

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

дисциплина: Основы администрирования операционных систем

Студент: Ко Антон Геннадьевич

Студ. билет № 1132221551

Группа: НПИбд-02-23

МОСКВА

2024 г.

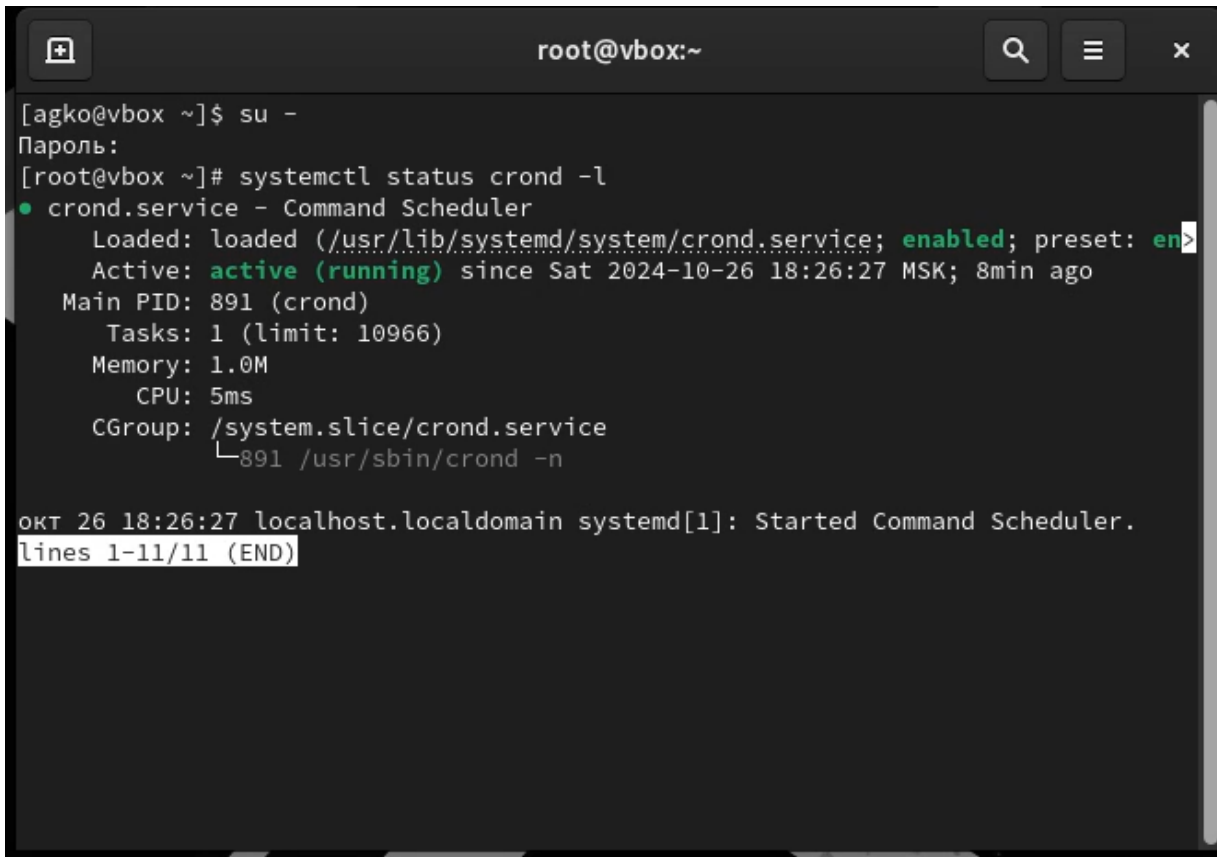
Цель работы:

Целью данной работы является получение навыков работы с планировщиками событий cron и at.

