Отчёт по второму разделу внешнего курса

Артём Дмитриевич Петлин

Содержание

1 Цель работы **5**

2 Задание **6**

3 Теоретическое введение **7**

4 Выполнение практических заданий **8**

5 Выполнение тестовых заданий **29**

6 Оценки тестов **51**

7 Выводы **66**

Список литературы **67**

Список иллюстраций

4.1 модуль 4 8
4.2 модуль 4 8
4.3 модуль 4 8
4.4 модуль 4 9
4.5 модуль 4 9
4.6 модуль 4 10
4.7 модуль 4 10
4.8 модуль 4 10
4.9 модуль 4 11
4.10 модуль 4 11
4.11 модуль 4 11
4.12 модуль 4 11
4.13 модуль 5 12
4.14 модуль 5 13
4.15 модуль 5 14
4.16 модуль 5 14
4.17 модуль 5 15
4.18 модуль 5 15
4.19 модуль 5 15
4.20 модуль 5 16
4.21 модуль 5 16
4.22 модуль 5 17
4.23 модуль 5 17
4.24 модуль 5 17
4.25 модуль 6 18
4.26 модуль 6 18
4.27 модуль 6 19
4.28 модуль 6 20
4.29 модуль 6 21
4.30 модуль 7 21
4.31 модуль 7 22
4.32 модуль 7 23
4.33 модуль 7 23
4.34 модуль 7 24
4.35 модуль 7 25
4.36 модуль 7 26

4.37 модуль 7 27
4.38 модуль 7 27
4.39 модуль 7 28
4.40 модуль 7 28

5.1 тест 29
5.2 тест 31
5.3 тест 32
5.4 тест 33
5.5 тест 34
5.6 тест 35
5.7 тест 36
5.8 тест 37
5.9 тест 38
5.10 тест 39
5.11 тест 40
5.12 тест 41
5.13 тест 42
5.14 тест 43
5.15 тест 44
5.16 тест 45
5.17 тест 46
5.18 тест 47
5.19 тест 48
5.20 тест 49

6.1 тест 1 51
6.2 тест 2 52
6.3 тест 3 53
6.4 тест 4 54
6.5 тест 5 55
6.6 тест 6 56
6.7 тест 7 57
6.8 тест 8 58
6.9 тест 9 59
6.10 тест 10 60
6.11 тест 11 61
6.12 тест 12 62
6.13 тест 13 63
6.14 тест 14 64
6.15 тест 15 65

Список таблиц

1 Цель работы

Выполнить второй раздел внешнего курса «Системный администратор Linux с нуля».

2 Задание

Задания четвертого, пятого, шестого и седьмого модулей, а также тесты.

3 Теоретическое введение

- Модуль 4. Получение справки. Использование справочных систем, работа с текстовыми файлами и логами
- Модуль 5. Управление пользователями и группами
- Модуль 6. Управление доступом
- Модуль 7. Управление процессами

4 Выполнение практических заданий

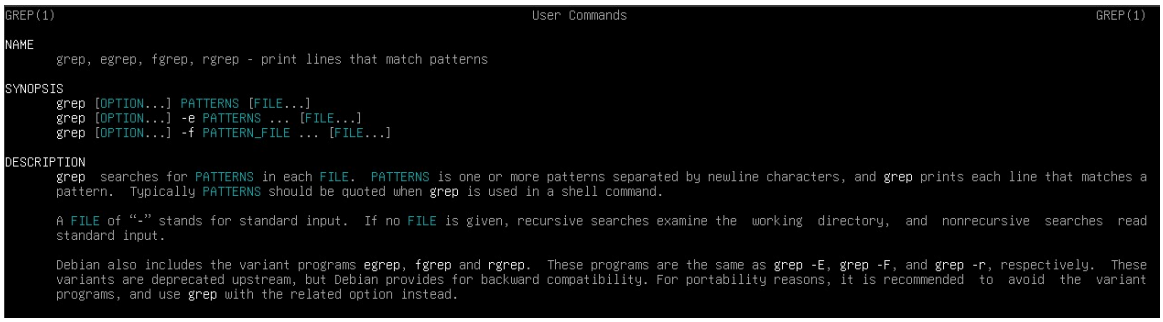

Рисунок 4.1: модуль 4

Используес man чтобы узнать, как работает команда grep.

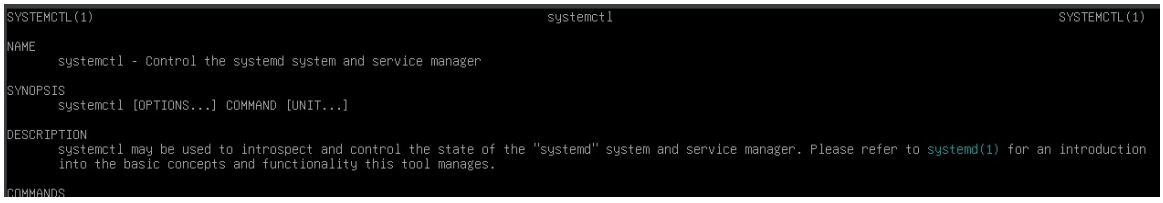

Рисунок 4.2: модуль 4

Находим документацию о systemctl с помощью info.

Рисунок 4.3: модуль 4

Откройте локальную документацию о info.

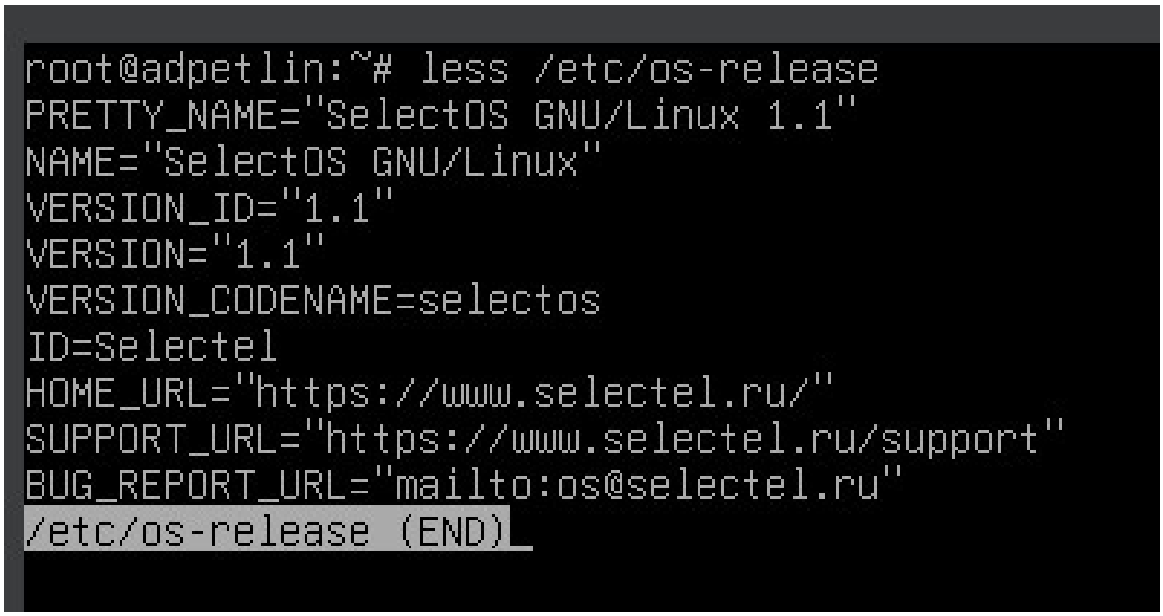

Рисунок 4.4: модуль 4

Открываем файл /etc/os-release с помощью less и находим название дистрибутива.

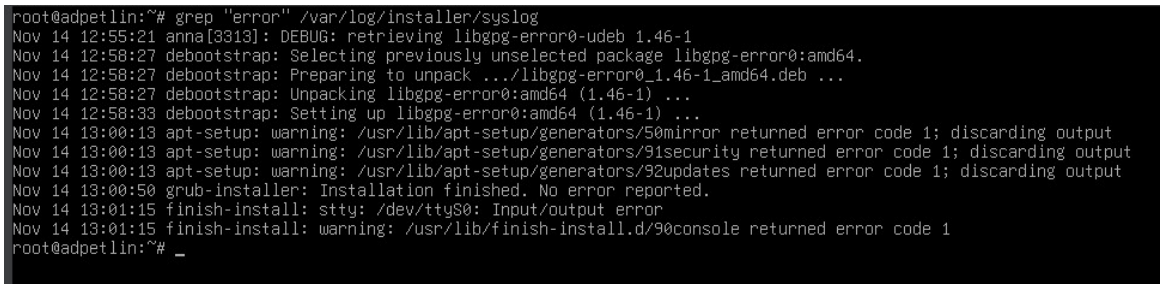

Рисунок 4.5: модуль 4

Исопльзуем grep, чтобы найти строки, содержащие «error» в файле /var/log/syslog.

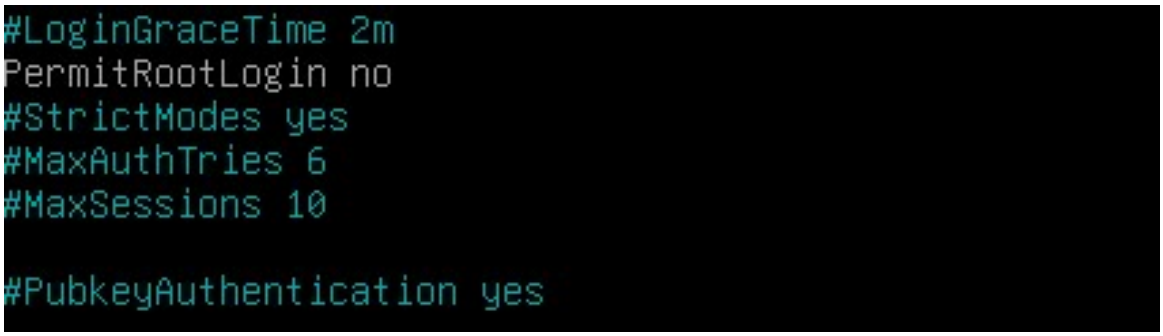

Рисунок 4.6: модуль 4

Отредактируем конфигурацию SSH с помощью nano или vim и изменяем параметр PermitRootLogin no.

Рисунок 4.7: модуль 4

Создаем небольшой файл с матрицей чисел произвольного размера, и поэкспериментируем с числами при помощи утилиты awk — например, посчитаем сумму чисел в каждой четной строке.

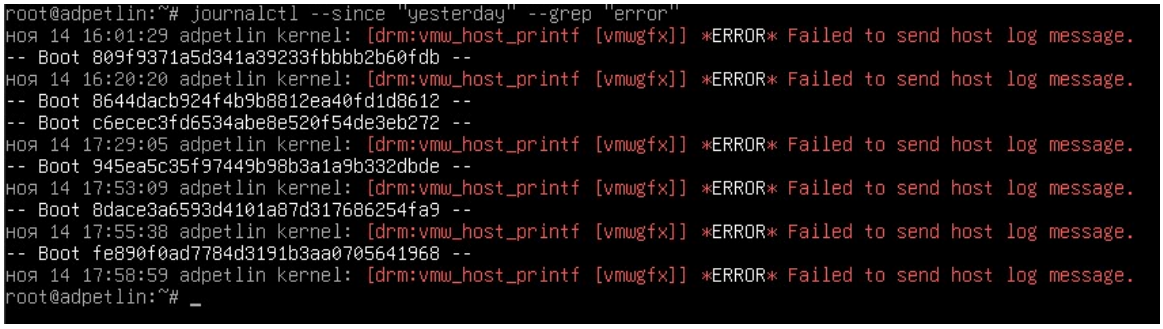

Рисунок 4.8: модуль 4

Находим все ошибки в системном журнале за последний день.

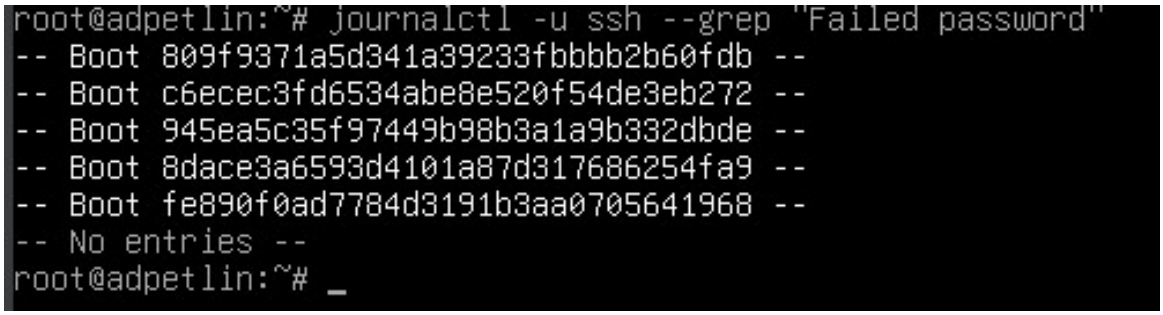

Рисунок 4.9: модуль 4

Проверяем логи SSH и находим неудачные попытки входа.

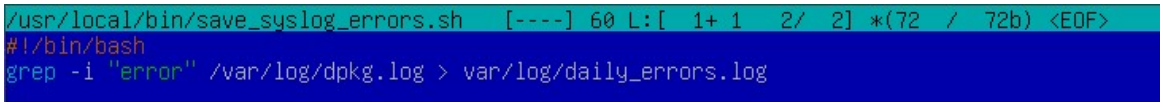

Рисунок 4.10: модуль 4

Пишем скрипт, который ежедневно сохраняет в файл все ошибки из /var/log/syslog.

Рисунок 4.11: модуль 4

Натстраиваем cron, чтобы скрипт выполнялся раз в день.

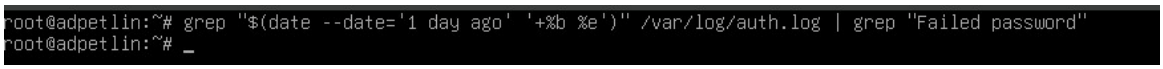

Рисунок 4.12: модуль 4

Используем grep, чтобы посчитать количество неудачных попыток входа за последние 24 часа.

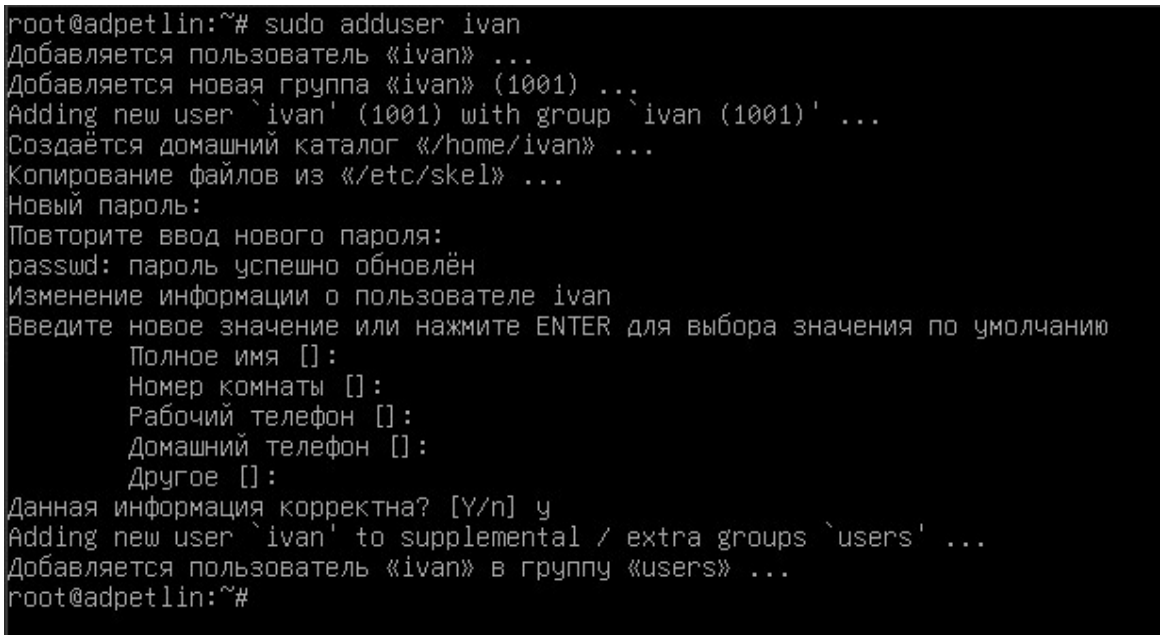

Рисунок 4.13: модуль 5

Создаем нового пользователя ivan и задаем ему пароль.

