

Шаблон отчёта по лабораторной работе

№8

Планировщики событий

Кхари Жекка Кализая Арсе

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 Выводы | 19 |
| Список литературы | 20 |

Список иллюстраций

| | | |
|------|--|----|
| 3.1 | терминал под пользователя root | 7 |
| 3.2 | статус демона crond | 8 |
| 3.3 | конфигурация файла /etc/crontab | 8 |
| 3.4 | список заданий в расписании | 9 |
| 3.5 | расписание | 9 |
| 3.6 | добавление строки | 10 |
| 3.7 | список заданий в расписании | 10 |
| 3.8 | журнал системных событий | 11 |
| 3.9 | изменение в файле crontab | 11 |
| 3.10 | список заданий в расписании | 12 |
| 3.11 | каталог /etc/cron.hourly и создание файла eachhour | 12 |
| 3.12 | файл eachhour | 13 |
| 3.13 | добавление строк в файл eachhour | 13 |
| 3.14 | права для запуска файла eachhour | 14 |
| 3.15 | файл eachhour в каталоге /etc/cron.d | 14 |
| 3.16 | добавление строки в файл /cron.d/eachhour | 15 |
| 3.17 | журнал системных событий | 15 |
| 3.18 | терминал | 16 |
| 3.19 | статус службы atd | 16 |
| 3.20 | команда logger | 17 |
| 3.21 | команды atq | 17 |
| 3.22 | журнал системных событий | 18 |

Список таблиц

1 Цель работы

Получение навыков работы с планировщиками событий cron и at.

2 Задание

1. Выполните задания по планированию задач с помощью crond (см. раздел 8.4.1).
2. Выполните задания по планированию задач с помощью atd (см. раздел 8.4.2).

3 Выполнение лабораторной работы

3.1 Планирование задач с помощью cron

Сначала я открыл терминал под пользователя root (рис. 3.1).

```
su -
```

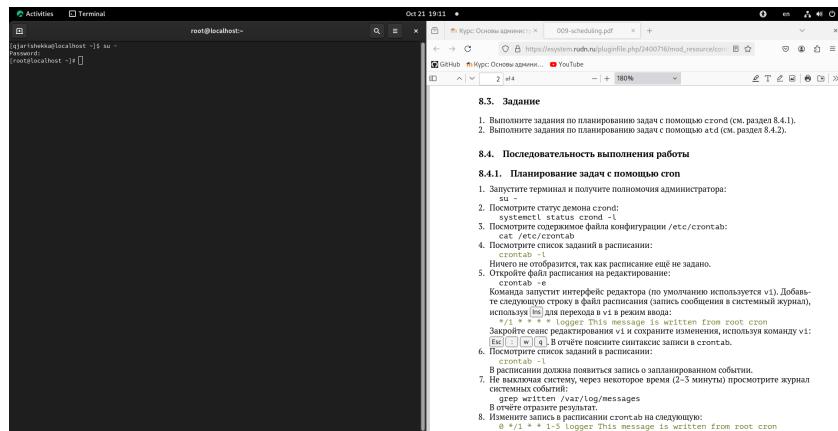


Рис. 3.1: терминал под пользователем root

Дальше я выполнил команду `systemctl` чтобы смотреть статус демона `crond` (рис. 3.2).

```
systemctl status crond -l
```

