

Планировщики событий

Часть 1

Славинский В.В.

25 октября 2025

Российский университет дружбы народов, Москва, Россия Россия

Информация

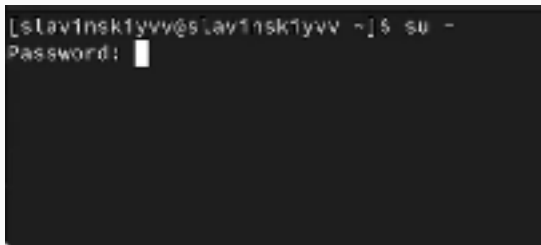
..... {.columns align=center} ::: {.column width="70%"}

- Славинский Владислав Вадимович
- Студент
- Российский университет дружбы народов
- [1132246169@pfur.ru]

::: ::: {.column width="30%"}

Вводная часть

Запустим терминал и получим полномочия администратора.



```
[slav1nsk1yvv@slav1nsk1yvv ~]$ su -  
Password: █
```

Рис. 1: sc1

Проверка статуса crond

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

```
root@slavinskiyvv:~  
[slavinskiyvv@slavinskiyvv ~]$ su -  
Password:  
[root@slavinskiyvv ~]# systemctl status crond -l  
• crond.service - Command Scheduler  
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; preset: en  
   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 inst  
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)
```

Содержимое файла конфигурации

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 выполняются ежечасно, ежедневно, еженедельно, ежемесячно.

```
root@slavinskiyvv ~  
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 inst  
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)
```

Список заданий

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

```
Oct 24 21:04:82 slavinskiyvv crond[1188]: (CRON) INFO (RANDOM_
Oct 24 21:04:82 slavinskiyvv crond[1188]: (CRON) INFO (running
[roet@slavinskiyvv ~]$ cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

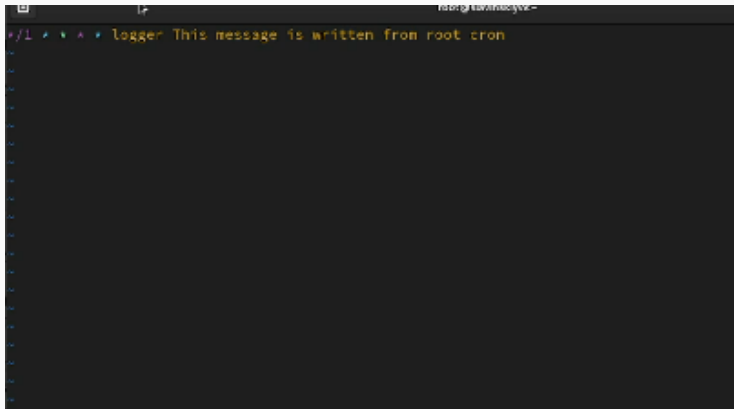
# For details see man 4 crontabs

# 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

[roet@slavinskiyvv ~]$ crontab -l
```


Редактирование файла расписания

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.



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

Проверка списка заданий в расписании

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

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

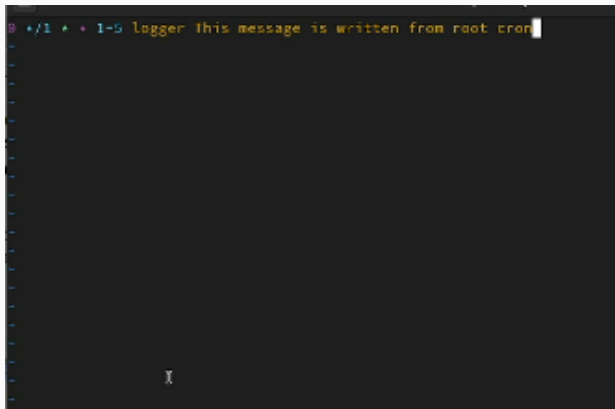
Рис. 6: sc6

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.

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

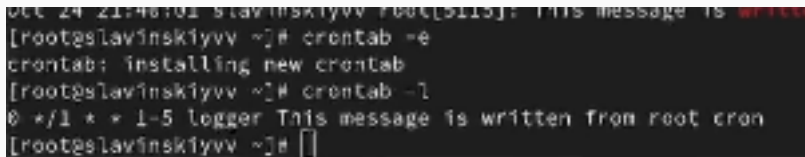
Рис. 7: sc7

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.

A screenshot of a terminal window with a dark background. The prompt is a root symbol. The command entered is `*/1 * 1-5 logger This message is written from root cron`. The cursor is at the end of the command.

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

Посмотрим список заданий в расписании: `crontab -l`.

