

Отчет о лабораторной работе

Лабораторная работа №8

Казначеев Сергей Ильич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Контрольные вопросы	10
4	Выводы	12

Список иллюстраций

2.1	1	6
2.2	2	6
2.3	3	6
2.4	4	7
2.5	5	7
2.6	6	7
2.7	7	7
2.8	8	7
2.9	9	8
2.10	10	8
2.11	11	8
2.12	12	8
2.13	13	8
2.14	14	9
2.15	15	9
2.16	16	9
2.17	17	9

Список таблиц

1 Цель работы

Получение навыков работы с планировщиками событий cron и at

2 Выполнение лабораторной работы

Для начала откроем терминал и перейдем в супер пользователя, затем посмотрим статус демона crond рис. 2.1).

```
[sikaznacheev@localhost ~]$ su -
Пароль:
[root@localhost ~]# systemctl status crond -l
* crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-10-21 12:49:32 +03; 8min ago
     Main PID: 1247 (crond)
       Tasks: 1 (limit: 48740)
      Memory: 1.0M
         CPU: 30ms
    CGroup: /system.slice/crond.service
            └─1247 /usr/sbin/crond -n

окт 21 12:49:32 localhost.localdomain systemd[1]: Started Command Scheduler.
окт 21 12:49:32 localhost.localdomain crond[1247]: (CRON) STARTUP (1.5.7)
окт 21 12:49:32 localhost.localdomain crond[1247]: (CRON) INFO (Syslog will be used instead of sendmail.)
окт 21 12:49:32 localhost.localdomain crond[1247]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 30% if used.)
окт 21 12:49:32 localhost.localdomain crond[1247]: (CRON) INFO (running with inotify support)
```

Рис. 2.1: 1

Теперь посмотрим содержимое файла конфигурации /etc/crontab (рис. 2.2).

```
[root@localhost ~]# 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
```

Рис. 2.2: 2

После чего посмотрим список заданий в расписании и он окажется пуст (рис.2.3).

```
[root@localhost ~]# crontab -l
no crontab for root
```

Рис. 2.3: 3

Далее открываем файл расписания на редактирования командой crontab -e и записываем `/1 * * * logger This message is written from root cron` (рис. 2.4).




Рис. 2.4: 4

После чего проверяем список заданий в расписании (рис. 2.5).

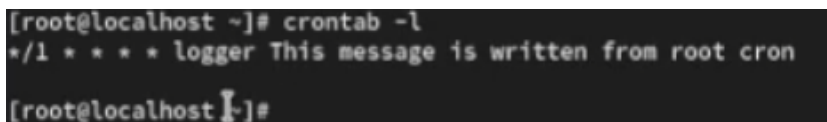


Рис. 2.5: 5

Не выключая систему через 2-3 мин посмотрим журнал системных событий (рис. 2.6).

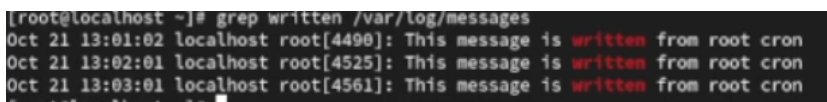


Рис. 2.6: 6

После чего изменяем запись в расписании crontab на следующую `0 /1 * * 1-5 logger This message is written from root cron` (рис. 2.7).



Рис. 2.7: 7

Затем просматриваем список заданий в расписании (рис. 2.8).

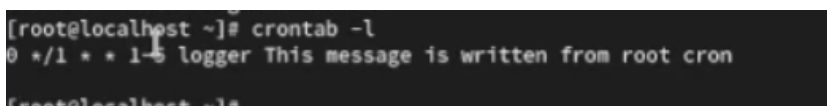


Рис. 2.8: 8

После чего переходим в каталог `/etc/cron.hourly` и создаем файл сценария с именем `crachhour` (рис. 2.9).

```
[root@localhost ~]# cd /etc/cron.hourly
[root@localhost cron.hourly]# touch eachhour
```

Рис. 2.9: 9

Далее в этом файле записываем `#!/bin/sh logger This message is written at $(date)` (рис. 2.10).