Рис. 3.2: статус демона crond

Потом я смотрел содержимое файла конфигурации /etc/crontab (рис. 3.3).

```
cat /etc/crontab
```

```
Activities Terminal Oct 21 19:11
root@localhost: ~
password:
Last login: Fri Oct 21 19:08:43 UTC 2022 on pts/0
[root@localhost ~]# systemctl status cron -l
● cron.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/cron.service; enabled; preset: enabled)
     Active: active (running) since Fri, 2024-10-21 19:05:43 MSK; 5min 52s ago
       Docs: man:crond(8)
   Main PID: 1 (wtmpd: 100025)
      Tasks: 1 (limit: 100025)
     Memory: 0 B
        CPU: 0ms
     CGroup: /proc/1/cgroup
           └─ 1 /sbin/wtmpd: wtmpd[100025]

Oct 21 19:05:43 localhost.localdomain systemd[1]: Started Command Scheduler.

Oct 21 19:05:43 localhost.localdomain cron[1227]: (CRON) INFO: No MTA installed, using 'mail' instead of 'sendmail.'
Oct 21 19:05:43 localhost.localdomain cron[1227]: (CRON) INFO: (deamon) delay will be used with factor 248 if used.
Oct 21 19:05:43 localhost.localdomain cron[1227]: (CRON) INFO: (deamon) delay will be used with factor 248 if used.
[root@localhost ~]# cat /etc/crontab
# /etc/crontab: system-wide cron job definitions
#
# This file defines the global cron daemon configuration.
# See /usr/share/doc/cron*/cron.conf for details.
#
# Example of job definitions:
# -----
# Minute Hour Day Month Year User Command
#----- ---- -- -- -- -
# * * * * * root    mail    whoami
# * * * * * root    mail    date
# * * * * * root    mail    ls -al
# * * * * * root    mail    /bin/true
# ----- day of week (0 - 6) (Sunday=0, Monday=1,...)
# ----- month (1 - 12) (January=1, February=2,...)
# ----- year (e.g., 0 for 2002, 10 for 2010)
# ----- user-name command to be executed

(crontab: executing crontab)

For details see /usr/share/doc/cron*/cron.conf

A. Example of job definitions:
-----
# Minute Hour Day Month Year User Command
#----- ---- -- -- --
# * * * * * root    mail    whoami
# * * * * * root    mail    date
# * * * * * root    mail    ls -al
# * * * * * root    mail    /bin/true
# ----- day of week (0 - 6) (Sunday=0, Monday=1,...)
# ----- month (1 - 12) (January=1, February=2,...)
# ----- year (e.g., 0 for 2002, 10 for 2010)
# ----- user-name command to be executed

(crontab: executing crontab) -r [q]
root@localhost: ~
```

Рис. 3.3: конфигурация файла /etc/crontab

Затем я посмотрел список заданий в расписании (рис. 3.4).

Рис. 3.4: список заданий в расписании

но команда не отобразил ничего потому что расписание еще не задано. Поэтому я открыл файл расписания с помощью команды crontab (рис. 3.5).

```
crontab -e
```

The screenshot shows a terminal window with the following content:

```
root@localhost:~# 009-scheduling.php
root@localhost:~# cd /etc/cron.d
root@localhost:~/cron.d# nano test
root@localhost:~/cron.d# cat
*/1 * * * * root echo "Hello world" > /tmp/cronout.log
root@localhost:~/cron.d# chmod +x test
root@localhost:~/cron.d# ls -l
total 0
root@localhost:~/cron.d# exit
```

Рис. 3.5: расписание

как только я запускал команду открылся текстовый редактор vim и там я добавил следующую строку (рис. 3.6).

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

```

root@localhost:~$ logger This message is written from root cron
root@localhost:~$ ./add_cronjob.sh
4. Просмотрите список заданий в расписании:
crontab -e
Ничего не отобразится, так как расписание ещё не задано.
5. Откройте файл расписания на редактирование:
crontab -e
Команда запустит интерфейс редактора (по умолчанию используется vi). Добавьте следующую строку в файл расписания (запись сообщения в системный журнал), используя [Esc] для перехода в в режим инвокации:
* * * * * logger This message is written from root cron
Закройте сеанс редактирования vi и сохраните изменения, используя команду vi:
Esc[ :wq! <Enter>
6. Просмотрите список заданий в расписании:
crontab -l
В расписании должна появиться запись о запланированном событии.
7. Не выключая систему, через некоторое время (2-5 минут) просмотрите журнал системных сообщений:
grep written /var/log/messages
В отчете появится синтаксис записи в cronjob.
8. Используйте команду crontab -e для просмотра списка cronjobов из следующего:
* * * * * logger This message is written from root cron
9. Просмотрите список заданий в расписании:
crontab -l
10. Передайте в каталог /etc/cron.hourly и создайте в нём файл скрипта с именем cronhour:
cd /etc/cron.hourly
touch cronhour
11. Откройте файл cronhour для редактирования и пропишите в нём следующий скрипт (запись сообщения в системный журнал):
#!/bin/sh
logger This message is written at $(date)
12. Создайте файл скрипта cronhour исполнением:
chmod +x cronhour
13. Теперь передайте в каталог /etc/cron.d и создайте в нём файл с расписанием cronhour:
cd /etc/cron.d

```

Рис. 3.6: добавление строки

я настроил запись в расписании чтобы записать в журнале задания каждую минуту Потом с помощью клавиш :wq я закрыл и сохранил файл.