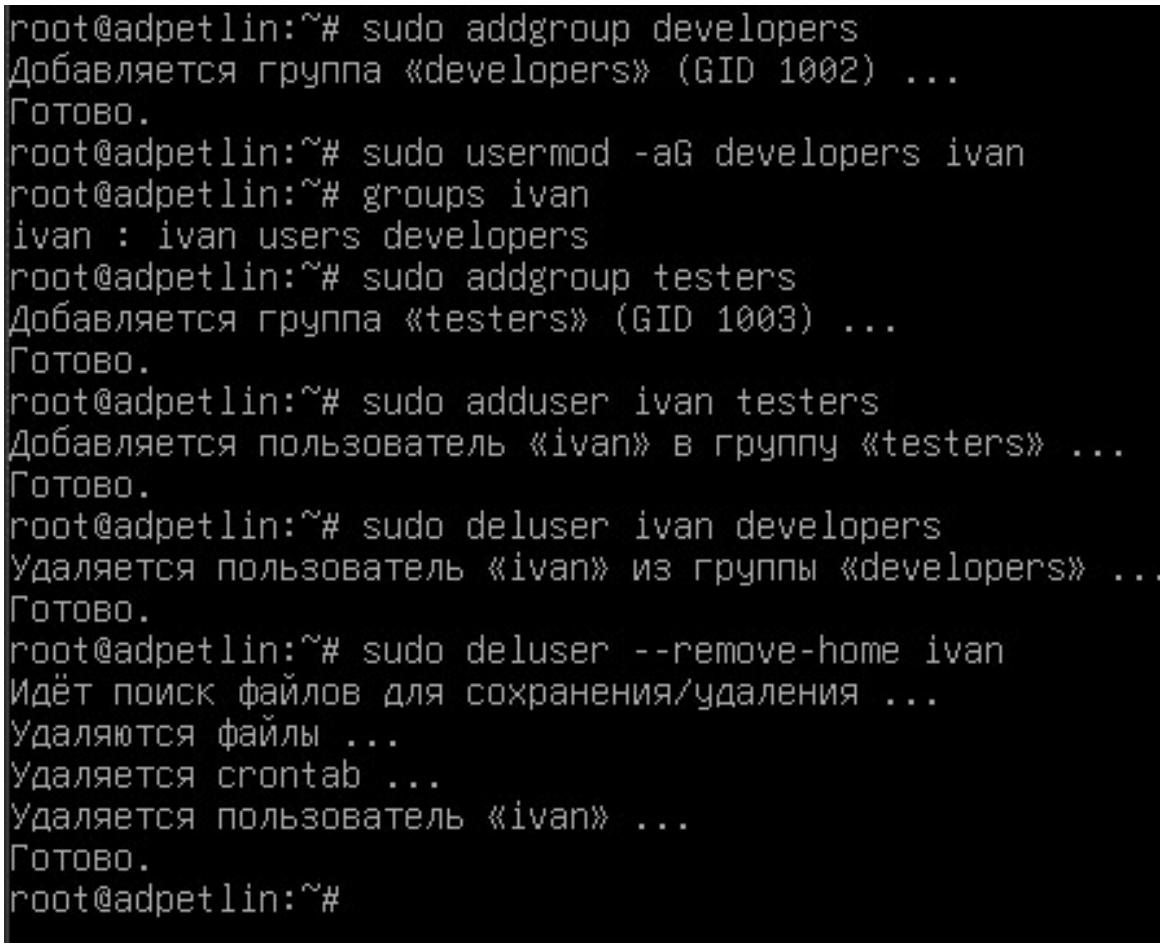

Рисунок 4.14: модуль 5

Создаем новую группу developers. Добавляем пользователя ivan в группу developers. Создаем группу testers и меняем принадлежность пользователя ivan группам с developers на testers. Удаляем учетную запись пользователя ivan вместе с его домашним каталогом.

```
root@adpetlin:~# touch sample.txt
root@adpetlin:~# ls -l sample.txt
-rw-r--r-- 1 root root 0 ноя 14 19:41 sample.txt
root@adpetlin:~# umask
0022
root@adpetlin:~#
```

Рисунок 4.15: модуль 5

Создаем где-нибудь новый файл. Изучаем разрешения, которые он получил автоматически. Имеют именно такие значения из-за umask.

```
root@adpetlin:~# touch sample.txt
root@adpetlin:~# ls -l sample.txt
-rw-r--r-- 1 root root 0 ноя 14 19:41 sample.txt
root@adpetlin:~# umask
0022
root@adpetlin:~# chmod g-w,o-r sample.txt
root@adpetlin:~# ls -l sample.txt
-rw-r----- 1 root root 0 ноя 14 19:41 sample.txt
root@adpetlin:~# chmod 644 sample.txt
root@adpetlin:~# ls -l
итого 16
drwxr-xr-x 2 root     root     4096 ноя 14 17:18 archieve_test
-rw-r--r-- 1 root     root      169 ноя 14 17:18 archieve_test.tar.gz
-rwxr-xr-x 1 adpetlin adpetlin  312 ноя 14 17:40 bin
drwxr-xr-x 3 root     root     4096 ноя 14 17:20 extracted
-rw-r--r-- 1 root     root        0 ноя 14 19:41 sample.txt
root@adpetlin:~# ls -l sample.txt
-rw-r--r-- 1 root root 0 ноя 14 19:41 sample.txt
root@adpetlin:~#
```

Рисунок 4.16: модуль 5

Для созданного файла убераем права на запись для группы и на чтение для всех остальных. Возвращаем первоначальные доступы файлу, используя числовую форму.

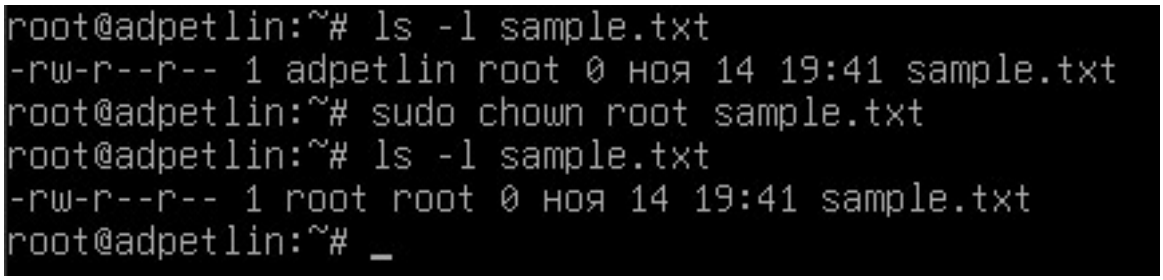

Рисунок 4.17: модуль 5

Делаем владельцем файла пользователя root.

Рисунок 4.18: модуль 5

Возвращаем себе владение файлом.

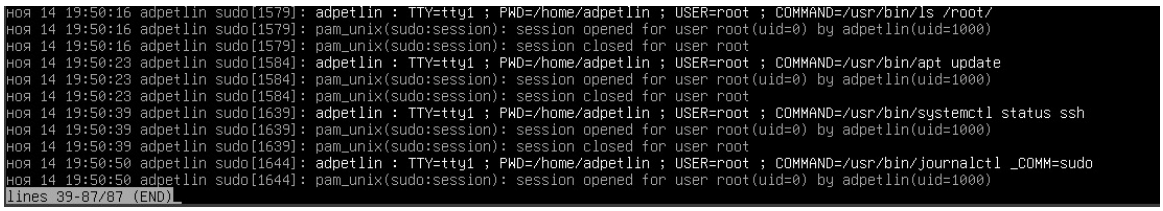

Рисунок 4.19: модуль 5

Выполняем несколько действий, используя sudo, и находим их в системном журнале.

```
adpetlin@adpetlin:~$ sudo -u ivan ls /root/
ls: невозможно открыть каталог '/root/': Отказано в доступе
adpetlin@adpetlin:~$ sudo -u ivan sudo ls /root/
[sudo] пароль для ivan:
ivan is not in the sudoers file.
adpetlin@adpetlin:~$ sudo usermod -aG sudo ivan
adpetlin@adpetlin:~$ sudo -u ivan sudo ls /root/
[sudo] пароль для ivan:
archieve_test  archieve_test.tar.gz  bin  extracted
adpetlin@adpetlin:~$
```

Рисунок 4.20: модуль 5

Пользуемся учетной записью ivan из предыдущих уроков. Проверяем, может ли он выполнять команды суперпользователя. Если нет, предоставляем ему такую возможность.

```
root@adpetlin:~# cat /etc/shadow
root:$y$j9T$qb0HnGKKNm589/BRB4fpr1$Niz7iJW0y8Z0C4w8om8hYsFW7eVPWaiuWk24z7HWjg8:20406:0:99999:7:::
daemon:*:20406:0:99999:7:::
bin:*:20406:0:99999:7:::
sys:*:20406:0:99999:7:::
sync:*:20406:0:99999:7:::
games:*:20406:0:99999:7:::
man:*:20406:0:99999:7:::
lp:*:20406:0:99999:7:::
mail:*:20406:0:99999:7:::
news:*:20406:0:99999:7:::
uucp:*:20406:0:99999:7:::
proxy:*:20406:0:99999:7:::
www-data:*:20406:0:99999:7:::
backup:*:20406:0:99999:7:::
list:*:20406:0:99999:7:::
irc:*:20406:0:99999:7:::
_apt:*:20406:0:99999:7:::
nobody:*:20406:0:99999:7:::
systemd-network:!*:20406::::::
sshd:!:20406::::::
messagebus:!:20406::::::
tcpdump:!:20406::::::
adpetlin:$y$j9T$X7v3oGk07bTUpneC1BNy61$gzDEV.dfv62ASy/wQmxmHwmzRsYz5XAr9qfWU8IQX02:20406:0:99999:7:::
ivan:$y$j9T$CXwOgObgqVavRGDmCDY9S1$mBiBLzSiaNtRMTboMlgSOsjL9cfvhunV0BXNrWeE3J5:20406:0:99999:7:::
root@adpetlin:~# awk -F: '$2 ~ /^\$6\$/ {print $1}' /etc/shadow
root@adpetlin:~# _
```

Рисунок 4.21: модуль 5

Выводим имена всех пользователей, у которых в качестве алгоритма хеширования пароля указан устаревший SHA-512 (его префикс 6).

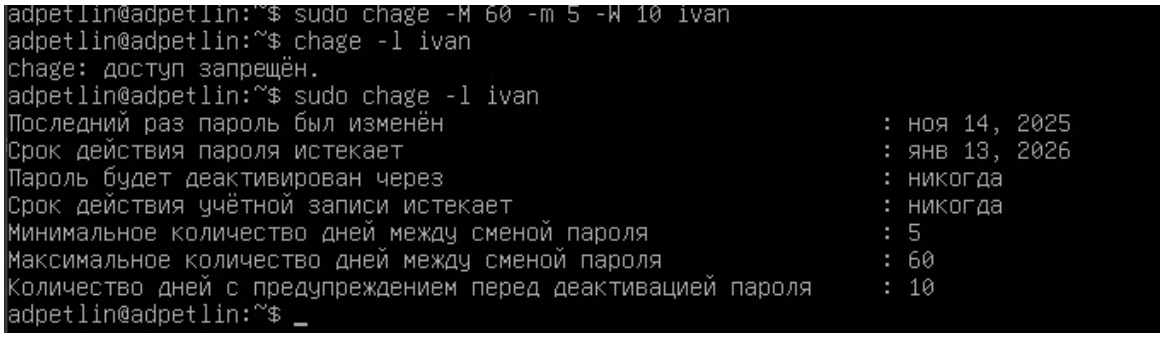

Рисунок 4.22: модуль 5

Создаем пользователя ivan. Устанавливаем для него нужные параметры политики паролей с помощью команды chage. Блокируем вход пользователю ivan.

Рисунок 4.23: модуль 5

Убеждаемся, что он не может войти в систему.

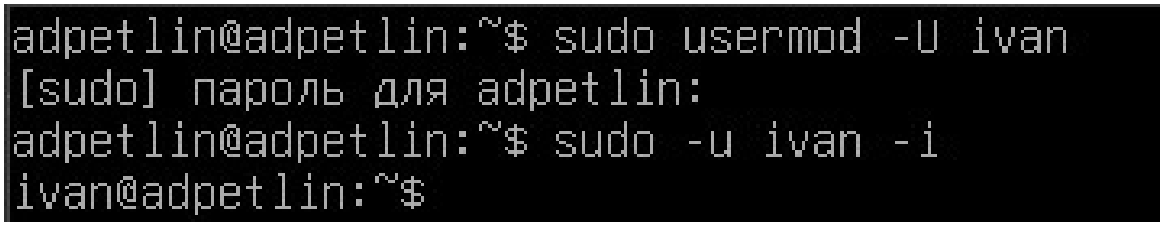

Рисунок 4.24: модуль 5

Возрвращаем пользователю ivan возможность авторизоваться.

Рисунок 4.25: модуль 6

Проверяем права доступа к /etc/passwd, /home и /var/log.

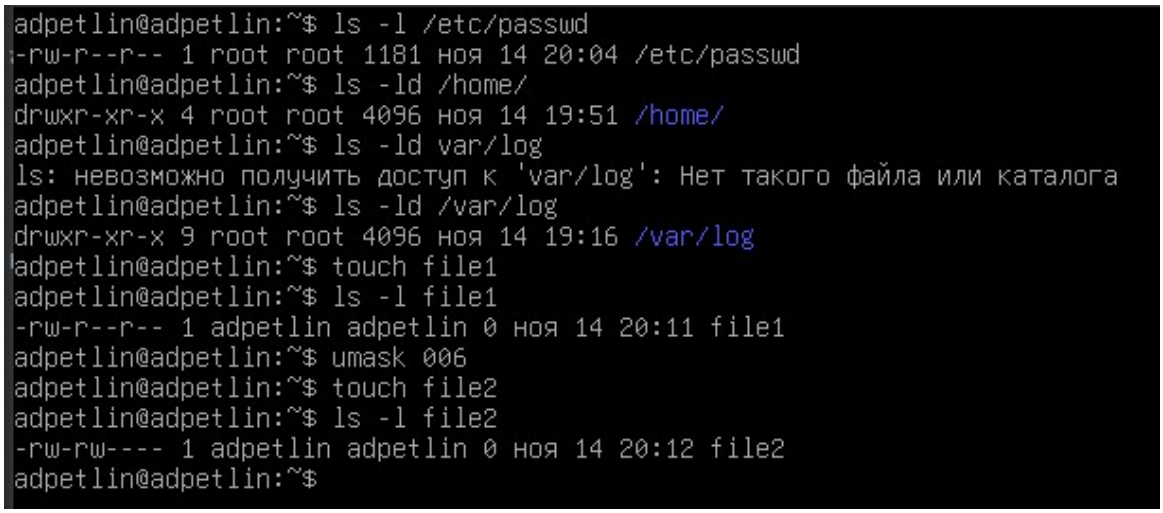

Рисунок 4.26: модуль 6

Создаем файл в домашнем каталоге, проверяем его права и настройте umask, чтобы у новых файлов не было прав для категории остальных пользователей.

