

Отчёт по лабораторной работе №8

Дисциплина: Основы администрирования операционных систем

Верниковская Екатерина Андреевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Планирование задач с помощью cron	7
3.2	Планирование заданий с помощью at	15
4	Контрольные вопросы + ответы	17
5	Выводы	19
6	Список литературы	20

Список иллюстраций

3.1	Режим суперпользователя	7
3.2	Статус демона crond	7
3.3	Содержимое файла конфигурации /etc/crontab	8
3.4	Список заданий	8
3.5	Открытие файла расписания	8
3.6	Файл расписания	9
3.7	Редактирование файла расписания (1)	9
3.8	Закрытие сеанса редактирование и сохранение изменений	10
3.9	Список заданий после редактирования файла расписания (1) . . .	10
3.10	Журнал системных событий (1)	11
3.11	Редактирование файла расписания (2)	11
3.12	Список заданий после редактирования файла расписания (2) . . .	12
3.13	Переход в каталог /etc/cron.hourly и создание файла сценария с именем eachhour	12
3.14	Открытие файла сценария eachhour	12
3.15	Файл сценария eachhour	13
3.16	Редактирование файла сценария eachhour	13
3.17	Права на исполнение файла сценария eachhour	13
3.18	Переход в каталог /etc/crond.d и создание файла с расписанием eachhour	13
3.19	Открытие файла с расписанием eachhour	13
3.20	Файл с расписанием eachhour	14
3.21	Редактирование файла с расписанием eachhour	14
3.22	Журнал системных событий (2)	15
3.23	Проверка статуса службы atd	15
3.24	Задаём выполнение команды logger message from at в 23:43	16
3.25	Проверка (1)	16
3.26	Проверка (2)	16

Список таблиц

1 Цель работы

Получение навыков работы с планировщиками событий cron и at.

2 Задание

1. Выполнить задания по планированию задач с помощью crond
2. Выполнить задания по планированию задач с помощью atd

3 Выполнение лабораторной работы

3.1 Планирование задач с помощью cron

Запускаем терминала и получаем полномочия суперпользователя, используя *su* - (рис. 3.1)

```
[eavernikovskaya@eavernikovskaya ~]$ su -  
Password:  
[root@eavernikovskaya ~]#
```

Рис. 3.1: Режим суперпользователя

Смотрим статус демона *crond* с помощью *systemctl status crond -l* (рис. 3.2)

```
[root@eavernikovskaya ~]# systemctl status crond -l  
● crond.service - Command Scheduler  
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; preset: enabled)  
   Active: active (running) since Thu 2024-10-24 20:09:35 MSK; 3min 22s ago  
     Main PID: 1084 (crond)  
       Tasks: 1 (limit: 12205)  
      Memory: 1.0M  
         CPU: 7ms  
    CGroup: /system.slice/crond.service  
            └─1084 /usr/sbin/crond -n  
  
Oct 24 20:09:35 eavernikovskaya.localdomain systemd[1]: Started Command Scheduler.  
Oct 24 20:09:35 eavernikovskaya.localdomain crond[1084]: (CRON) STARTUP (1.5.7)  
Oct 24 20:09:35 eavernikovskaya.localdomain crond[1084]: (CRON) INFO (Syslog will be used instead of sendmail.)  
Oct 24 20:09:35 eavernikovskaya.localdomain crond[1084]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 31% if   
Oct 24 20:09:35 eavernikovskaya.localdomain crond[1084]: (CRON) INFO (running with inotify support)  
[root@eavernikovskaya ~]#
```

Рис. 3.2: Статус демона *crond*

Далее смотрим содержимое файла конфигурации */etc/crontab* (рис. 3.3)

```
[root@eavernikovskaya ~]# 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@eavernikovskaya ~]#
```

Рис. 3.3: Содержимое файла конфигурации /etc/crontab

Далее смотрим список заданий в расписании: *crontab -l*. Ничего не отобразится, так как расписание ещё не задано (рис. 3.4)

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

Рис. 3.4: Список заданий

Открываем файл расписания на редактирование: *crontab -e* (рис. 3.5), (рис. 3.6)

```
[root@eavernikovskaya ~]# crontab -e
```