Затем я выполнил комманду crontab чтобы смотреть список заданий в расписаний (рис. 3.7).

`crontab -l`

```

root@localhost:~$ crontab -l
4. Просмотрите список заданий в расписании:
crontab -e
Ничего не отобразится, так как расписание ещё не задано.
5. Откройте файл расписания на редактирование:
crontab -e
Команда запустит интерфейс редактора (по умолчанию используется vi). Добавьте следующую строку в файл расписания (запись сообщения в системный журнал), используя [Esc] для перехода в в режим инвокации:
* * * * * logger This message is written from root cron
Закройте сеанс редактирования vi и сохраните изменения, используя команду vi:
Esc[ :wq! <Enter>
6. Просмотрите список заданий в расписании:
crontab -l
В расписании должна появиться запись о запланированном событии.
7. Не выключая систему, через некоторое время (2-5 минут) просмотрите журнал системных сообщений:
grep written /var/log/messages
В отчете появится синтаксис записи в cronjob.
8. Используйте команду crontab -e для просмотра списка cronjobов из следующего:
* * * * * logger This message is written from root cron
9. Просмотрите список заданий в расписании:
crontab -l
10. Передайте в каталог /etc/cron.hourly и создайте в нём файл скрипта с именем cronhour:
cd /etc/cron.hourly
touch cronhour
11. Откройте файл cronhour для редактирования и пропишите в нём следующий скрипт (запись сообщения в системный журнал):
#!/bin/sh
logger This message is written at $(date)
12. Создайте файл скрипта cronhour исполнением:
chmod +x cronhour
13. Теперь передайте в каталог /etc/cron.d и создайте в нём файл с расписанием cronhour:
cd /etc/cron.d

```

Рис. 3.7: список заданий в расписании

Потом я ждал за 2 минуты и посмотрел журнал системных событий (рис. 3.8).

`grep written /var/log/messages`

Рис. 3.8: журнал системных событий

Потом я еще раз изменил запись в расписании crontab (рис. 3.9).

```
crontab -e
```

0 */1 * * 1-5 logger This message is written from root cron

The screenshot shows a terminal window titled 'root@localhost' and a browser window titled 'Курс: Основы администрирования'. The terminal contains the following text:

```
root@localhost:~# crontab -e
root@localhost:~# 0 * * * * /usr/bin/php /var/www/html/cronjob.php > /dev/null
```

The browser window displays a PHP page with the following content:

```
0 1 * * * /usr/bin/php /var/www/html/cronjob.php > /dev/null
```

Рис. 3.9: изменение в файле crontab

тогда я настроил запись в расписании чтобы записать в журнале за каждый час.

Сохранив и закрыв файл я посмотрел список заданий в расписании (рис. 3.10).

```
crontab -l
```

Рис. 3.10: список заданий в расписании

Потом я перешел в каталог /etc/cron.hourly и создал файл eachhour (рис. 3.11).

```
cd /etc/cron.hourly  
touch eachhour
```

Рис. 3.11: каталог /etc/cron.hourly и создание файла eachhour

Потом с помощью текстового редактора vim я открыл созданный файл (рис. .12).

vim eachhour

```

root@localhost:~# cat /etc/cron.hourly/eachhour
#!/bin/sh
logger This message is written at $(date)

```

Рис. 3.12: файл eachhour

Там я написал следующую строки (рис. 3.13):

```

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

```

```

root@localhost:~# cat /etc/cron.hourly/eachhour
#!/bin/sh
logger This message is written at $(date)