A terminal window showing the installation and listing of a crontab. The prompt is [root@slavinskiyvv ~]#. The command crontab -e is entered, followed by crontab: installing new crontab. Then crontab -l is entered, followed by the output: @ */1 * * 1-5 logger This message is written from root cron. The prompt returns to [root@slavinskiyvv ~]#.

```
[root@slavinskiyvv ~]# crontab -e
crontab: installing new crontab
[root@slavinskiyvv ~]# crontab -l
@ */1 * * 1-5 logger This message is written from root cron
[root@slavinskiyvv ~]#
```

Рис. 9: sc9

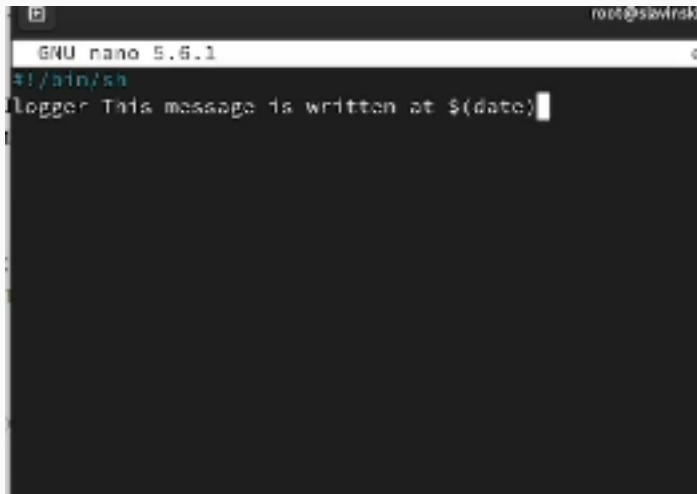
Перейдем в каталог /etc/cron.hourly и создадим в нём файл сценария с именем eachhour: `cd /etc/cron.hourly, touch eachhour`.

```
Oct 24 21:40:51 slavinskiyvv root[5115]: 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 ~]# cd /etc/cron.hourly
[root@slavinskiyvv cron.hourly]# touch eachhour
[root@slavinskiyvv cron.hourly]#
```

Рис. 10: sc10

Редактирование eachhour

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



```
root@slavinsk:~# nano /etc/cron.d/eachhour
GNU nano 5.6.1 /etc/cron.d/eachhour
#!/bin/sh
logger This message is written at $(date)
```

Сделаем файл сценария eachhour исполняемым: `chmod +x eachhour`.

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

Рис. 12: sc12

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

```
[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. 13: sc13

Редактирование eachhour в cron.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.



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

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`.

```
[root@slavinskiyvv cron.d]# cat /etc/cron.hourly/eachhour
[root@slavinskiyvv cron.d]# nano eachhour
[root@slavinskiyvv cron.d]# grep written /var/log/messages
Oct 24 21:37:02 slavinskiyvv root[5032]: This message is written from root cron
Oct 24 21:38:01 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[5425]: This message is written from root cron
Oct 24 22:01:01 slavinskiyvv root[5453]: This message is written at Fri Oct 24 10:01:01 PM MSK 2025
Oct 24 22:11:01 slavinskiyvv root[5583]: This message is written from /etc/cron.d
Oct 24 23:00:01 slavinskiyvv root[6133]: This message is written from root cron
Oct 24 23:01:01 slavinskiyvv root[6155]: This message is written at Fri Oct 24 11:01:01 PM MSK 2025
Oct 24 23:11:01 slavinskiyvv root[6274]: This message is written from /etc/cron.d
[root@slavinskiyvv cron.d]#
```

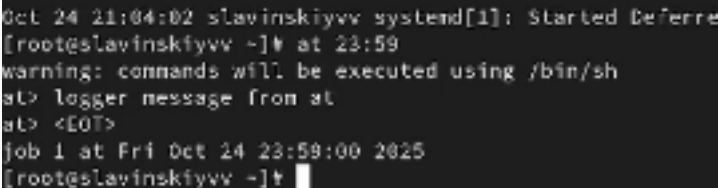
Проверка службы atd

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

```
root@slavinskiyvv:~# su -
[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: 40747)
   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 ~]#
```

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

A terminal window with a dark background and light gray text. The text shows the execution of the 'at' command to schedule a job. The output includes the system time, the user 'slavinskiyvv', the job status, and the command to be executed. The prompt returns to the root user after the job is scheduled.

```
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> <[OT>
job 1 at Fri Oct 24 23:59:00 2025
[root@slavinskiyvv ~]#
```

Рис. 17: sc17

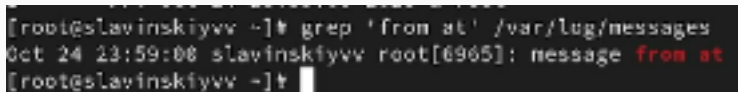
Проверка запланировано ли задание

Проверяем, что задание запланировано: atq. Видим, что оно запланировано.

```
[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. 18: sc18

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.

A terminal window with a dark background. The prompt is [root@slavinskiyvv ~]#. The command entered is grep 'from at' /var/log/messages. The output is Oct 24 23:59:00 slavinskiyvv root[6965]: message from at. The words 'from at' are highlighted in red. The prompt returns to [root@slavinskiyvv ~]# with a cursor.

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

Рис. 19: sc19