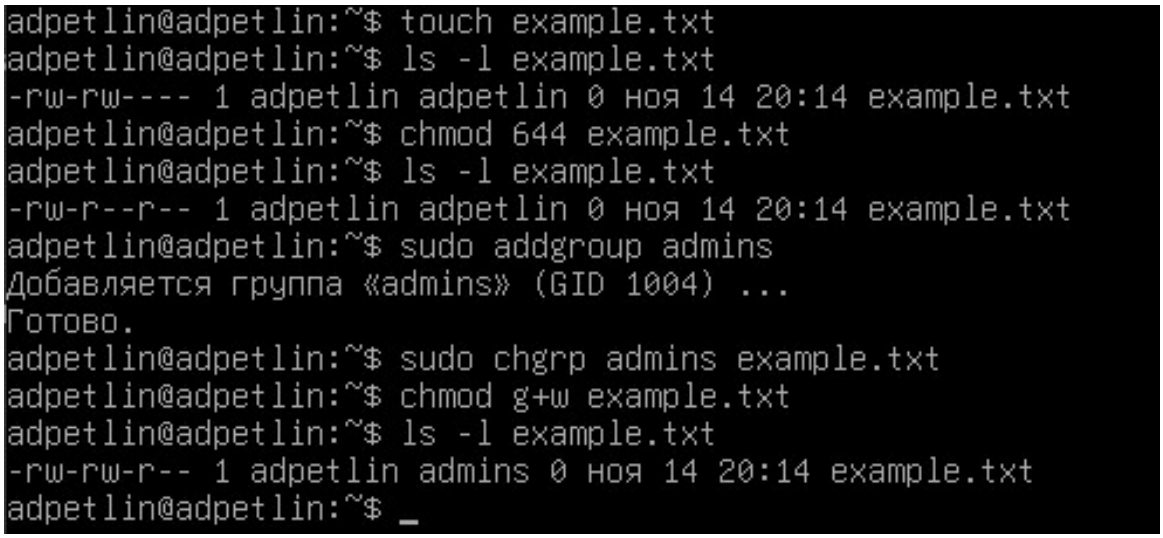

Рисунок 4.27: модуль 6

Создаем файл и настройте ему права 644, чтобы владелец мог редактировать, а остальные — только читать. Создаем группу admins, меняем группу у созданного файла в первом задании на эту и передаем группе право на изменение (запись).

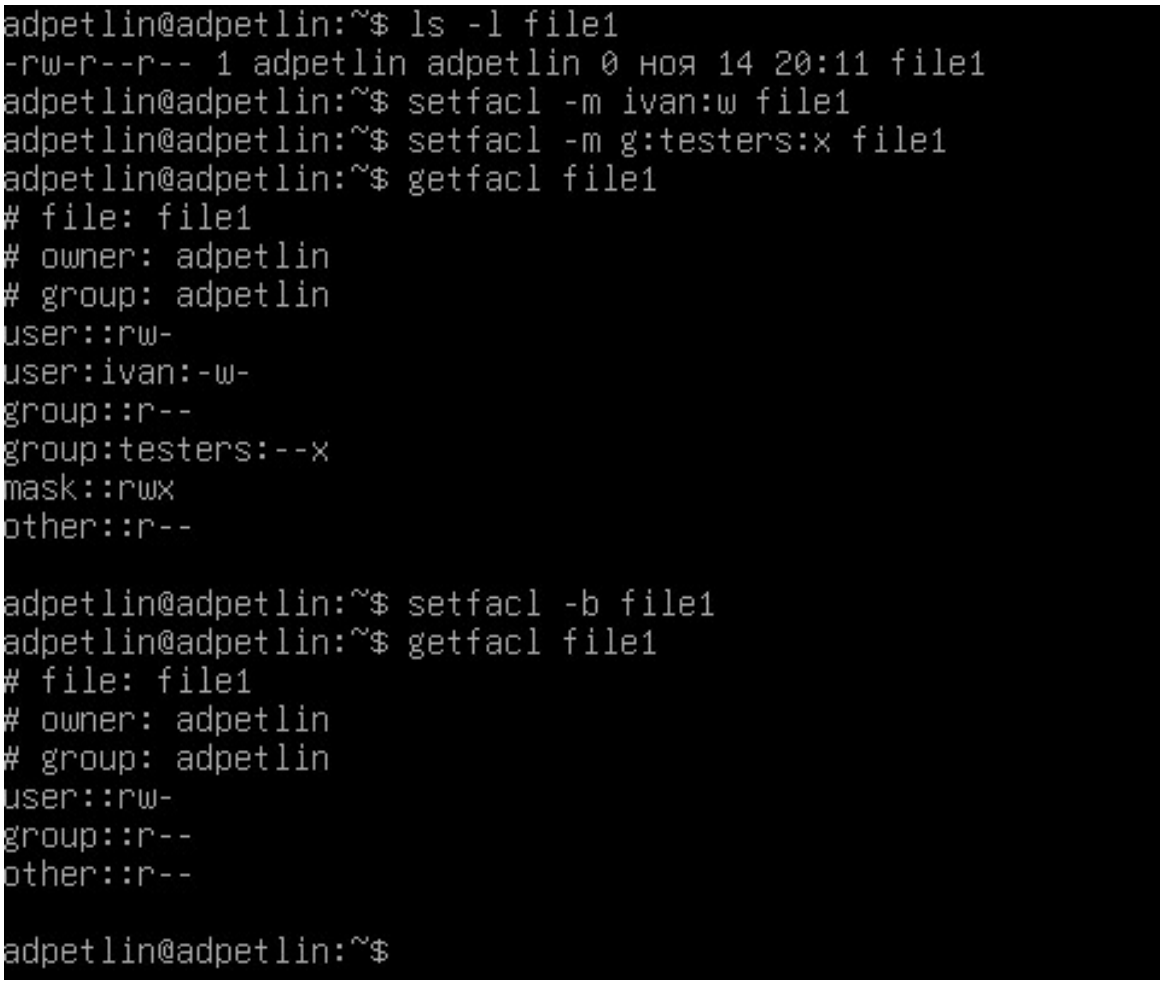

Рисунок 4.28: модуль 6

Разрешаем конкретному пользователю изменять файл, не добавляя его в основную группу владельца. Утсанавливаем особые права для группы, чтобы она могла только выполнять файл. Проверяем и сбрасываем все ACL с файла.

```
adpetlin@adpetlin:~$ chmod u+x file1
adpetlin@adpetlin:~$ chmod u+s file1
adpetlin@adpetlin:~$ mkdir directory1
adpetlin@adpetlin:~$ chmod g+x directory1/
adpetlin@adpetlin:~$ chmod g+s directory1/
adpetlin@adpetlin:~$ mkdir public_folder
adpetlin@adpetlin:~$ chmod o=wx public_folder/
adpetlin@adpetlin:~$ chmod +t public_folder/
adpetlin@adpetlin:~$ ls -l file1
-rwsr--r-- 1 adpetlin adpetlin 0 ноя 14 20:11 file1
adpetlin@adpetlin:~$ ls -l directory1/
итого 0
adpetlin@adpetlin:~$ ls -ld directory1/
drwxrws--x 2 adpetlin adpetlin 4096 ноя 14 20:21 directory1/
adpetlin@adpetlin:~$ ls -ld public_folder/
drwxrwx-wt 2 adpetlin adpetlin 4096 ноя 14 20:22 public_folder/
adpetlin@adpetlin:~$
```

Рисунок 4.29: модуль 6

Устанавливаем SUID на исполняемый файл, чтобы он запускался от имени владельца. Настраиваем SGID для каталога, чтобы новые файлы наследовали группу. Делаем так, чтобы категория других пользователей могли записывать в /public_folder, но не могли удалять чужие файлы.

```
PID to signal/kill [default pid = 252] 2102_
    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    252 root      20   0   65736  23388  18204 S   0,0   0,6   0:00.16 systemd-
      1 root      20   0  167516  12084   9156 S   0,0   0,3   0:00.60 systemd
    552 root       0 -20   10160   9024   3728 S   0,0   0,2   0:00.13 atop
    557 root      20   0   15400   8784   7552 S   0,0   0,2   0:00.00 sshd
    543 root      20   0   24884   7740   6756 S   0,0   0,2   0:00.05 systemd-
    277 root      20   0   26592   6108   4604 S   0,0   0,2   0:00.06 systemd-
   2103 adpetlin  20   0   11648   5492   3348 R   0,0   0,1   0:00.01 top
   1837 adpetlin  20   0    8248   5296   3692 S   0,0   0,1   0:00.16 bash
    539 message+  20   0    7912   3752   3384 S   0,0   0,1   0:00.02 dbus-dae
    450 root      20   0    5740   3676   2852 S   0,0   0,1   0:00.00 dhclient
   1832 root      20   0    4260   2936   2500 S   0,0   0,1   0:00.02 login
    538 root      20   0    6608   2712   2464 S   0,0   0,1   0:00.01 cron
    895 root      20   0    5876   1032    944 S   0,0   0,0   0:00.00 agetty
   2102 adpetlin  20   0    5464    904    816 S   0,0   0,0   0:00.00 sleep
```

Рисунок 4.30: модуль 7

Находим процесс, потребляющий больше всего памяти и завершаем его.

Рисунок 4.31: модуль 7

Запускаем процесс в фоновом режиме, возвращаем его на передний план, приостанавливаем процесс, затем возвращаем в фон.

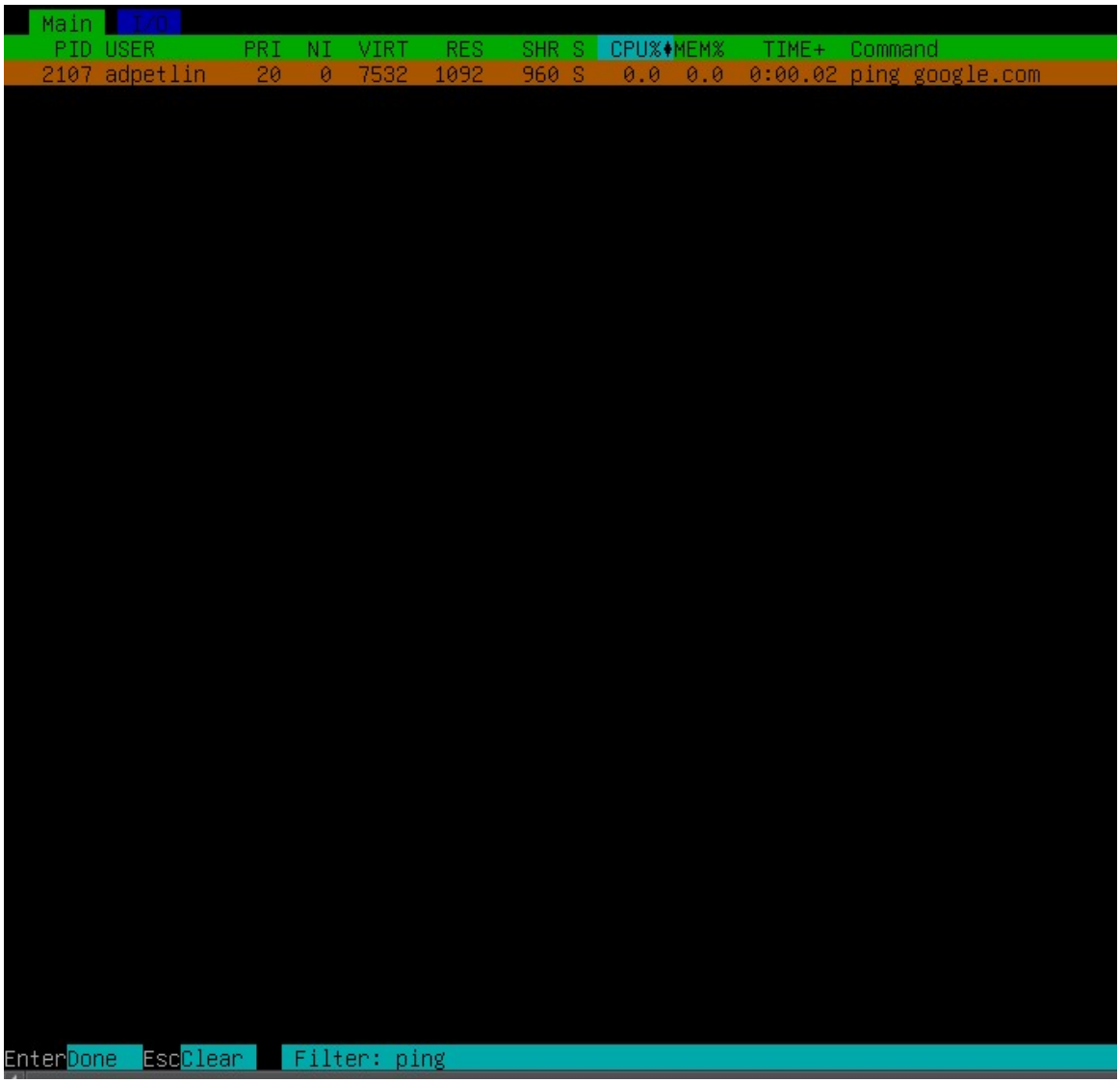

Рисунок 4.32: модуль 7

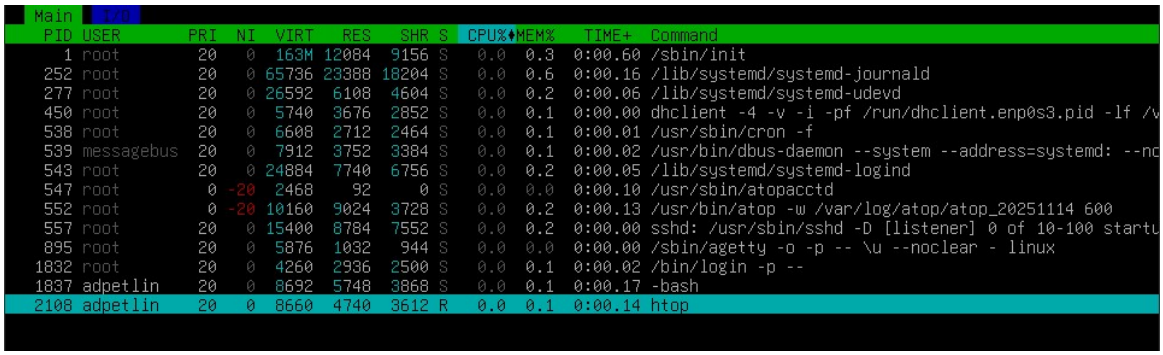

Рисунок 4.33: модуль 7

Устанавилваем и запускаем htop, настраиваем нужные вам колонки, находим по фильтру процесс и «убиваем его».