```

Рис. 3.13: добавление строк в файл eachhour

Потом я дал файлу права для запуска (рис. 3.14).

```
cdmod +x eachhour
```

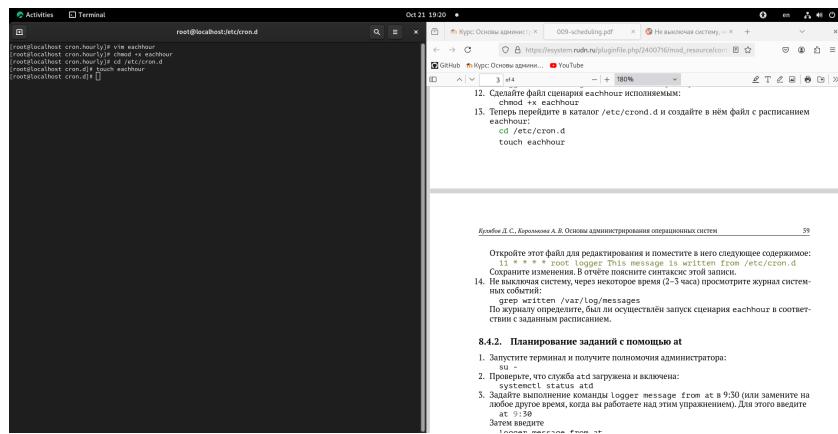


Рис. 3.14: права для запуска файла eachhour

Потом я перешел в другой каталог cron.d и там я создал файл eachhour (рис. 3.15).

```
cd /etc/cron.d
touch eachhour
```

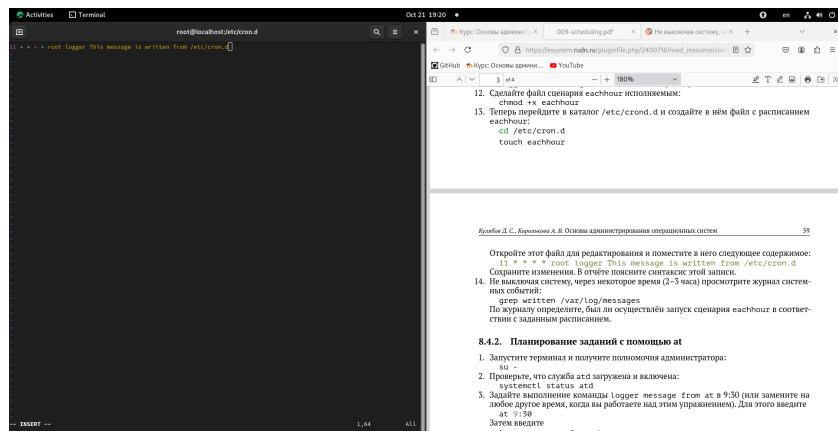


Рис. 3.15: файл eachhour в каталоге /etc/cron.d

и там я написал следующую строку (рис. 3.16):

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

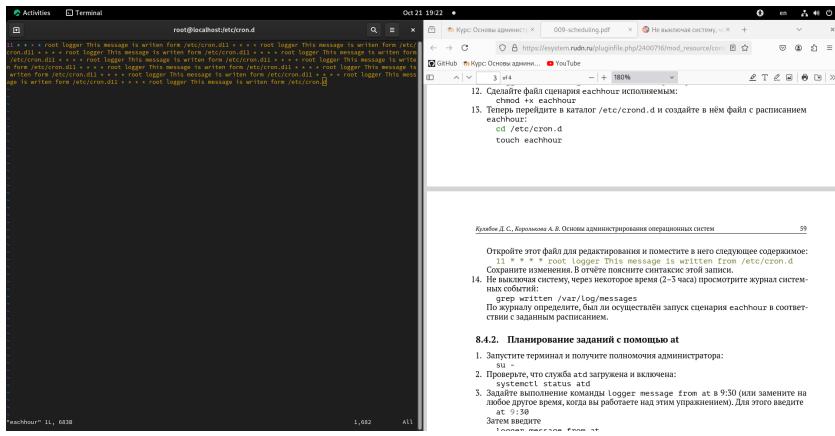


Рис. 3.16: добавление строки в файл /cron.d/eachhour

Эта строка создала еще 11 строк которая содержит ту же строку

Потом я ждал за 2 часа и выполнил следующий шаг чтобы смотреть журнал системных событий (рис. 3.17).

```
grep written /var/log/messages
```