Планирование задач с помощью cron:

Мониторинг журнала системных событий в реальном времени:

Запустим терминал и получим полномочия администратора: **su -**.
Просмотрим статус демона crond: **systemctl status crond -l** и содержимое файла конфигурации **/etc/crontab**: **cat /etc/crontab**:



```
root@vbox:~
[agko@vbox ~]$ su -
Пароль:
[root@vbox ~]# systemctl status crond -l
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; preset: en
   Active: active (running) since Sat 2024-10-26 18:26:27 MSK; 8min ago
     Main PID: 891 (crond)
       Tasks: 1 (limit: 10966)
      Memory: 1.0M
         CPU: 5ms
    CGroup: /system.slice/crond.service
            └─891 /usr/sbin/crond -n

окт 26 18:26:27 localhost.localdomain systemd[1]: Started Command Scheduler.
lines 1-11/11 (END)
```

Рис. 1. Запуск терминала и получение полномочий администратора, просмотр статуса демона crond и содержимого файла конфигурации **/etc/crontab**.

```
[root@vbox ~]# 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. Просмотр содержимого файла конфигурации /etc/crontab.

Теперь посмотрим список заданий в расписании: **crontab -l**. Ничего не отобразилось, так как расписание ещё не задано. Далее откроем файл расписания на редактирование: **crontab -e**:

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

Рис. 3. Просмотр списка заданий в расписании, открытие файла расписания на редактирование.

Предыдущая команда запустила интерфейс редактора (по умолчанию используется vi). Добавим следующую строку в файл расписания (запись сообщения в системный журнал), используя **Ins** для перехода в vi в режим ввода: ***/1 * * * * logger This message is written from root cron**. Закроем сеанс редактирования vi и сохраним изменения, используя команду vi: **Esc : wq**.

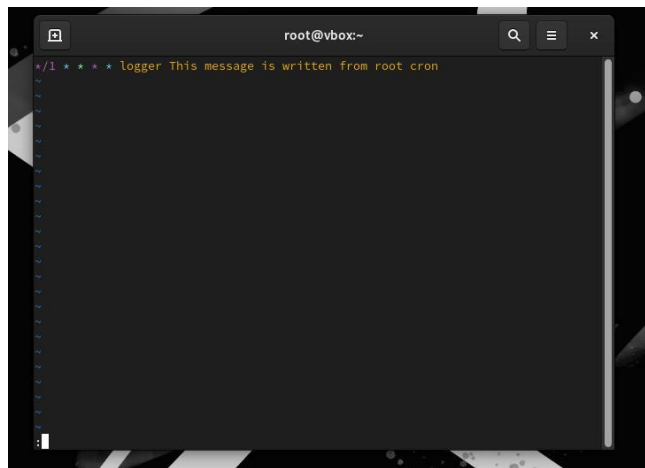


Рис. 4. Открытие текстового редактора vi и добавление строки в файл расписания.

Просмотрим список заданий в расписании: **crontab -l** (в расписании появилась запись о запланированном событии). Не выключая систему, через некоторое время (2–3 минуты) посмотрим журнал системных событий: **grep written /var/log/messages**:

```
[root@vbox ~]# crontab -l
*/1 * * * * logger This message is written from root cron
[root@vbox ~]# grep written /var/log/messages
Oct 26 19:14:01 vbox root[6402]: This message is written from root cron
Oct 26 19:15:02 vbox root[6468]: This message is written from root cron
```

Рис. 5. Просмотр списка заданий в расписании, просмотр журнала системных событий.

Вернёмся в текстовый редактор vi и изменим запись в расписании crontab на следующую: **0 */1 * * 1-5 logger This message is written from root cron**

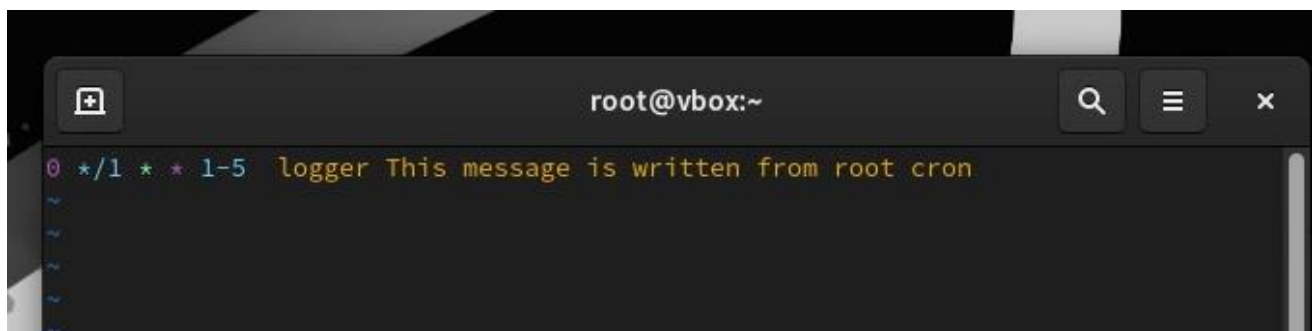


Рис. 6. Изменение записи в расписании crontab.

Теперь посмотрим список заданий в расписании: **crontab -l**.