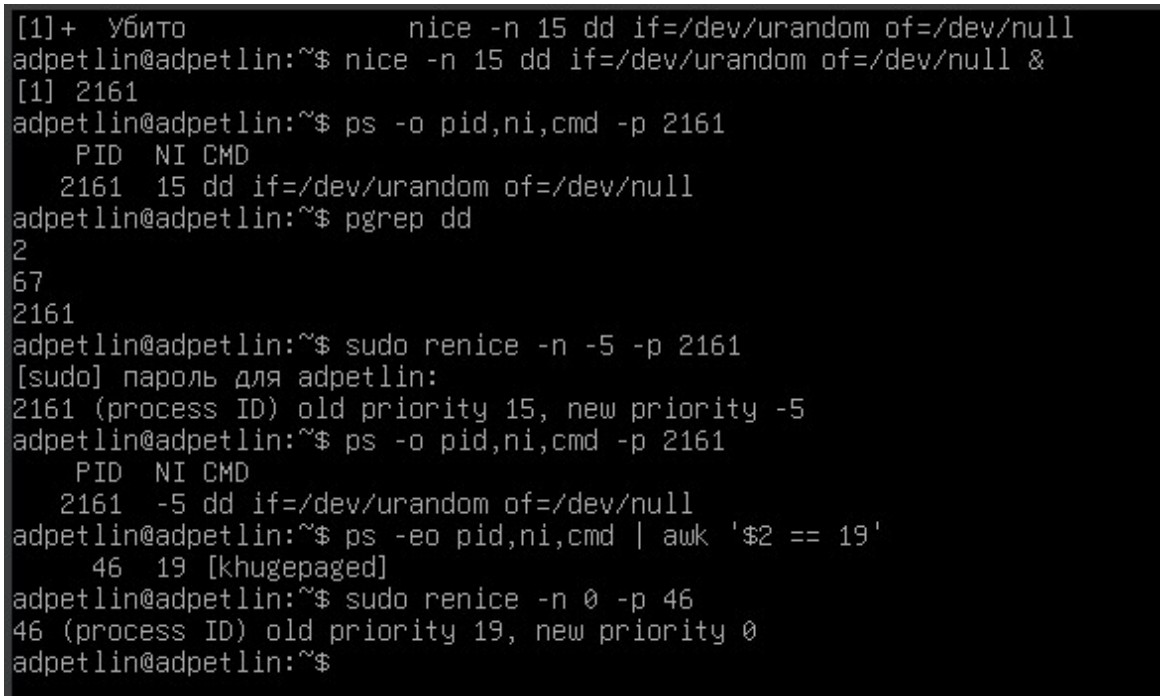

Рисунок 4.34: модуль 7

Запускаем задачу у низким приоритетом — например, с nice=15. Находим PID процесса, который запускали ранее, и повышаем приоритет до -5. Находим процесс с самым низким приоритетом и пробуем повысить приоритет данного процесса.

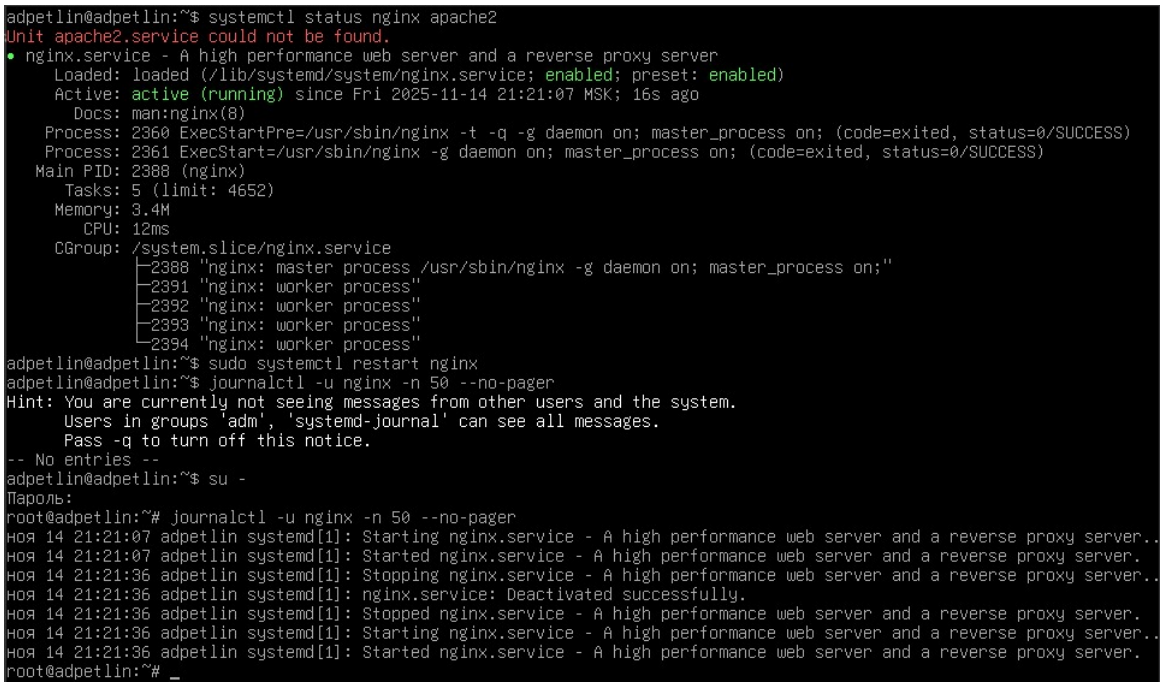

Рисунок 4.35: модуль 7

Выводим список всех активных сервисов и фильтруем их по фильтру «Network». Определяем, какой веб-сервер установлен, проверяем статус сервиса и перезапускаем его. После перезапуска проверяем журнал на наличие ошибок.

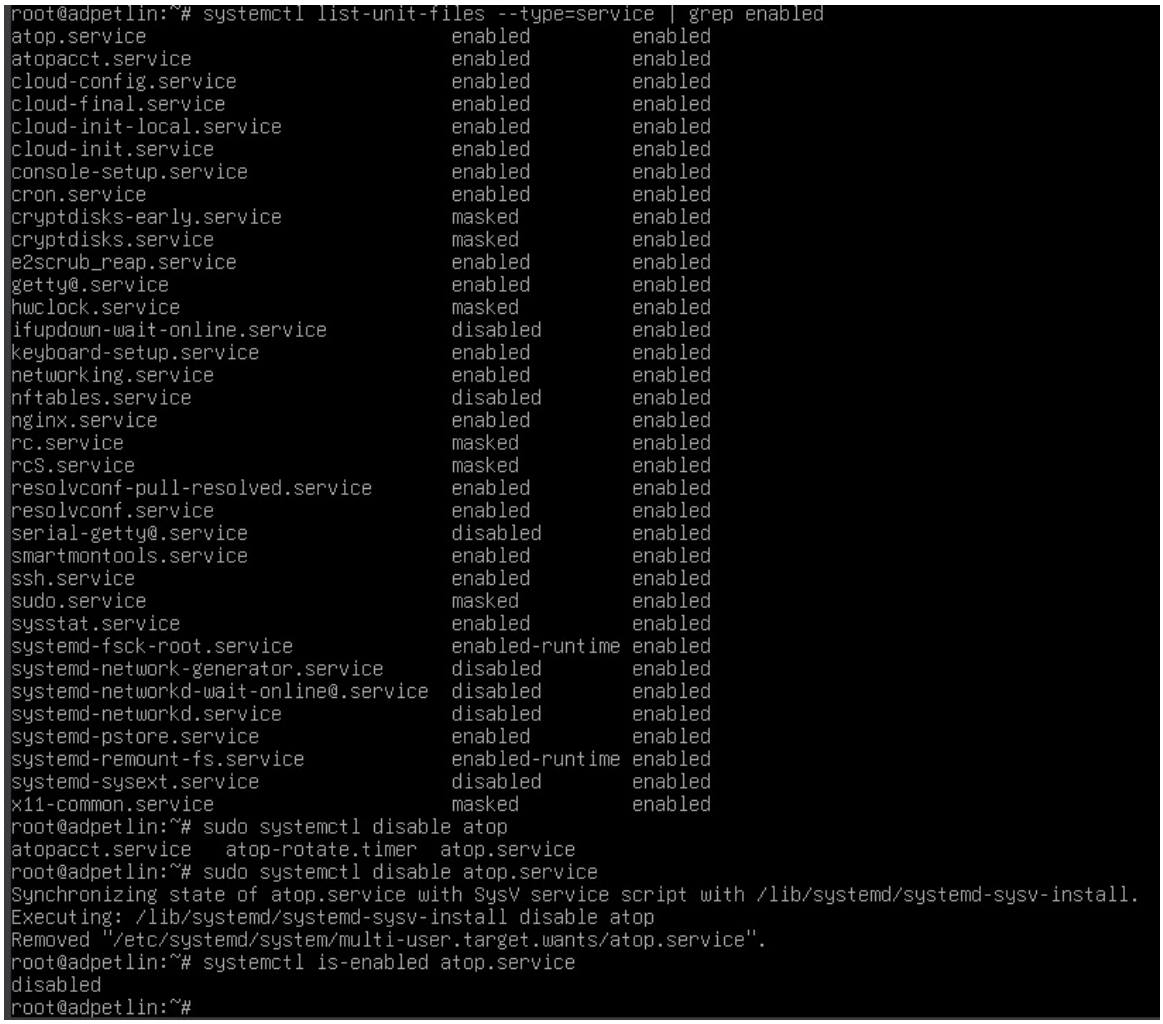

Рисунок 4.36: модуль 7

Находим ненужный сервис. Делаем так, чтобы при перезапуске системы данный процесс не запускался. Перезагружаем устройство, на котором работаете, и проверяем, что выбранный сервис выключен.

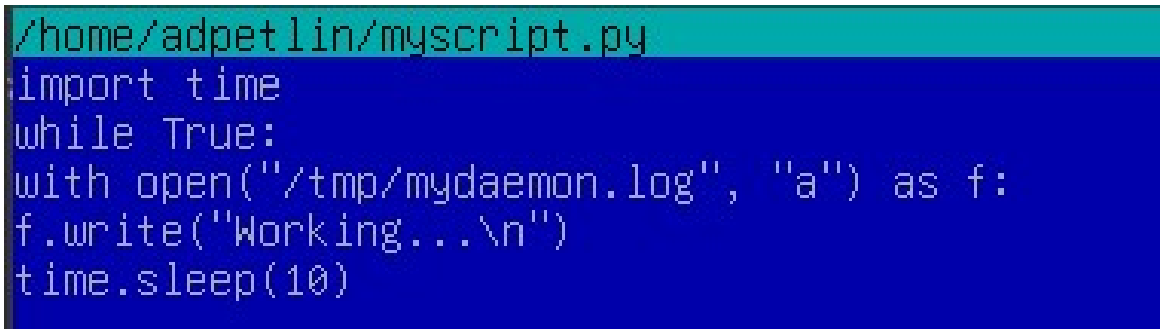

Рисунок 4.37: модуль 7

Создаем простой Python-скрипт и даем данному файлу права на запуск.

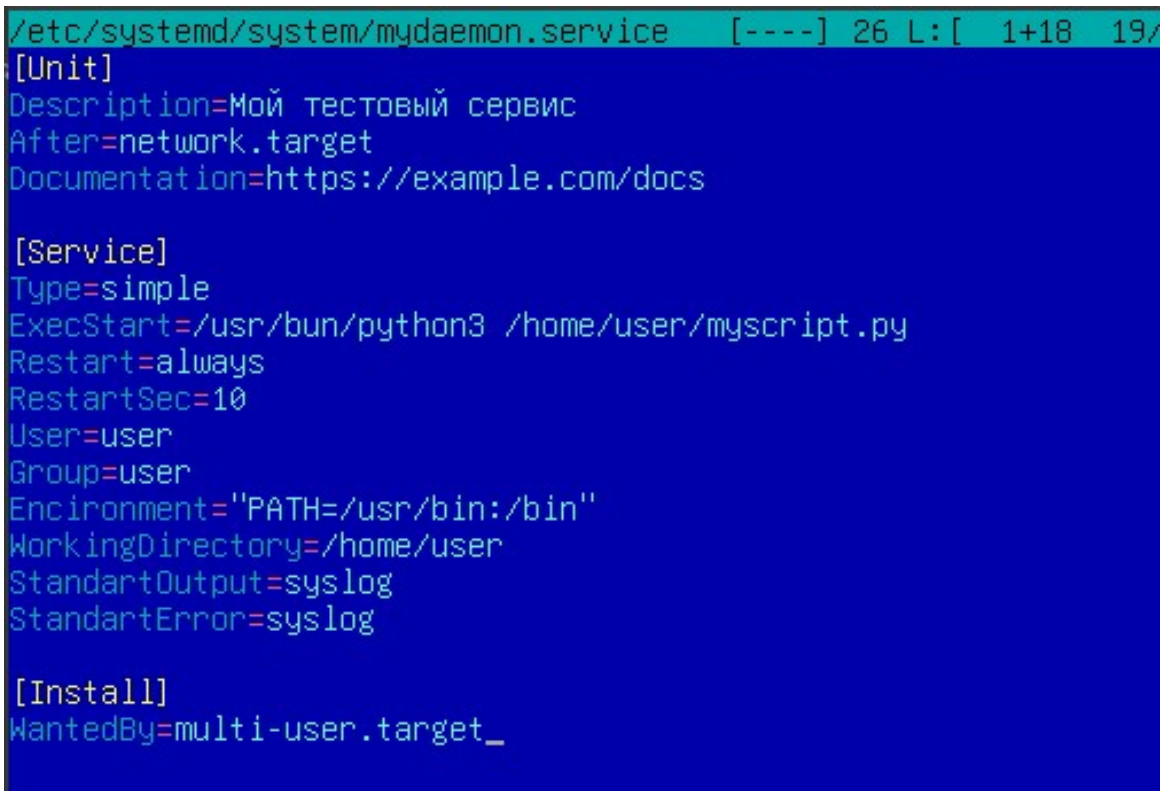

Рисунок 4.38: модуль 7

Создаем unit-файл — активируем и проверяем статус вашего юнита.

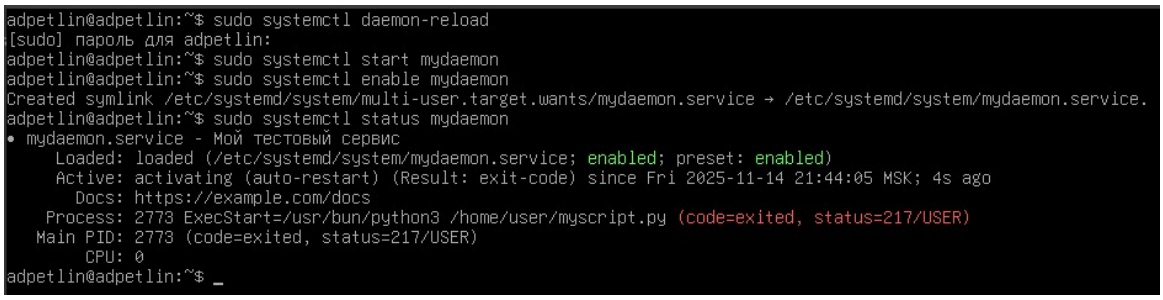

Рисунок 4.39: модуль 7

Принудительно завершаем процесс, который был запущен в предыдущем задании, и проверяем статус сервиса.

Рисунок 4.40: модуль 7

Перезагружаем устройство, на котором работаем, и проверяем, сработал ли автозапуск вашего процесса.

5 Выполнение тестовых заданий

Какая команда поможет узнать, как работает утилита grep?

а) grep --info

б) man grep

в) grep /?

г) info man

Верный ответ: man grep

Что делает команда info?