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

Рис. 2.10: 10

После чего сделаем файл сценария `eachhour` исполняемым и перейдем в каталог `/etc/crond.d` и создадим файл с расписанием `eachhour` (рис. 2.11).

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

Рис. 2.11: 11

В него запишем следующий скрипт `11 * * * * root logger This message is written from /etc/cron.d` данный скрипт каждую 11 минуту каждого часа, любого дня и месяца, cron запускает команду `logger` от имени пользователя `root` (рис. 2.12).

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

Рис. 2.12: 12

Не выключая систему через 2-3 часа просмотрели журнал системных событий с помощью команды `grep written /var/log/messages`

```
[root@localhost cron.d]# grep written /var/log/messages
Oct 21 13:01:02 localhost root[4490]: This message is written from root cron
Oct 21 13:02:01 localhost root[4525]: This message is written from root cron
Oct 21 13:03:01 localhost root[4561]: This message is written from root cron
Oct 21 13:04:02 localhost root[4622]: This message is written from root cron
Oct 21 13:11:01 localhost root[4885]: This message is written from /etc/cron.d
Oct 21 14:00:01 localhost root[5686]: This message is written from root cron
Oct 21 14:01:01 localhost root[5718]: This message is written at Бт 21 окт 2025 14:01:01 +03
Oct 21 14:11:01 localhost root[5850]: This message is written from /etc/cron.d
Oct 21 15:00:01 localhost root[6673]: This message is written from root cron
Oct 21 15:01:01 localhost root[6706]: This message is written at Бт 21 окт 2025 15:01:01 +03
Oct 21 15:11:01 localhost root[6828]: This message is written from /etc/cron.d
```

Рис. 2.13: 13

После чего запустили терминал и получили полномочия администратора и проверили что служба atd загружена и включена (рис. 2.14).

```
sikaznacheev@localhost ~]$ su -
Пароль:
[root@localhost ~]# systemctl status atd
● atd.service - Deferred execution scheduler
   Loaded: loaded (/usr/lib/systemd/system/atd.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-10-21 12:49:32 +03; 20min ago
     Docs: man:atd(8)
    Main PID: 1245 (atd)
      Tasks: 1 (limit: 48740)
     Memory: 904.0K
        CPU: 31ms
    CGroup: /system.slice/atd.service
            └─1245 /usr/sbin/atd -f

окт 21 12:49:32 localhost.localdomain systemd[1]: Started Deferred execution scheduler.
```

Рис. 2.14: 14

Далее зададим выполнение команды logger message from at в 13:11(рис. 2.15).

```
[root@localhost ~]# at 13:11
warning: commands will be executed using /bin/sh
at>
```

Рис. 2.15: 15

Теперь проверим что задание запланировано

```
[root@localhost ~]# atq
1      Tue Oct 21 13:11:00 2025 a root
```

Рис. 2.16: 16

С помощью команды grep 'from at' /var/log/messages просмотрим появилось ли соответствующее сообщение в лог-файле в указанное время и как увидим да все получилось (рис. 2.17).

```
[root@localhost ~]# grep 'from at' /var/log/messages
Oct 21 13:11:00 localhost root[4882]: message from at
```

Рис. 2.17: 17

3 Контрольные вопросы

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

Ответ - найтрока `0 2 * * 1 test $(($(date +%V) % 2)) -eq 0 && cmd`

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

Ответ - найтрока `0 2 1,15 * * /path/to/script.sh`

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

Ответ - настройка `/2 * * * /path/to/script.sh`

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

Ответ - настройка `0 0 19 9 * /path/to/script.sh`

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

Ответ - настройка `0 0 * 9 4 /path/to/script.sh`

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

Ответ - команда `sudo crontab -u alice -e` Пример `0 3 * * * /path/to/script.sh`

7. Какая команда позволяет вам видеть сообщения journald после последней перезагрузки системы?

Ответ - нужно указать `echo bob | sudo tee -a /etc/cron.deny`

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

Ответ - это можно сделать прописав `persistent=true`

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

Ответ - команда `atq`

4 Выводы

В результате выполнения лабораторной работы я получил навыки работы с планировщиками событий `stop` и `at`