```
[root@vbox ~]# crontab -e
\crontab: installing new crontab
[root@vbox ~]# crontab -l
0 */1 * * 1-5 logger This message is written from root cron
```

Рис. 7. Просмотр списка заданий в расписании.

Перейдём в каталог `/etc/cron.hourly` и создадим в нём файл сценария с именем `eachhour`:

```
cd /etc/cron.hourly
```

```
touch eachhour
```

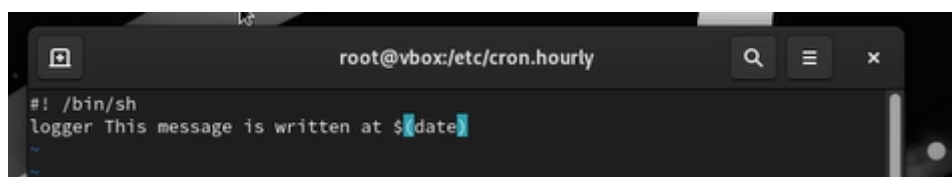
Далее откроем файл `eachhour` для редактирования и пропишем в нём следующий скрипт (запись сообщения в системный журнал)

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

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

```
[root@vbox ~]# cd /etc/cron.hourly
[root@vbox cron.hourly]# touch eachhour
[root@vbox cron.hourly]# eachhour -e
bash: eachhour: команда не найдена...
[root@vbox cron.hourly]# -e
bash: -e: команда не найдена...
[root@vbox cron.hourly]# #!/bin/sh
[root@vbox cron.hourly]# ls
ananacron  eachhour
[root@vbox cron.hourly]# eachhour -e
bash: eachhour: команда не найдена...
[root@vbox cron.hourly]# touch eachhour
[root@vbox cron.hourly]# vi eachhour
[root@vbox cron.hourly]# cd eachhou
```

Рис. 8. Открытие каталога `/etc/cron.hourly` и создание в нём файла сценария с именем `eachhour`.



```
root@vbox:/etc/cron.hourly
#!/bin/sh
logger This message is written at $(date)
```

Рис. 9. Открытие файла `eachhour` для редактирования и прописывание в нём скрипта.

Сделаем файл сценария `eachhour` исполняемым:

`chmod +x eachhour`

Теперь перейдём в каталог `/etc/cron.d` и создадим в нём файл с расписанием `eachhour`

`cd /etc/cron.d`

`touch eachhour`

Откроем этот файл для редактирования и поместим в него следующее содержимое:

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

Сохраним изменения.

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

Рис. 10. Делаем файл сценария `eachhour` исполняемым, открытие каталога `/etc/cron.d` и создание в нём файла с расписанием `eachhour`, открытие файла на редактирование.

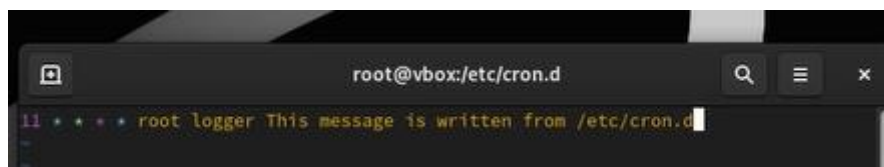
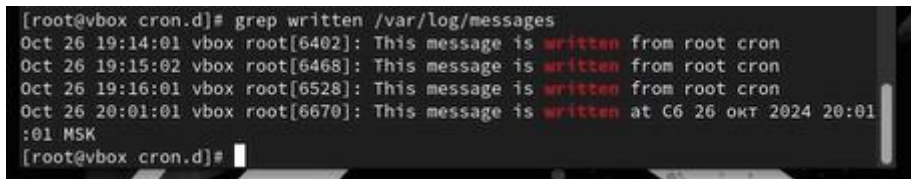


Рис. 11. Добавление содержимого в файл и сохранение изменений.

Не выключая систему, через некоторое время (час) посмотрим журнал системных событий:

grep written /var/log/messages

По журналу видим, что запуск сценария eachhour в соответствии с заданным расписанием был осуществлён