а) Показывает список всех установленных программ

б) Открывает справку в формате info для заданной команды

в) Проверяет системные обновления

г) Запускает файловый менеджер

Верный ответ: Открывает справку в формате info для заданной команды

Где чаще всего находятся текстовые справочные файлы (документация) к установленным программам в Linux?

а) /etc/configs

б) /var/log/info

в) /usr/share/doc

г) /bin/documents

Верный ответ: /usr/share/doc

Рисунок 5.1: тест

Когда нам нужно получить полное и структурированное описание утилиты, ее опций и примеров использования, мы обращаемся к встроенному справочнику с помощью команды man Мы используем команду info как альтернативу man для просмотра более подробной документации. Мы знаем, что после установки пакета его дополнительная документация, лицензии и примеры конфигураций обычно размещаются в каталоге /usr/share/doc/имя_пакета.

Что делает команда cat > файл.txt?

а) Показывает содержимое файла

б) Объединяет файлы

в) Создает новый файл и записывает в него

г) Добавляет строку в конец файла

Верный ответ: Создает новый файл и записывает в него

Чем отличается cat >> файл.txt от cat > файл.txt?

а) >> удаляет файл полностью, > — помещает в корзину

б) >> дописывает в конец, > — перезаписывает

в) >> копирует файл, , > — вырезает

г) Нет отличий

Верный ответ: >> дописывает в конец, > — перезаписывает

Чем less отличается от cat при просмотре больших файлов?

а) Нет принципиальных отличий

б) less не поддерживает поиск по содержимому

в) less показывает содержимое файла постранично с навигацией и поиском

г) less работает только с бинарными файлами

Верный ответ: less показывает содержимое файла постранично с навигацией и поиском

Рисунок 5.2: тест

Мы используем конструкцию cat > файл.txt для перенаправления стандартного ввода в новый файл. Мы применяем оператор », когда нужно добавить данные в конец существующего файла, не стирая его предыдущее содержимое. Оператор > всегда создает файл заново или полностью перезаписывает существующий. Для просмотра больших файлов мы предпочитаем использовать less, так как он позволяет нам листать

содержимое страницами, искать по тексту и свободно перемещаться по файлу, в то время как cat вываливает всё содержимое сразу, что неудобно для навигации.

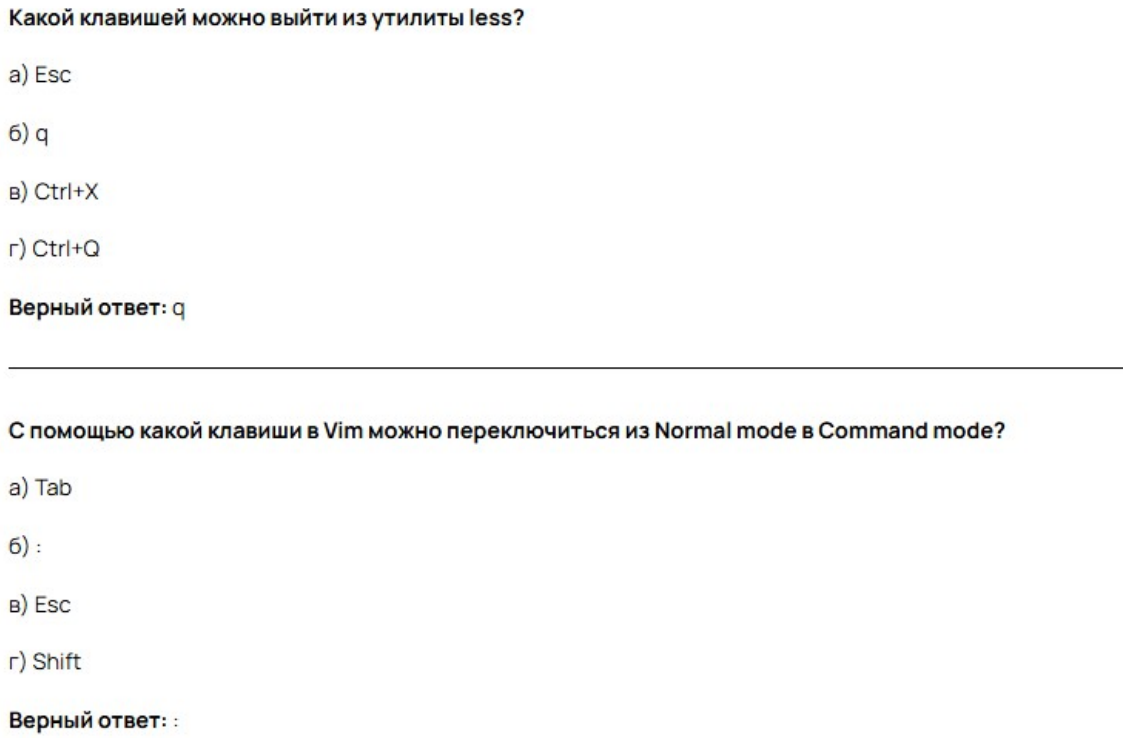

Рисунок 5.3: тест

Для выхода из просмотрщика less мы просто нажимаем клавишу q. В редакторе Vim, находясь в Normal mode, мы нажимаем клавишу :, чтобы перейти в Command mode.

Где хранятся основные лог-файлы в Linux?

а) /etc/logs/

б) /var/log/

в) /usr/logs/

г) /tmp/logs/

Верный ответ: /var/log/

Что делает команда journalctl -u sshd --since today?

а) Показывает ошибки входа за сегодня

б) Показывает все логи sshd до сегодняшнего дня

в) Показывает логи sshd за сегодня

г) Показывает логи sshd за вчера

Верный ответ: Показывает логи sshd за сегодня

Какой параметр journalctl показывает последние 20 записей?

а) -e

б) -u

в) --tail 20

г) -n 20

Верный ответ: -n 20

Рисунок 5.4: тест

Мы обращаемся к каталогу /var/log/, так как он является стандартным местом хранения логов системы и приложений. Мы используем эту команду, чтобы

отфильтровать и просмотреть все записи журнала systemd, относящиеся к службе sshd, начиная с начала текущих суток. Чтобы увидеть ограниченное количество последних записей, мы используем ключ -n. Команда journalctl -n 20 выведет последние 20 строк журнала.

Какая команда позволяет в реальном времени отслеживать новые строки в лог-файле?

а) cat /var/log/nginx.log

б) less /var/log/nginx.log

в) tail -f /var/log/nginx.log

г) watch -n 5 /var/log/nginx.log

Верный ответ: tail -f /var/log/nginx.log

Где по умолчанию хранятся пользовательские задания cron?

а) /var/spool/cron/

б) /home/user/.cronjobs

в) /etc/cron.d/

г) /opt/cron/tasks

Верный ответ: /var/spool/cron/

Рисунок 5.5: тест

Для наблюдения за растущим лог-файлом в реальном времени мы используем команду tail с ключом -f. Мы знаем, что индивидуальные задания планировщика cron для каждого пользователя хранятся в виде файлов в каталоге /var/spool/cron/

Какой символ в crontab означает «любое значение»?

а) *

б) -

в) /

г) %

Верный ответ: *

Как удалить все задания cron для текущего пользователя?

а) cron --clear

б) rm -rf /var/spool/cron

в) crond -reset

г) crontab -r

Верный ответ: crontab -r

Рисунок 5.6: тест

В записи cron мы используем символ звездочки * для обозначения «каждого» допустимого значения в поле (например, каждую минуту, каждый час). Чтобы полностью очистить нашу личную таблицу cron, мы выполняем команду crontab -r.

Какой флаг команды useradd используется для создания домашней директории пользователя?

а)-s

б)-G

в)-m

г) -d

Верный ответ: -m

Какая команда удалит пользователя вместе с его домашней директорией?

а)sudo userdel admin

б)sudo deluser admin

в)sudo userdel -r admin

г)sudo deluser --remove-all-files admin

Верный ответ: sudo userdel -r admin

Рисунок 5.7: тест

При создании пользователя мы используем ключ -m, чтобы система автоматически создала для него домашний каталог по умолчанию. Для полного удаления учетной записи пользователя и его домашнего каталога со всем содержимым мы применяем команду userdel с ключом -r

Что делает команда sudo usermod -aG sudo admin?

а) Заменяет основную группу пользователя admin на sudo

б) Добавляет пользователя admin в группу sudo, сохранив остальные группы

в) Удаляет пользователя admin из группы sudo

г) Создает новую группу sudo

Верный ответ: Добавляет пользователя admin в группу sudo, сохранив остальные группы

Какая команда используется для безопасного редактирования файла /etc/passwd?

а) nano /etc/passwd

б) vipw

в) vim /etc/passwd

г) usermod

Верный ответ: vipw

Рисунок 5.8: тест

Мы используем эту команду, чтобы предоставить пользователю admin права суперпользователя. Ключ -aG гарантирует, что пользователь будет добавлен в группу sudo, а не будет перемещен туда, потеряв все другие групповые членства. Для редактирования критичных системных файлов /etc/passwd и /etc/shadow мы используем специальную утилиту vipw. Она блокирует файл на время редактирования, предотвращая его одновременное изменение из другого места, что обеспечивает целостность данных.

Что означает первый символ d в строке прав доступа при выполнении команды ls -l?

а) Обычный файл

б) Символическая ссылка

в) Директория (каталог)

г) Специальный системный файл

Верный ответ: Директория (каталог)

Какая команда сменит владельца и группу файла /home/ivan/file.txt на ivan и friends соответственно?

а) chown ivan /home/ivan/file.txt

б) chgrp friends /home/ivan/file.txt

в) chown ivan:friends /home/ivan/file.txt

г) chmod ivan:friends /home/ivan/file.txt

Верный ответ: chown ivan:friends /home/ivan/file.txt

Какие права доступа соответствуют числовому значению 754?

а) rwxr--r–

б) rwxr-xr–

в) rwxr-xr-x

г) rwxrw-r--

Верный ответ: rwxr-xr–

Рисунок 5.9: тест

Первый символ в выводе ls -l указывает на тип файла. Буква d означает, что это каталог. Мы используем команду chown в формате «chown владелец:группа файл» для одновременного изменения владельца и группы файла. Мы расшифровываем числовой формат прав так: 7 (владелец: rwx), 5 (группа: r-x), 4 (остальные: r–). Это соответствует строке rwxr-xr–.

Что произойдет, если вы используете команду sudo для выполнения действий?

а) Команда выполнится с правами суперпользователя, и действия не будут записаны в журнал

б) Будет зафиксировано, кто и когда выполнил команду с правами суперпользователя

в) Все действия, выполненные с помощью sudo, не записываются

г) Все команды под sudo выполняются с ограниченными правами, независимо от пользователя

Верный ответ: Будет зафиксировано, кто и когда выполнил команду с правами суперпользователя

Что произойдет, если вы измените порт SSH с 22 на 47022, но не обновите файрвол?

а) SSH будет работать на новом порту, но файрвол не разрешит подключения

б) Подключение будет возможно только по стандартному порту 22

в) Все подключения будут заблокированы

Верный ответ: SSH будет работать на новом порту, но файрвол не разрешит подключения

Рисунок 5.10: тест

Все команды, выполненные через sudo, записываются в системный журнал, что позволяет нам отслеживать, кто, когда и что делал с привилегиями root. После изменения порта SSH демон будет работать на новом порту, но межсетевой экран по умолчанию блокирует все входящие соединения, кроме разрешенных правилами.

Какую команду следует выполнить для того, чтобы добавить в файрвол ufw разрешение на подключение к нестандартному порту 47022 по протоколу TCP?

а) sudo ufw allow 22/tcp

б) sudo ufw allow 47022

в) sudo ufw allow 47022/tcp

Верный ответ: sudo ufw allow 47022/tcp

Почему рекомендуется использовать sudo, а не su?

а) sudo позволяет работать с командой от имени root без предоставления пароля

б) sudo не требует ввода пароля root, а выполняемые команды журналируются, обеспечивая прозрачность действий

в) su позволяет больше команд, чем sudo

г) su более безопасен, чем sudo

Верный ответ: sudo не требует ввода пароля root, а выполняемые команды журналируются, обеспечивая прозрачность действий

Рисунок 5.11: тест

Чтобы разрешить входящие подключения к нестандартному порту SSH, мы добавляем в UFW правило, явно указывающее номер порта и протокол: sudo ufw allow 47022/tcp. Мы предпочитаем sudo, потому что он позволяет предоставлять привилегированный доступ без разглашения пароля root, обеспечивает детальное логирование действий и дает более гибкое управление правами через файл /etc/sudoers.

Какой из следующих вариантов наиболее точно описывает хеширование пароля?

а) Пароль шифруется с возможностью обратного расшифрования

б) Пароль сохраняется в виде обычного текста

в) Пароль преобразуется в уникальный отпечаток, который нельзя восстановить обратно

г) Пароль кодируется с помощью Base64

Верный ответ: Пароль преобразуется в уникальный отпечаток, который нельзя восстановить обратно

Что обозначает первая часть строки пароля в /etc/shadow, которая начинается с символа $ (например, $y)?

а) Уровень безопасности пользователя

б) Алгоритм шифрования

в) Используемый алгоритм хеширования

г) Имя пользователя

Верный ответ: Используемый алгоритм хеширования

Какова функция «соли» (salt) при хешировании пароля?

а) Обеспечить возможность расшифровать хеш

б) Сделать пароль легче для пользователя

в) Сделать каждый хеш уникальным и предотвратить атаки с использованием rainbow-таблиц

г) Изменить пароль на другой

Верный ответ: Сделать каждый хеш уникальным и предотвратить атаки с использованием rainbow-таблиц

Рисунок 5.12: тест

Мы понимаем, что хеширование — это односторонняя криптографическая функция. Идентификатор алгоритма хеширования, такой как 6 (SHA-512) или *y* (yescrypt), указывает системе, какой метод использовался для создания этого хеша. Мы добавляем «соль» — случайную строку — к паролю перед хешированием. Это гарантирует, что даже одинаковые пароли будут иметь разные хеши.

Какая команда позволяет просмотреть текущие параметры политики пароля для пользователя ivan?

а) passwd -s ivan

б) cat /etc/shadow | grep ivan

в) chage -l ivan

г) usermod -p ivan

Верный ответ: chage -l ivan

Что произойдет после выполнения команды sudo usermod -L username?

а) Удалится пользователь из системы

б) У пользователя сменится оболочка

в) Учетная запись будет заблокирована, и вход станет невозможным

г) Пользователю будет назначен временный пароль

Верный ответ: Учетная запись будет заблокирована, и вход станет невозможным

Рисунок 5.13: тест

Чтобы проверить настройки политики паролей, мы используем команду chage с ключом -l. Команда usermod -L блокирует учетную запись пользователя, добавляя знак ! в начало его хешированного пароля в /etc/shadow, что делает невозможным вход в систему с этим паролем.

Какую команду надо ввести, чтобы посмотреть, какие права выданы файлам?

а) ls -a

б) lshb

в) rwx -l

г) ls -l

Верный ответ: ls -l

Какую команду надо ввести, чтобы посмотреть, какие права выданы файлам, в том числе — скрытым?

а) ls -a

б) grep etc/files

в) ls -la

г) umask 006

Верный ответ: ls -la

Как будут записаны права r w - в восьмеричном формате?

а) 3

б) 110

в) 6

г) - - -

Верный ответ: 6

Рисунок 5.14: тест

Для просмотра подробного списка файлов, включая их права доступа, владельца и группу, мы используем команду ls -l. Комбинация флагов -l и -a в команде ls позволяет нам увидеть права доступа у всех файлов, включая скрытые. Мы переводим символьные права в числовые: r (read) = 4, w (write) = 2, - (no execute) = 0. Суммируем: 4 + 2 + 0 = 6.