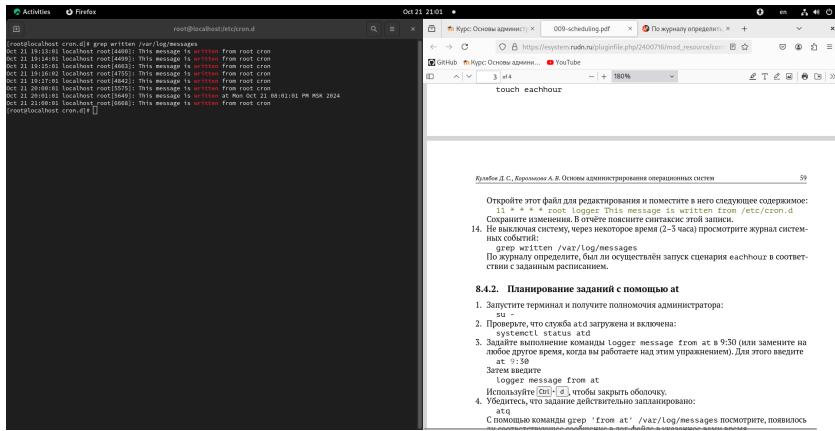


Рис. 3.17: журнал системных событий

3.2 Планирование заданий с помощью ат

здесь я использовал то же терминал (рис. 3.18).

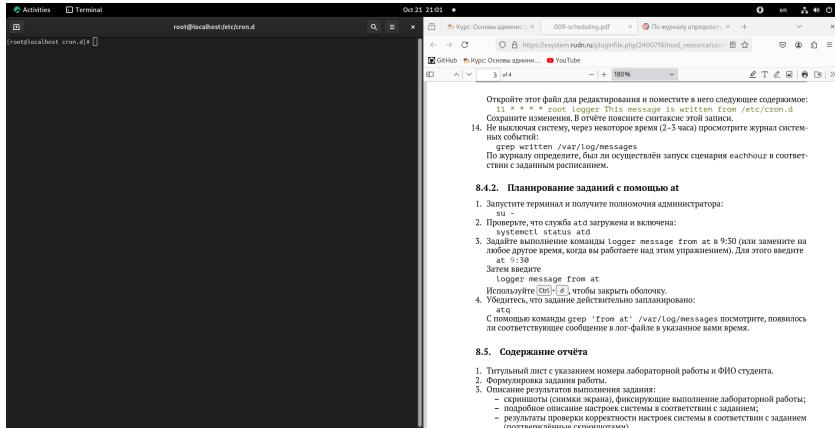


Рис. 3.18: терминал

Затем я выполнил команду `systemctl` чтобы смотреть статус службы `atd` (рис. 3.19).

```
systemctl status atd
```

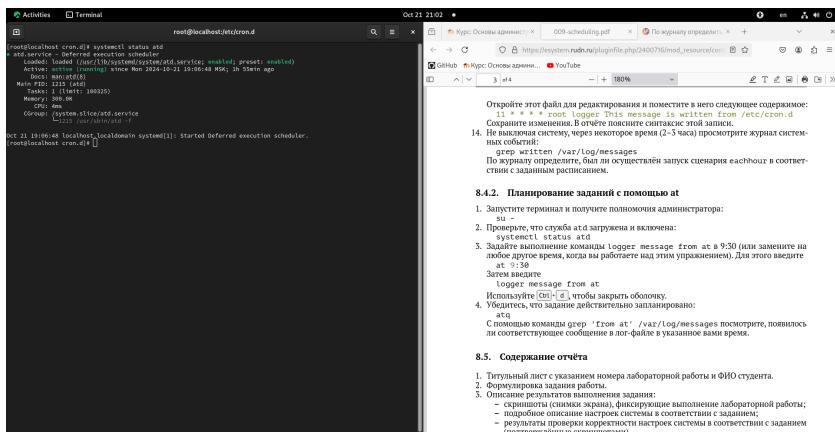


Рис. 3.19: статус службы atd

Потом я выполнил команду logger message from at и в конце я написал час в который я выполнил лабораторную работу (рис. 3.20).

logger message from at 7:00

```

root@localhost:~# logger message from at 9:00
root@localhost:~# atq

```

Определите этот файл для редактирования и поместите в него следующее содержимое:

```

13 * * * root logger This message is written from /etc/cron.d
Сохраните изменения. В отчёте поисните синтаксис этой записи.
14. Не выключая систему, через некоторое время (2-3 часа) просмотрите журнал системных сообщений:
grep written /var/log/messages
По журналу определите, был ли осуществлен запуск скрипта eashhour в соответствии с заданным расписанием.

