]# touch eachhour
[root@slavinskiyvv cron.hourly]#
```

Рис. 2.10: Создание сценария eachhour

Откроем eachhour для редактирования и пропишем в нем: `#!/bin/sh`, `logger This message is written at $(date)`. (рис. 2.11)



```
GNU nano 5.6.1
#!/bin/sh
logger This message is written at $(date)
```

Рис. 2.11: Редактирование eachhour

Сделаем файл сценария eachhour исполняемым: `chmod +x eachhour`. (рис. 2.12)

```
[root@slavinskiyvv cron.hourly]# touch eachhour
[root@slavinskiyvv cron.hourly]# nano eachhour
[root@slavinskiyvv cron.hourly]# chmod +x eachhour
[root@slavinskiyvv cron.hourly]#
```

Рис. 2.12: Становление файла eachhour исполняемым

Перейдем в каталог `/etc/cron.d` и создадим в нём файл с расписанием eachhour: `cd /etc/cron.d, touch eachhour`. (рис. 2.13)

```
[root@slavinskiyvv cron.hourly]# nano eachhour
[root@slavinskiyvv cron.hourly]# chmod +x eachhour
[root@slavinskiyvv cron.hourly]# cd /etc/cron.d
[root@slavinskiyvv cron.d]# touch eachhour
[root@slavinskiyvv cron.d]#
```

Рис. 2.13: Создание файла с расписанием в каталоге /etc/crond.d

Откроем файл для редактирования и пропишем следующее: 11 \* \* \* \* root logger This message is written from /etc/cron.d. Здесь же у нас будет выполняться в 11-ую минуту каждого часа и будет выполняться каждый час, каждый день, каждый месяц и каждый день недели.(рис. 2.14)

```
GNU nano 5.6.1                                eachhour
11 * * * * root logger This message is written from /etc/cron.d
1
```

Рис. 2.14: Редактирование eachhour в cron.d

Теперь оставим нашу систему на 2 часа и потом посмотрим в журнал системных событий: `grep written /var/log/messages`. И видим, что запуск сценария eachhour был осуществлен, так как мы видим, что у нас есть записи, сгенерированные сценарием eachhour: is written at Fri Oct 24 10:01:01 PM MSK 2025. (рис. 2.15)

```
root@slavinskiyvv:~# cat /etc/crontab
root@slavinskiyvv:~# nano /etc/crontab
root@slavinskiyvv:~# grep written /var/log/messages
Oct 24 21:37:02 slavinskiyvv root[5032]: This message is written from root cron
Oct 24 21:38:02 slavinskiyvv root[5051]: This message is written from root cron
Oct 24 21:39:01 slavinskiyvv root[5087]: This message is written from root cron
Oct 24 21:40:01 slavinskiyvv root[5115]: This message is written from root cron
Oct 24 21:41:01 slavinskiyvv root[5144]: This message is written from root cron
Oct 24 22:00:01 slavinskiyvv root[5429]: This message is written from root cron
Oct 24 22:01:01 slavinskiyvv root[5453]: This message is written at Fri Oct 24 18:31:01 PM MSK 2025
Oct 24 22:09:01 slavinskiyvv root[5589]: This message is written from /etc/crontab
Oct 24 23:09:01 slavinskiyvv root[6139]: This message is written from root cron
Oct 24 23:01:01 slavinskiyvv root[6159]: This message is written at Fri Oct 24 11:31:01 PM MSK 2025
Oct 24 23:11:01 slavinskiyvv root[6274]: This message is written from /etc/crontab
root@slavinskiyvv:~#
```

Рис. 2.15: Журнал системных событий

Проверим, что служба atd загружена и включена: `systemctl status atd`. Как видим, все работает.(рис. 2.16)

```
root@slavinskiyvv:~# systemctl status atd
[slavinskiyvv@slavinskiyvv ~]$ su -
Password:
[root@slavinskiyvv ~]# systemctl status atd
• atd.service - Deferred execution scheduler
   Loaded: loaded (/usr/lib/systemd/system/atd.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-10-24 21:04:02 MSK; 2h 53min ago
     Docs: man:atd(8)
    Main PID: 1184 (atd)
      Tasks: 1 (Limit: 48747)
    Memory: 896.0K
       CPU: 10ms
    CGroup: /system.slice/atd.service
            └─1184 /usr/sbin/atd -f

Oct 24 21:04:02 slavinskiyvv systemd[1]: Started Deferred execution scheduler.
[root@slavinskiyvv ~]#
```

Рис. 2.16: Проверка службы atd

Зададим выполнение команды `logger message from at` в 23:59. Далее введем `logger message from at` и используем `ctrl+d` для закрытия оболочки.(рис. 2.17)

```
Oct 24 21:04:02 slavinskiyvv systemd[1]: Started Deferre
[root@slavinskiyvv ~]# at 23:59
warning: commands will be executed using /bin/sh
at> logger message from at
at> <EOF>
job 1 at Fri Oct 24 23:59:00 2025
[root@slavinskiyvv ~]#
```

Рис. 2.17: Настройка выполнения команды

Проверяем, что задание запланировано: atq. Видим, что оно запланировано.(рис. 2.18)

```
[root@slavinskiyvv ~]# at 23:59
warning: commands will be executed using /bin/sh
at> logger message from at
at> <EOF>
job 1 at Fri Oct 24 23:59:00 2025
[root@slavinskiyvv ~]# atq
1          Fri Oct 24 23:59:00 2025 a root
[root@slavinskiyvv ~]#
```

Рис. 2.18: Проверка запланировано ли задание

И в 23:59 вводим команду grep 'from at' /var/log/messages, чтобы проверить, появилось ли сообщение в нужное нам время. И как видим, у нас все работает.(рис. 2.19)

```
[root@slavinskiyvv ~]# grep 'from at' /var/log/messages  
Oct 24 23:59:00 slavinskiyvv root[6965]: message from at  
[root@slavinskiyvv ~]#
```

Рис. 2.19: Проверка сообщения

## 3 Выводы

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



## 4 Ответы на контрольные вопросы

1. 0 0 1,15 \* \* выполнение 1-ого и 15-го числа каждый месяц
2. 0 2 1,15 \* \* нулевая минута, 2:00 ночи, 1,15 числа месяца, любой месяц, любой день недели
3. /2 \* \* \* каждые две минуты
4. 0 0 19 9 \* в 00:00, 19-ое число, сентябрь, любой день недели
5. 0 0 \* 9 4 в 00:00, каждый день месяца, сентябрь, четверг
6. crontab -u alice -e, 0 9 \* \* 1-5 /home/scripts/daily\_report.sh, crontab -u alice -l (рис. 4.1)

```
-s      selinux context
-V      print version and exit
-x <mask> enable debugging

Default operation is replace, per 1003.2
[root@slavinskiyv ~]# crontab -u alice -l
0 9 * * 1-5 /home/alice/scripts/daily_report.sh
[root@slavinskiyv ~]#
```

Рис. 4.1: Подтверждающий пример