Какую команду нужно использовать для изменения прав файлов и каталогов?

а) chown

б) chmod

в) chgrp

г) nano file.txt

Верный ответ: chmod

С помощью каких операторов можно указать тип изменения прав?

а) + - =

б) r w x

в) u g o a

Верный ответ: + - =

Какую опцию нужно применить, чтобы изменить прав ко всем каталогам, в которые вложен целевой файл?

а) -R

б) -man

в) -la

г) o-rx

Верный ответ: -R

Рисунок 5.15: тест

Для изменения прав доступа мы используем команду chmod. В символьном режиме команды chmod мы используем операторы: + для добавления прав, - для отзыва прав и = для установки прав в точное значение. Чтобы рекурсивно применить изменения прав ко всем файлам и подкаталогам внутри директории, мы используем ключ -R.

Как называется специальное разрешение, благодаря которому файлы в каталоге, которому выставлен этот бит разрешения, могут быть удалены только их владельцами или владельцами каталога, где лежит этот файл?

а) Sticky Bit

б) SGID

в) SUID

г) total

Верный ответ: Sticky Bit

Как называется параметр безопасности, благодаря которому можно разрешить пользователям запускать программу от имени владельца? При условии, что права на выполнение выданы изначально.

а) SGID

б) SUID

в) Sticky Bit

г) нет правильного ответа

Верный ответ: SUID

Как называется параметр безопасности, благодаря которому можно разрешить пользователям запускать файл от имени владельца группы файла? При условии, что права на выполнение не выданы изначально.

а) SGID

б) Sticky Bit

в) SUID

г) нет правильного ответа

Верный ответ: нет правильного ответа

Рисунок 5.16: тест

Мы устанавливаем Sticky Bit на общедоступные каталоги вроде /tmp, чтобы пользователи могли удалять только свои собственные файлы, даже если у них есть право на запись в каталог. SUID — это специальное право, которое заставляет исполняемый файл запускаться с правами его владельца, а не пользователя, который его запустил. Такого параметра не существует.

Какой командой можно вывести список всех процессов с детальной информацией об использовании CPU и памяти?

а) ps -ef

б) ps aux

в) top -n 1

Верный ответ: ps aux

Какой сигнал отправляется процессу командой kill -9?

а) SIGTERM (15)

б) SIGKILL (9)

в) SIGSTOP (19)

Верный ответ: SIGKILL (9)

Какой командой можно приостановить выполнение процесса и перевести его в фон?

а) Ctrl+Z, затем bg %1

б) kill -STOP

в) fg %1

Верный ответ: Ctrl+Z, затем bg %1

Рисунок 5.17: тест

Для получения подробного списка всех процессов в системе с информацией о пользователе, PID, использовании CPU и памяти мы используем команду ps aux. Когда процесс не реагирует на другие сигналы, мы используем команду kill -9, которая отправляет сигнал SIGKILL. Мы приостанавливаем процесс, работающий в терминале, нажатием Ctrl+Z. Затем, чтобы возобновить его выполнение в фоновом режиме, мы вводим команду bg %1.

Какое значение nice имеет наивысший приоритет?

а) -20

б) 0

в) 19

г) ps

Верный ответ: -20

Какой командой изменить приоритет уже запущенного процесса с PID 1234 на nice=10?

а) nice -n 10 -p 1234

б) renice -n 10 -p 1234

в) priority -n 10 1234

г) nice = = 10 -f 1234

Верный ответ: renice -n 10 -p 1234

Какой параметр в unit-файле systemd устанавливает приоритет CPU для сервиса?

а) CPUPriority=10

б) Nice=10

в) Priority=10

г) Renice=10

Верный ответ: Nice=10

Рисунок 5.18: тест

Значение nice определяет приоритет планировщика. Чем оно ниже, тем выше приоритет процесса. Таким образом, значение -20 является наивысшим приоритетом, а 19 — самым низким. Для изменения приоритета уже работающего процесса мы используем команду renice. В unit-файле systemd мы используем директиву Nice=, чтобы установить значение nice для всех процессов этого сервиса при их запуске.

Какой командой проверить статус сервиса nginx?

а) systemctl status nginx

б) service nginx check

в) ps aux | grep nginx

г) нет правильного ответа

Верный ответ: systemctl status nginx

Какой параметр в таймере systemd указывает ежедневный запуск в полночь?

а) OnTime=daily

б) OnCalendar=daily

в) Schedule=24h

г) нет правильного ответа

Верный ответ: OnCalendar=daily

Какой командой включить автозапуск сервиса при загрузке системы?

а) systemctl start servicename

б) systemctl enable servicename

в) systemctl reload servicename

г) нет правильного ответа

Верный ответ: systemctl enable servicename

Рисунок 5.19: тест

Для проверки состояния, активности и последних логов системного сервиса мы используем команду systemctl status имя_сервиса. В таймере systemd мы используем директиву OnCalendar= с параметром daily, чтобы настроить ежедневный запуск в полночь. Чтобы сервис автоматически запускался при загрузке системы, мы «включаем» его с помощью команды systemctl enable.

Это создает необходимые символические ссылки.

Какой параметр в unit-файле обеспечивает перезапуск сервиса при любом завершении?

а) Restart=on-failure

б) Restart=always

в) AutoRestart=true

г) Type=simple

Верный ответ: Restart=always

Какой командой просмотреть логи сервиса в реальном времени?

а) tail -f /var/log/syslog

б) journalctl -u servicename -f

в) systemctl log servicename

г) watch -n 1 "ps aux | grep 'python3 /home/user/myscript.py'"

Верный ответ: journalctl -u servicename -f

Какой командой проверить синтаксис unit-файла перед запуском?

а) systemctl check mydaemon.service

б) systemd-analyze verify mydaemon.service

в) validate-unit mydaemon.service

г) sudo systemctl daemon-reload

Верный ответ: systemd-analyze verify mydaemon.service

Рисунок 5.20: тест

Мы используем директиву Restart=always в unit-файле systemd, когда нам нужно, чтобы сервис автоматически перезапускался независимо от того, как он завершился — штатно, с ошибкой или был принудительно остановлен. Это обеспечивает максимальную отказоустойчивость службы. Для мониторинга журнала конкретного сервиса в реальном времени мы используем команду

journalctl -u servicename -f. Перед тем как запускать новый сервис, мы всегда проверяем его unit-файл на наличие синтаксических ошибок с помощью команды systemd-analyze verify.