```

8.4.2. Планирование задачий с помощью at

1. Запустите терминал и получите полномочия администратора:
su
2. Проверьте, что служба atd загружена и включена:
systemctl status atd
3. Задайте выполнение команды logger message from at 9:30 (или замените на любое другое время, когда мы работаем над этим упражнением). Для этого введите
at 9:30

Задайте месседж:

тот же message. Готов at

Используйте **(Ctrl+D)**, чтобы закрыть оболочку.
4. Убедитесь, что задание действительно запланировано:
atq

С помощью команды grep "Готов at" /var/log/messages посмотрите, появилось ли соответствующее сообщение в лог-файле в указанное вами время.

8.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулка задания работы.
3. Ответы на поставленные вопросы задания:
 - скриншоты (снимки экрана), фиксирующие выполнение лабораторной работы;
 - подробное описание настроек системы в соответствии с заданием;
 - результаты проверки корректности настроек системы в соответствии с заданием (приведите ссылки на скриншоты).

Рис. 3.20: команда logger

Затем я нажал клавиши Ctrl + d чтобы закрывать оболочку и потом я выполнил команду atq (рис. 3.21).

atq

```

root@localhost:~# logger message from at 9:00
root@localhost:~# atq

```

Определите этот файл для редактирования и поместите в него следующее содержимое:

```

13 * * * root logger This message is written from /etc/cron.d
Сохраните изменения. В отчёте поисните синтаксис этой записи.
14. Не выключая систему, через некоторое время (2-3 часа) просмотрите журнал системных сообщений:
grep written /var/log/messages
По журналу определите, был ли осуществлен запуск скрипта eashhour в соответствии с заданным расписанием.

```

8.4.2. Планирование задачий с помощью at

1. Запустите терминал и получите полномочия администратора:
su
2. Проверьте, что служба atd загружена и включена:
systemctl status atd
3. Задайте выполнение команды logger message from at 9:30 (или замените на любое другое время, когда мы работаем над этим упражнением). Для этого введите
at 9:30

Задайте месседж:

тот же message. Готов at

Используйте **(Ctrl+D)**, чтобы закрыть оболочку.
4. Убедитесь, что задание действительно запланировано:
atq

С помощью команды grep "Готов at" /var/log/messages посмотрите, появилось ли соответствующее сообщение в лог-файле в указанное вами время.

8.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулка задания работы.
3. Ответы на поставленные вопросы задания:
 - скриншоты (снимки экрана), фиксирующие выполнение лабораторной работы;
 - подробное описание настроек системы в соответствии с заданием;
 - результаты проверки корректности настроек системы в соответствии с заданием (приведите ссылки на скриншоты).

Рис. 3.21: команда atq

и в конце концов я выполнил команду grep чтобы смотреть журнал системных событий (рис. 3.22).

grep written /var/log/messages

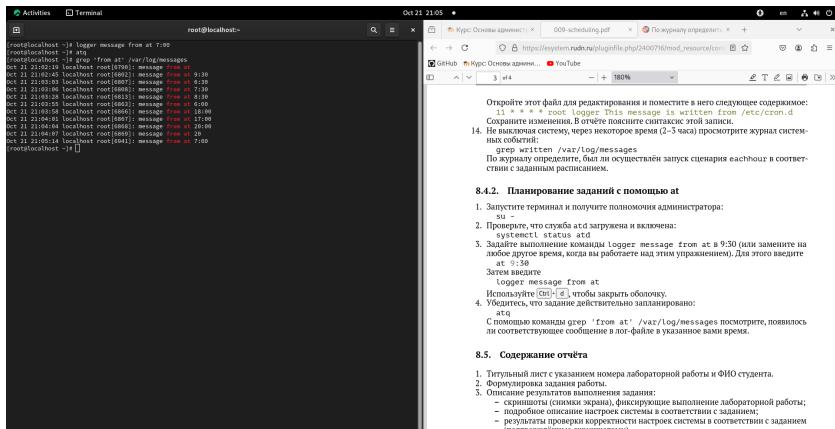


Рис. 3.22: журнал системных событий

здесь я смог просмотреть что расписание работает как я хотел там расписание по часам указано и отличается в чем что у него есть другой конец.

4 Выводы

Чтобы настроить расписание записи в журнале мы можем следовать эти шаги и сам изменить минуту час день как мы потребуем

Список литературы