```
[root@vbox cron.d]# grep written /var/log/messages
Oct 26 19:14:01 vbox root[6402]: This message is written from root cron
Oct 26 19:15:02 vbox root[6468]: This message is written from root cron
Oct 26 19:16:01 vbox root[6528]: This message is written from root cron
Oct 26 20:01:01 vbox root[6670]: This message is written at C6 26 окт 2024 20:01
:01 MSK
[root@vbox cron.d]#
```

Рис. 12. Просмотр журнала системных событий через час.

Планирование заданий с помощью at:

Запустим терминал и получим полномочия администратора: **su -**. Проверим, что служба atd загружена и включена: **systemctl status atd**. Зададим выполнение команды **logger message from at** в 9:30. Для этого введём **at 9:30**, а затем **logger message from at**. Используем **Ctrl + d**, чтобы закрыть оболочку. Убедимся, что задание действительно запланировано: **atq**. С помощью команды **grep from at /var/log/messages** посмотрим, появилось ли соответствующее сообщение в лог-файле в указанное вами время (сообщение появилось).

```

[agko@vbox ~]$ su -
Пароль:
[root@vbox ~]# systemctl status atd
• atd.service - Deferred execution scheduler
   Loaded: loaded (/usr/lib/systemd/system/atd.service; enabled; preset: enab
   Active: active (running) since Sat 2024-10-26 18:26:27 MSK; 1h 37min ago
     Docs: man:atd(8)
    Main PID: 890 (atd)
      Tasks: 1 (limit: 10966)
     Memory: 300.0K
        CPU: 3ms
    CGroup: /system.slice/atd.service
            └─890 /usr/sbin/atd -f

окт 26 18:26:27 localhost.localdomain systemd[1]: Started Deferred execution sc
lines 1-12/12 (END)

```

Рис. 13. Запуск терминала и получение полномочий администратора, проверка загрузки и включения службы atd.

```

lines 1-12/12 (END)
[root@vbox ~]# at 20:05
warning: commands will be executed using /bin/sh
at> ^[[
at>
at> logger message from at
at> <EOT>
job 1 at Sun Oct 27 20:05:00 2024
[root@vbox ~]# atq
1          Sun Oct 27 20:05:00 2024 a root
[root@vbox ~]# at 20:07
warning: commands will be executed using /bin/sh
at> logger message from at
at> <EOT>
job 2 at Sat Oct 26 20:07:00 2024
[root@vbox ~]# grep 'from at' /var/log/messages
Oct 26 20:07:00 vbox root[6816]: message from at
[root@vbox ~]#

```


Рис. 14. Создание задачи выполнения команды `logger message from at` в 9:30, закрытие оболочки. Проверка планировки задачи и выполнения её в указанное время.

Ответы на контрольные вопросы:

1. Как настроить задание cron, чтобы оно выполнялось раз в 2 недели? `00 00 1,15 * * logger task`
2. Как настроить задание cron, чтобы оно выполнялось 1-го и 15-го числа каждого месяца в 2 часа ночи? `00 02 1,15 * * logger task`
3. Как настроить задание cron, чтобы оно выполнялось каждые 2 минуты каждый день? `*/2 * * * * logger task`
4. Как настроить задание cron, чтобы оно выполнялось 19 сентября ежегодно? `* * 19 9 logger task`
5. Как настроить задание cron, чтобы оно выполнялось каждый четверг сентября ежегодно? `* * * * 4 logger task`
6. Какая команда позволяет вам назначить задание cron для пользователя `alice`? Приведите подтверждающий пример. `* * * * alice logger task`
7. Как указать, что пользователю `bob` никогда не разрешено назначать задания через cron? Приведите подтверждающий пример. записать его в `/etc/cron.deny`
8. Вам нужно убедиться, что задание выполняется каждый день, даже если сервер во время выполнения временно недоступен. Как это сделать? Найти задание в логах `grep cron /var/log/messages`
9. Какая команда позволяет узнать, запланированы ли какие-либо задания на выполнение планировщиком `atd`? `atq`

Вывод:

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