Рис. 3.5: Открытие файла расписания

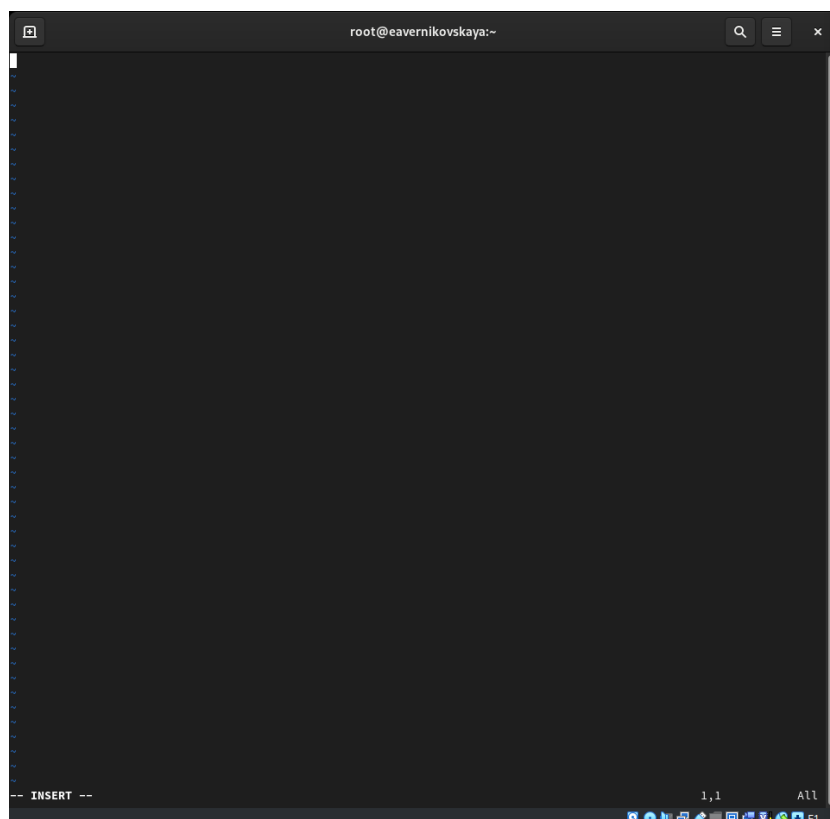


Рис. 3.6: Файл расписания

Команда *crontab -e* запустит интерфейс редактора (по умолчанию используется vi). Добавляем следующую строку в файл расписания (запись сообщения в системный журнал), используя клавишу Ins для перехода в vi в режим ввода: `/1 * * * * logger This message is written from root cron` (рис. 3.7)

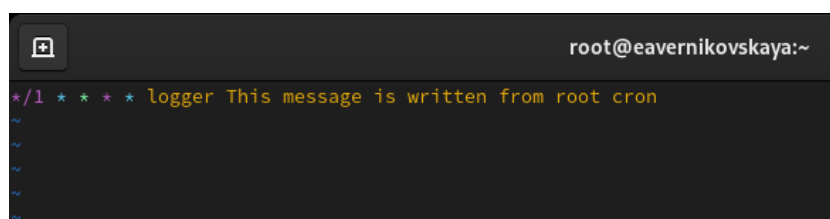


Рис. 3.7: Редактирование файла расписания (1)

Закрываем сеанс редактирования vi и сохраняем изменения, используя команду vi: *Esc : w q* (рис. 3.8)

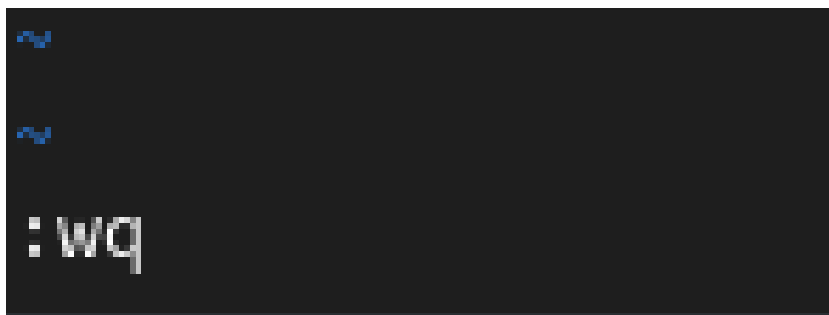


Рис. 3.8: Закрытие сеанса редактирование и сохранение изменений

Пояснения к синтаксису записи в crontab:

1. */1*: Это поле для минут. Значение */1* означает, что задача будет выполняться каждую минуту
2. *:* Поле для часов. Значение *:* означает, что задача будет выполняться каждый час
3. *:* Поле для дня месяца. *:* означает, что задача будет выполняться каждый день месяца
4. *:* Поле для месяца. *:* означает, что задача будет выполняться каждый месяц
5. *:* Поле для дня недели. *:* означает, что задача будет выполняться каждый день недели.
6. `logger This message is written from root cron.`: Это команда, которую нужно выполнить. `logger` - стандартная команда в Unix/Linux системах, которая пишет сообщения в системный журнал

В итоге эта запись crontab означает, что каждую минуту будет выполняться команда `logger "This message is written from root cron."`, которая запишет сообщение в системный журнал

Посмотрим список заданий в расписании с помощью `crontab -l`. В расписании появилась запись о запланированном событии (рис. 3.9)

```
[root@eavernikovskaya ~]# crontab -l
*/1 * * * * logger This message is written from root cron
[root@eavernikovskaya ~]#
```

Рис. 3.9: Список заданий после редактирования файла расписания (1)

Не выключая систему, через некоторое время (2–3 минуты) посмотрим журнал системных событий: `grep written /var/log/messages`. Мы видим что каждую минуту выполнялась команда `logger` “This message is written from root cron.”, которая каждую минуту записывала сообщение в системный журнал (рис. 3.10)

```
[root@eavernikovskaya ~]# grep written /var/log/messages
Oct 24 20:18:02 eavernikovskaya root[3098]: This message is written from root cron
Oct 24 20:19:01 eavernikovskaya root[3173]: This message is written from root cron
Oct 24 20:20:01 eavernikovskaya root[3259]: This message is written from root cron
Oct 24 20:21:02 eavernikovskaya root[3336]: This message is written from root cron
Oct 24 20:22:01 eavernikovskaya root[3420]: This message is written from root cron
[root@eavernikovskaya ~]#
```

Рис. 3.10: Журнал системных событий (1)

Далее меняем запись в расписании `crontab` на следующую: `0 /1 * 1-5 logger This message is written from root cron` (рис. 3.11)

```
root@eavernikovskaya:~
0 */1 * * 1-5 logger This message is written from root cron
~
~
```

Рис. 3.11: Редактирование файла расписания (2)

Пояснения к синтаксису записи в `crontab`:

1. 0: Это поле для минут. Значение 0 означает, что задача будет выполняться в начале каждого часа (в 00 минут)
2. /1: Поле для часов. Значение /1 означает, что задача будет выполняться каждый час
3. : Поле для дня месяца. Значение означает, что задача будет выполняться каждый день месяца
4. : Поле для месяца. Значение означает, что задача будет выполняться каждый месяц
5. 1-5: Поле для дня недели. 1-5 означает, что задача будет выполняться с понедельника по пятницу (1 - понедельник, 7 - воскресенье)

6. `logger This message is written from root cron.:` Это команда, которую нужно выполнить. `logger` - стандартная команда в Unix/Linux системах, которая пишет сообщения в системный журнал

В итоге эта запись `crontab` означает, что в начале каждого часа (00 минут) с понедельника по пятницу будет выполняться команда `logger "This message is written from root cron."`, которая запишет сообщение в системный журнал.