6 Оценки тестов

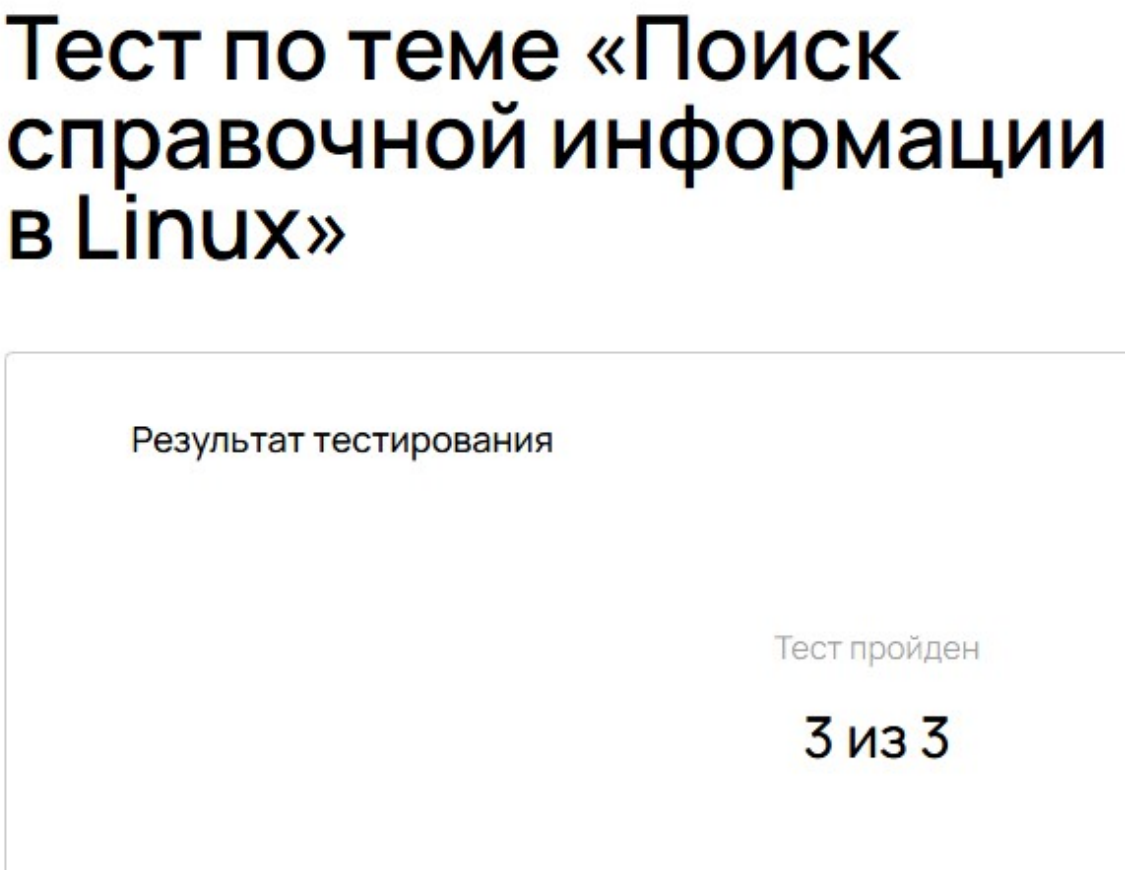

Рисунок 6.1: тест 1

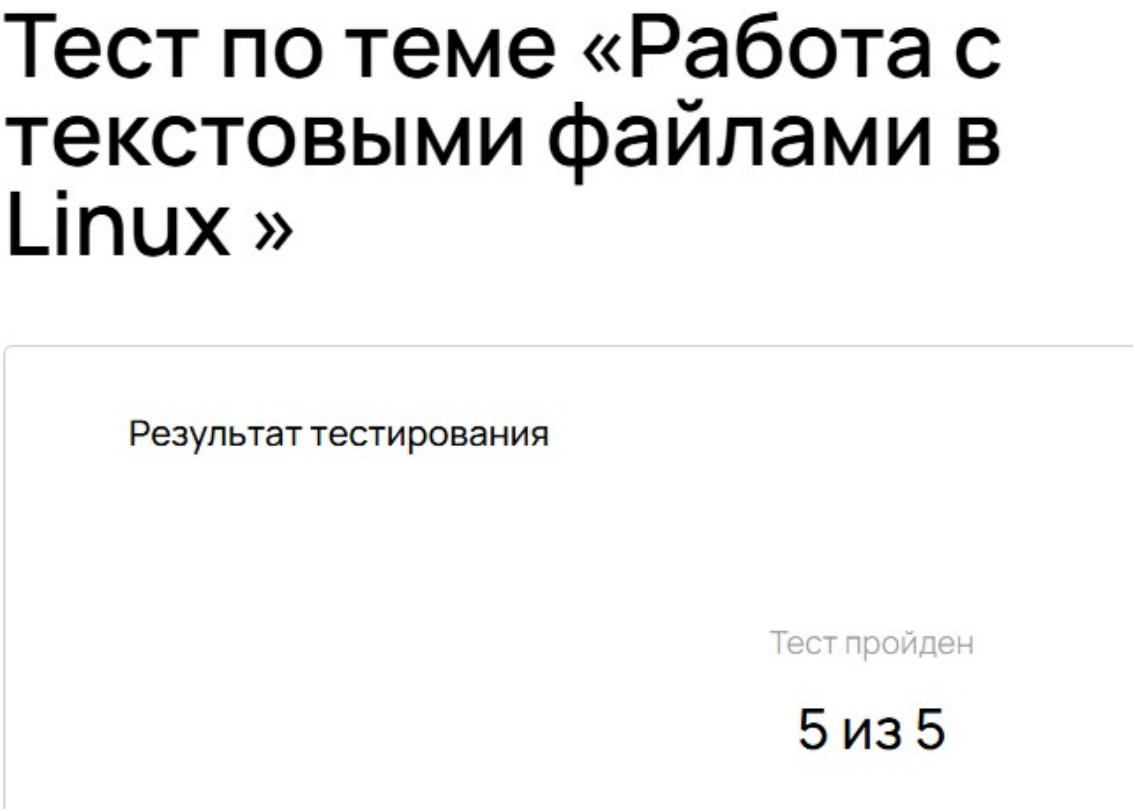

Рисунок 6.2: тест 2

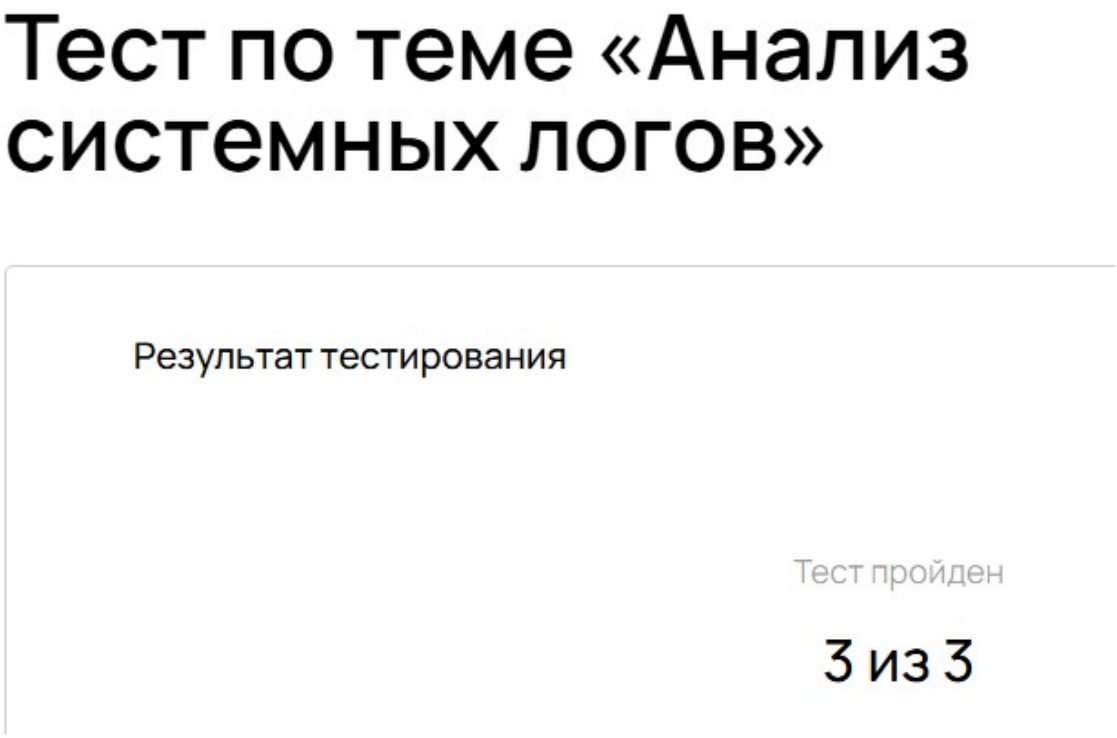

Рисунок 6.3: тест 3

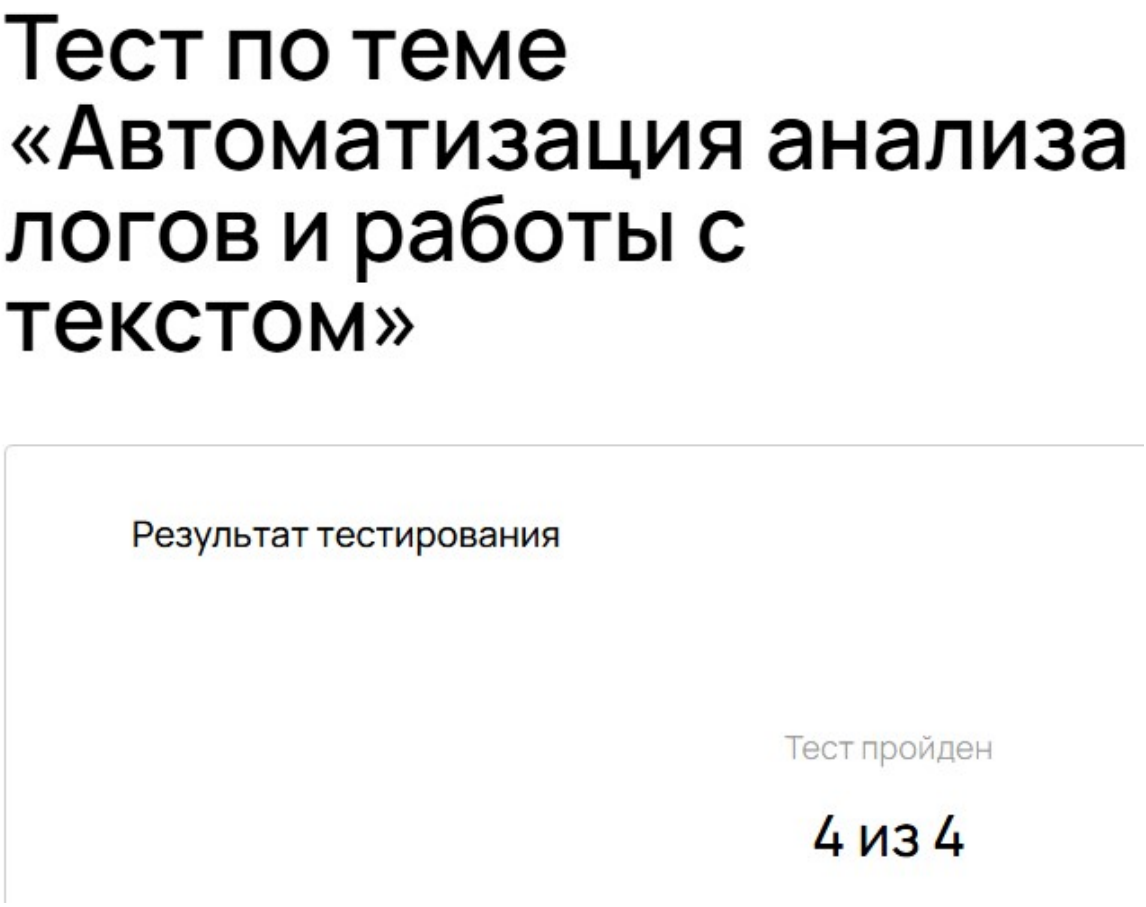

Рисунок 6.4: тест 4

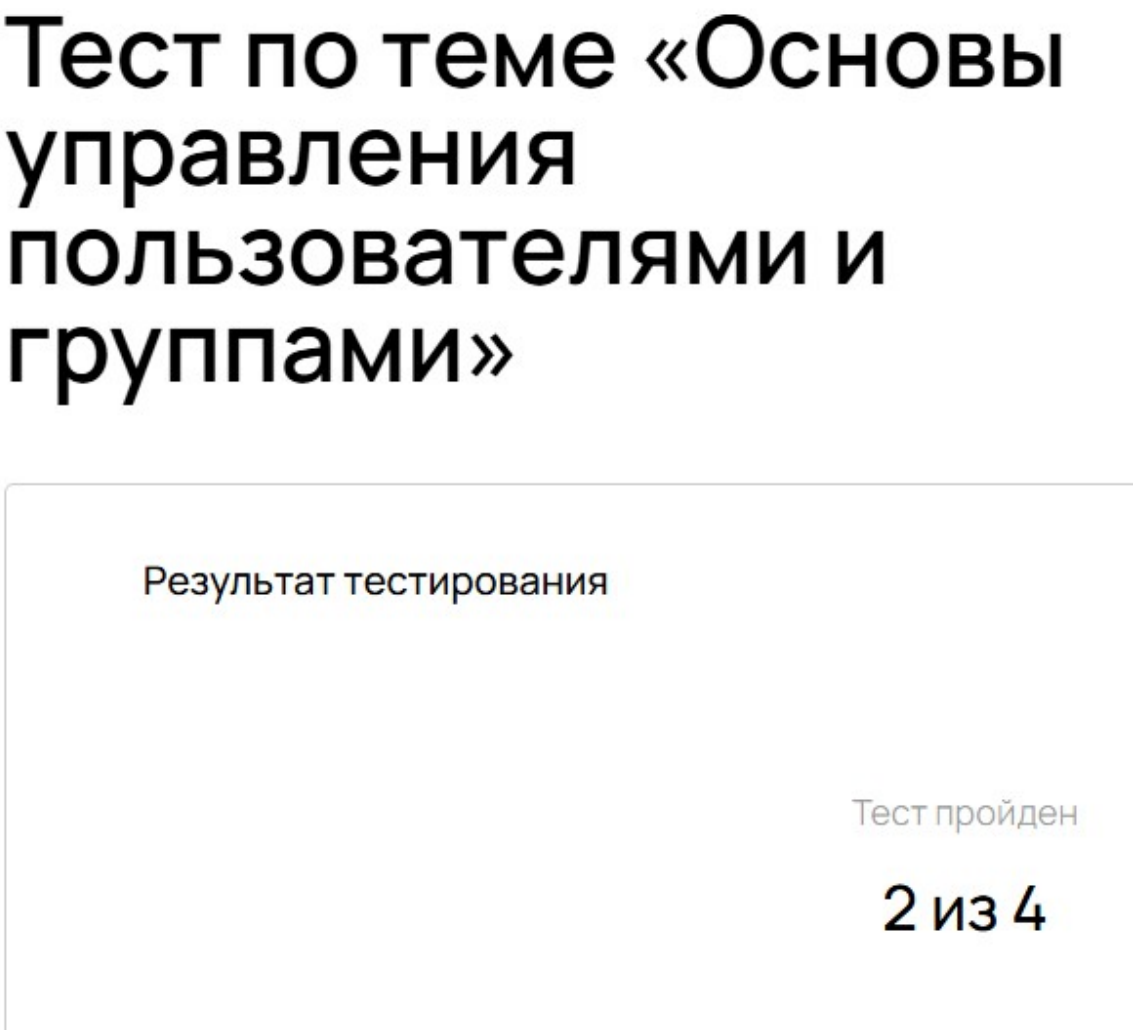

Рисунок 6.5: тест 5

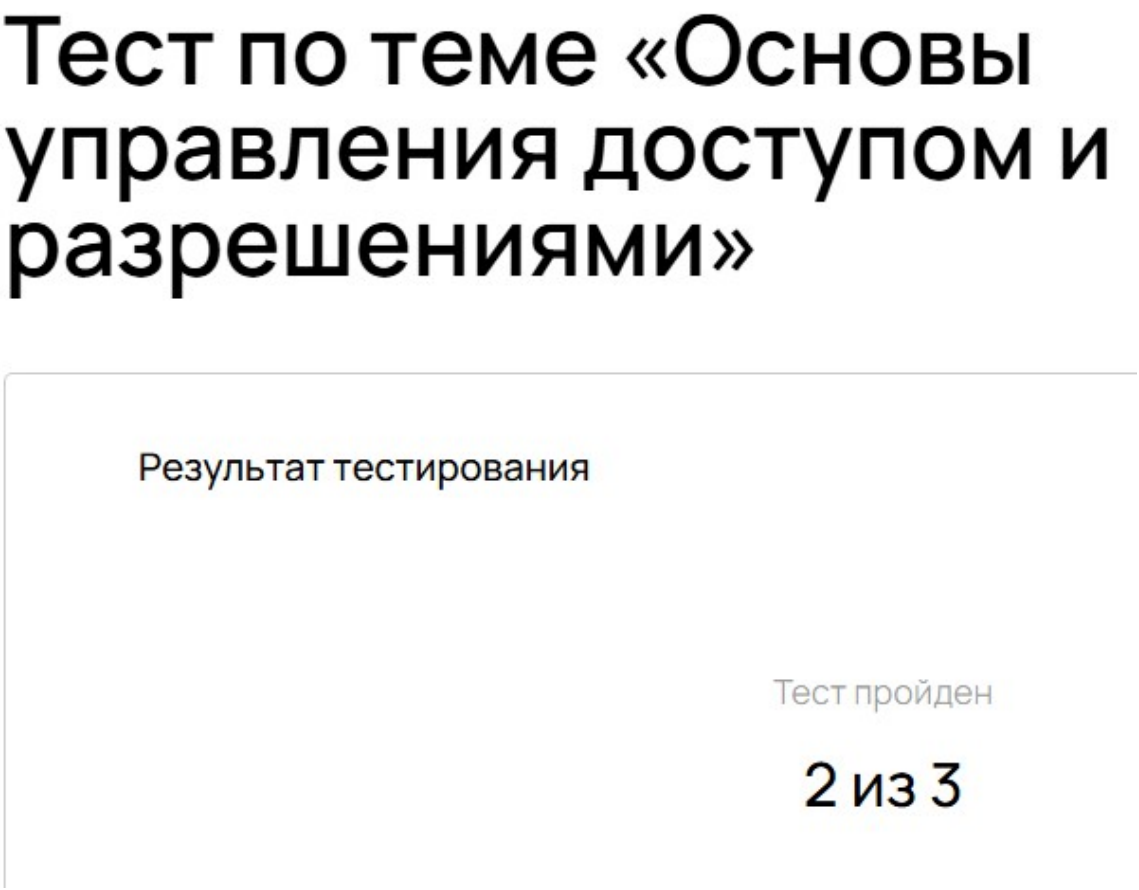

Рисунок 6.6: тест 6

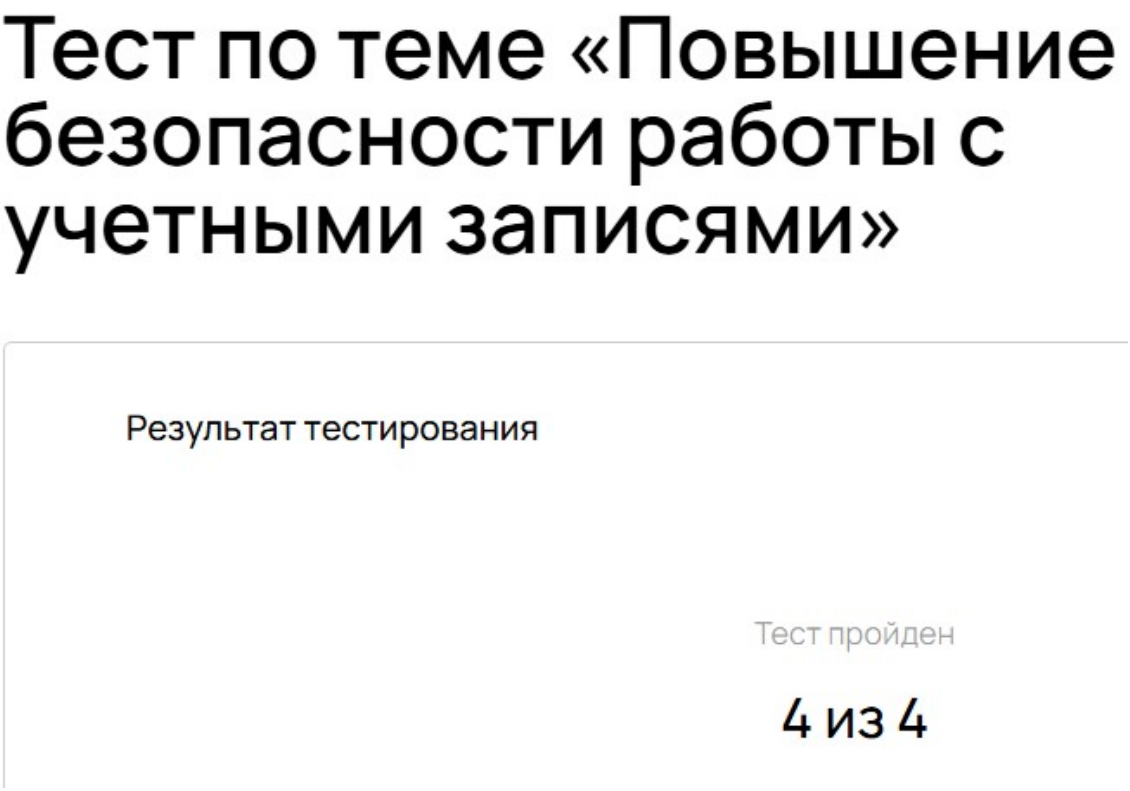