Снова посмотрим список заданий в расписании (рис. 3.12)

```
[root@eavernikovskaya ~]# crontab -l
0 */1 * * 1-5 logger This message is written from root cron
[root@eavernikovskaya ~]#
```

Рис. 3.12: Список заданий после редактирования файла расписания (2)

Переходим в каталог `/etc/cron.hourly` и создайте в нём файл сценария с именем `eachhour` (рис. 3.13)

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

Рис. 3.13: Переход в каталог `/etc/cron.hourly` и создание файла сценария с именем `eachhour`

Открываем файл `eachhour` для редактирования и прописываем в нём следующий скрипт (запись сообщения в системный журнал): (рис. 3.14), (рис. 3.15), (рис. 3.16)

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

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

```
[root@eavernikovskaya cron.hourly]# gedit eachhour
```

Рис. 3.14: Открытие файла сценария `eachhour`



Рис. 3.15: Файл сценария eachhour

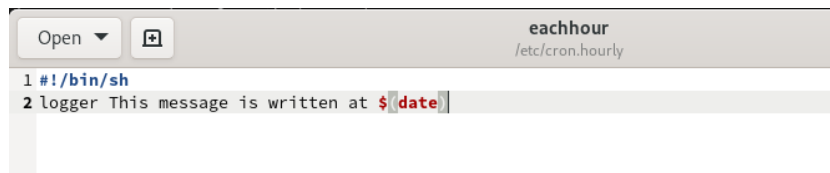


Рис. 3.16: Редактирование файла сценария eachhour

Делаем файл сценария eachhour исполняемым: `chmod +x eachhour` (рис. 3.17)

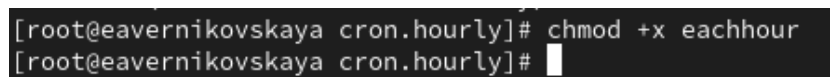


Рис. 3.17: Права на исполнение файла сценария eachhour

Теперь переходим в каталог `/etc/crond.d` и создаём в нём файл с расписанием eachhour (рис. 3.18)

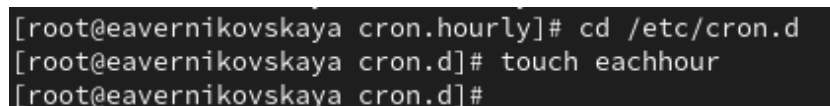


Рис. 3.18: Переход в каталог /etc/crond.d и создание файла с расписанием eachhour

Открываем этот файл для редактирования и помещаем в него следующее содержимое: `11 * * * * root logger This message is written from /etc/cron.d` (рис. 3.19), (рис. 3.20), (рис. 3.21)

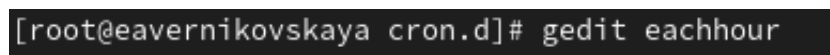


Рис. 3.19: Открытие файла с расписанием eachhour



Рис. 3.20: Файл с расписанием eachhour

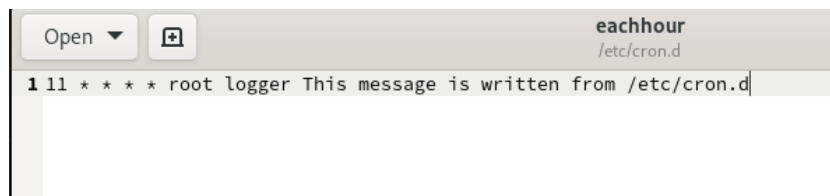


Рис. 3.21: Редактирование файла с расписанием eachhour

Пояснения к синтаксису записи:

1. 11: Это поле для минут. Значение 11 означает, что задача будет выполняться в 11 минут каждого часа
2. : Поле для часов. Значение * означает, что задача будет выполняться каждый час
3. : Поле для дня месяца. Значение * означает, что задача будет выполняться каждый день месяца
4. : Поле для месяца. Значение * означает, что задача будет выполняться каждый месяц
5. : Поле для дня недели. Значение * означает, что задача будет выполняться каждый день недели
6. root: Это поле для пользователя, от имени которого будет выполняться команда. В данном случае это суперпользователь root
7. logger This message is written from /etc/cron.d: Это команда, которую нужно выполнить. logger - стандартная команда в Unix/Linux системах, которая пишет сообщения в системный журнал

В итоге эта запись crontab означает, что каждую минуту с 11-й по 12-ю минуту

каждого часа будет выполняться команда logger “This message is written from /etc/cron.d” от имени суперпользователя root

Не выключая систему, через некоторое время (2–3 часа) посмотрим журнал системных событий. Мы видим что сообщение *This message is written from root cron* записывалось в журнал каждый час, а сообщение *This message is written from /etc/cron.d* записывалось в журнал каждую минуту с 11-ой по 12-ую каждого часа (рис. 3.22)

```
[root@eavernikovskaya cron.d]# grep written /var/log/messages
Oct 24 20:18:02 eavernikovskaya root[3098]: This message is written from root cron
Oct 24 20:19:01 eavernikovskaya root[3173]: This message is written from root cron
Oct 24 20:20:01 eavernikovskaya root[3259]: This message is written from root cron
Oct 24 20:21:02 eavernikovskaya root[3336]: This message is written from root cron
Oct 24 20:22:01 eavernikovskaya root[3420]: This message is written from root cron
Oct 24 20:23:01 eavernikovskaya root[3509]: This message is written from root cron
Oct 24 21:00:01 eavernikovskaya root[3872]: This message is written from root cron
Oct 24 21:01:01 eavernikovskaya root[3924]: This message is written at Thu Oct 24 09:01:01 PM MSK 2024
Oct 24 21:11:01 eavernikovskaya root[3986]: This message is written from /etc/cron.d
Oct 24 22:00:02 eavernikovskaya root[4351]: This message is written from root cron
Oct 24 22:01:01 eavernikovskaya root[4399]: This message is written at Thu Oct 24 10:01:01 PM MSK 2024
Oct 24 22:11:01 eavernikovskaya root[4445]: This message is written from /etc/cron.d
Oct 24 23:00:01 eavernikovskaya root[4539]: This message is written from root cron
Oct 24 23:01:01 eavernikovskaya root[4589]: This message is written at Thu Oct 24 11:01:01 PM MSK 2024
Oct 24 23:11:01 eavernikovskaya root[4600]: This message is written from /etc/cron.d
[root@eavernikovskaya cron.d]#
```

Рис. 3.22: Журнал системных событий (2)

3.2 Планирование заданий с помощью at

Проверяем, что служба atd загружена и включена: `systemctl status atd` (рис. 3.23)

```
[root@eavernikovskaya ~]# systemctl status atd
● atd.service - Deferred execution scheduler
   Loaded: loaded (/usr/lib/systemd/system/atd.service; enabled; preset: enabled)
   Active: active (running) since Thu 2024-10-24 20:09:35 MSK; 3h 30min ago
     Docs: man:atd(8)
    Main PID: 1080 (atd)
      Tasks: 1 (limit: 12205)
    Memory: 304.0K
       CPU: 4ms
    CGroup: /system.slice/atd.service
            └─1080 /usr/sbin/atd -f

Oct 24 20:09:35 eavernikovskaya.localdomain systemd[1]: Started Deferred execution scheduler.
[root@eavernikovskaya ~]#
```

Рис. 3.23: Проверка статуса службы atd

Задаём выполнение команды *logger message from at* в 23:43. Для этого вводим сначала *at 23:43*, а затем *logger message from at*. После нажимаем `ctrl+d` чтобы закрыть оболочку (рис. 3.24)

```
[root@eavernikovskaya ~]# at 23:43
warning: commands will be executed using /bin/sh
at> logger message from at
at> <EOT>
job 1 at Thu Oct 24 23:43:00 2024
[root@eavernikovskaya ~]#
```

Рис. 3.24: Задаём выполнение команды `logger message from at` в 23:43

Убедимся, что задание действительно запланировано с помощью `atq` (рис. 3.25)

```
[root@eavernikovskaya ~]# atq
1          Thu Oct 24 23:43:00 2024 a root
[root@eavernikovskaya ~]#
```

Рис. 3.25: Проверка (1)

С помощью команды `grep 'from at' /var/log/messages` посмотрим, появилось ли соответствующее сообщение в лог-файле в указанное нами время (рис. 3.26)

```
[root@eavernikovskaya ~]# grep 'from at' /var/log/messages
Oct 24 23:43:00 eavernikovskaya root[4737]: message from at
[root@eavernikovskaya ~]#
```

Рис. 3.26: Проверка (2)

4 Контрольные вопросы + ответы

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`

5 Выводы

В ходе выполнения лабораторной работы мы получили навыки работы с планировщиками событий cron и at.

6 Список литературы

1. Лабораторная работа №8 [Электронный ресурс] URL: https://esystem.rudn.ru/pluginfile.php/1000000/mod_resource/content/1/scheduling.pdf