Рисунок 6.7: тест 7

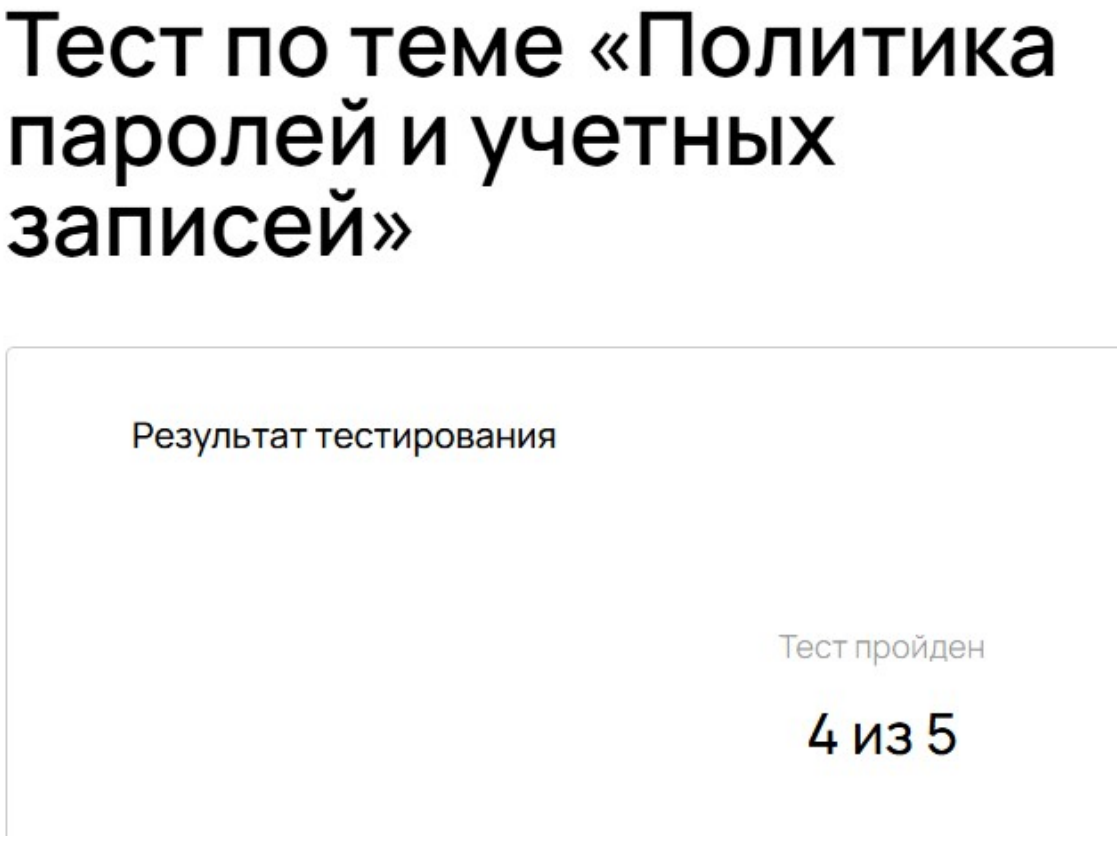

Рисунок 6.8: тест 8

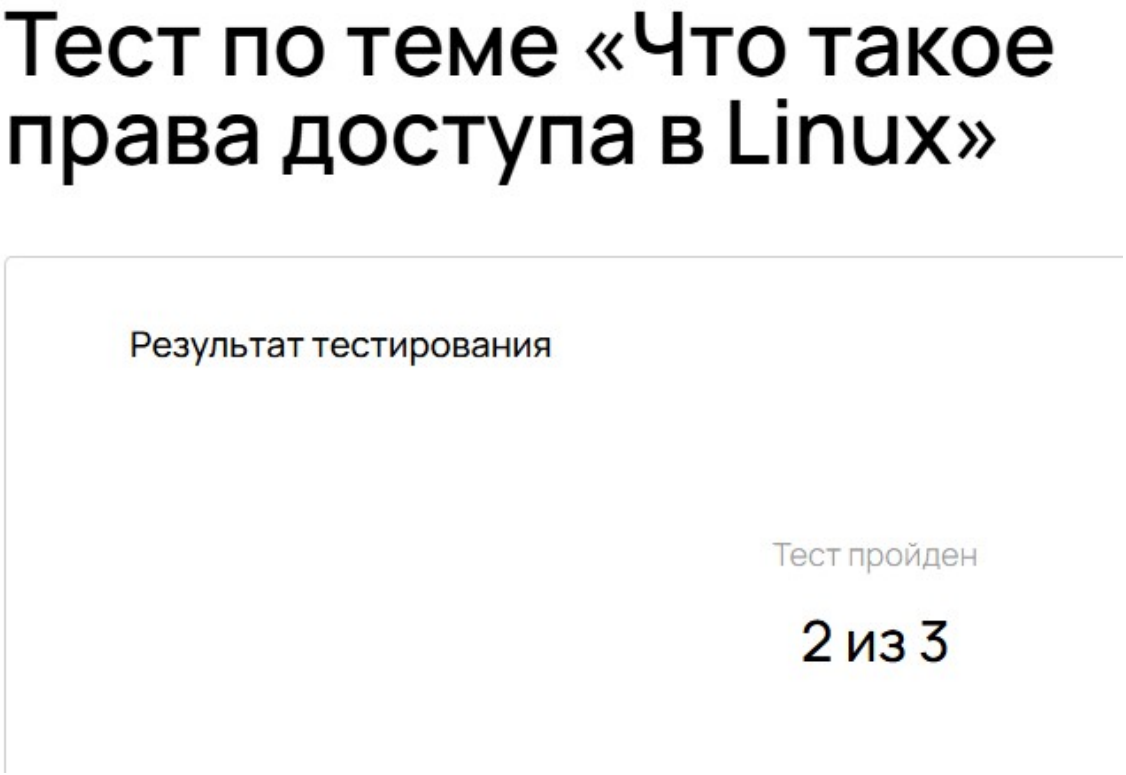

Рисунок 6.9: тест 9

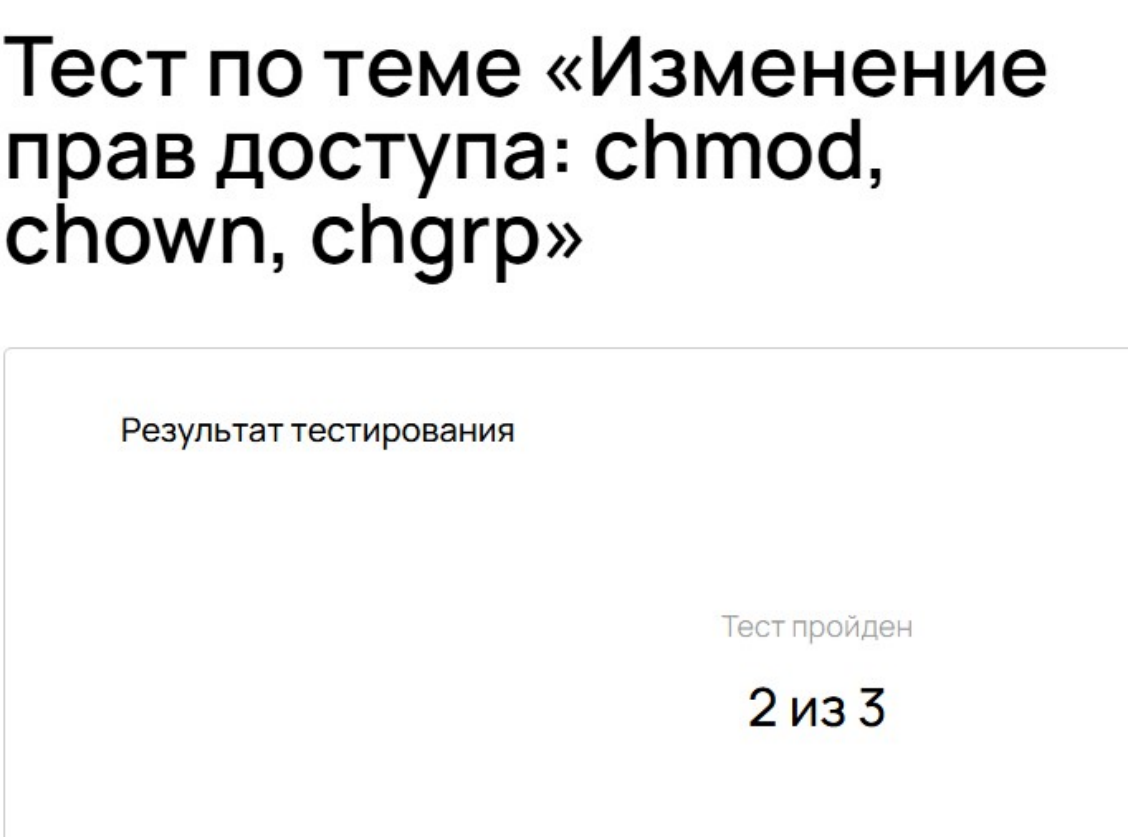

Рисунок 6.10: тест 10

Рисунок 6.11: тест 11

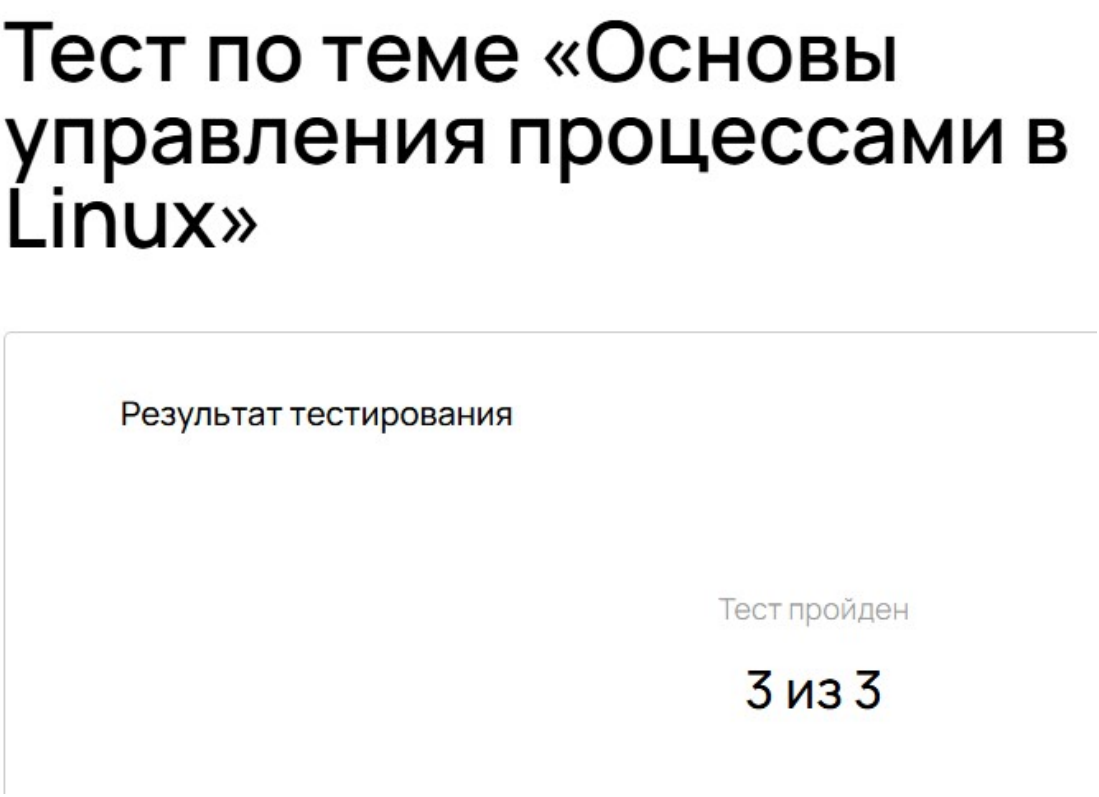

Рисунок 6.12: тест 12

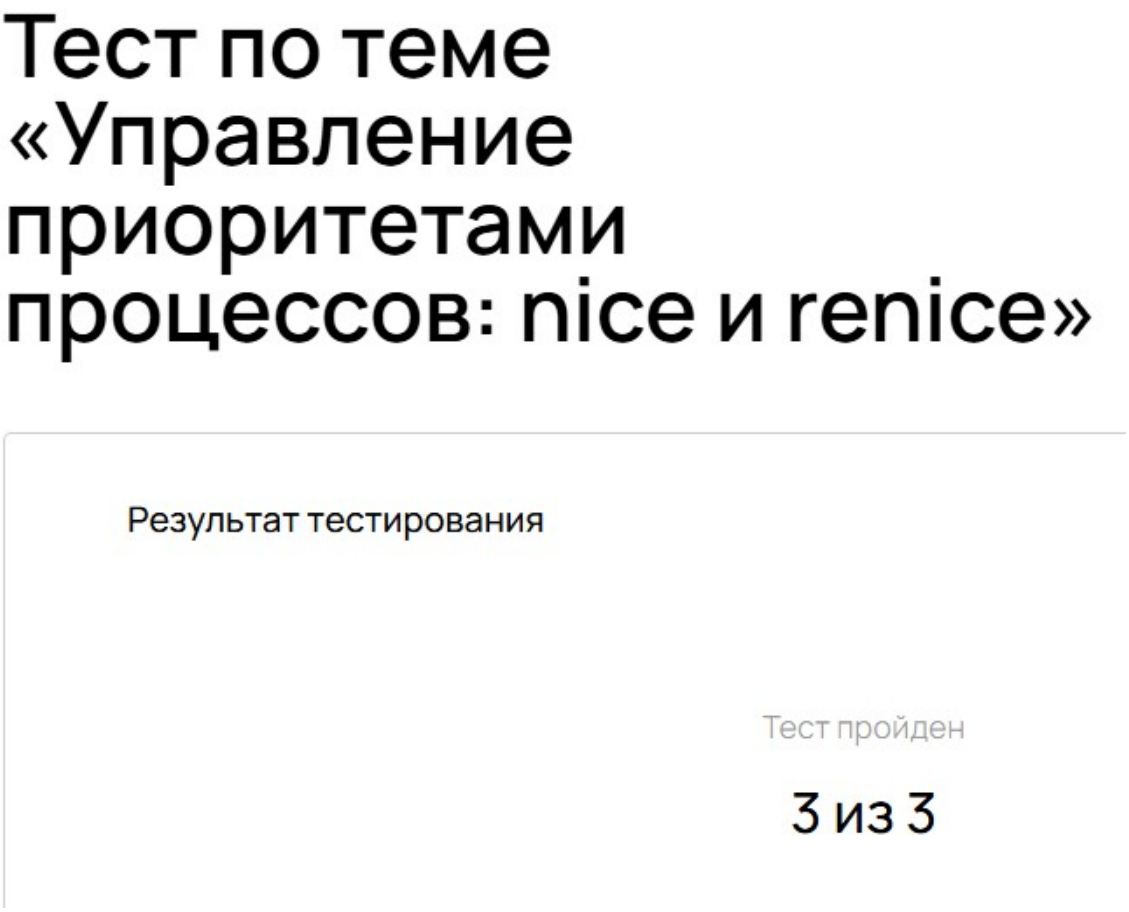

Рисунок 6.13: тест 13

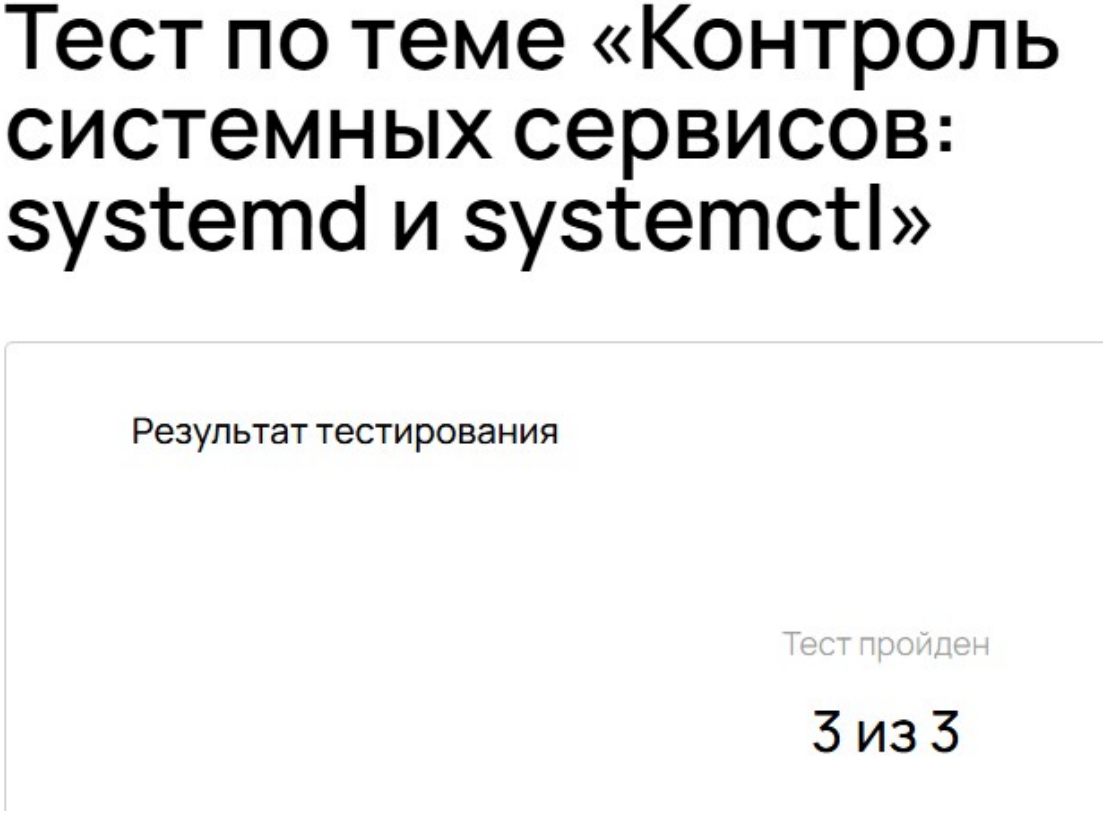

Рисунок 6.14: тест 14

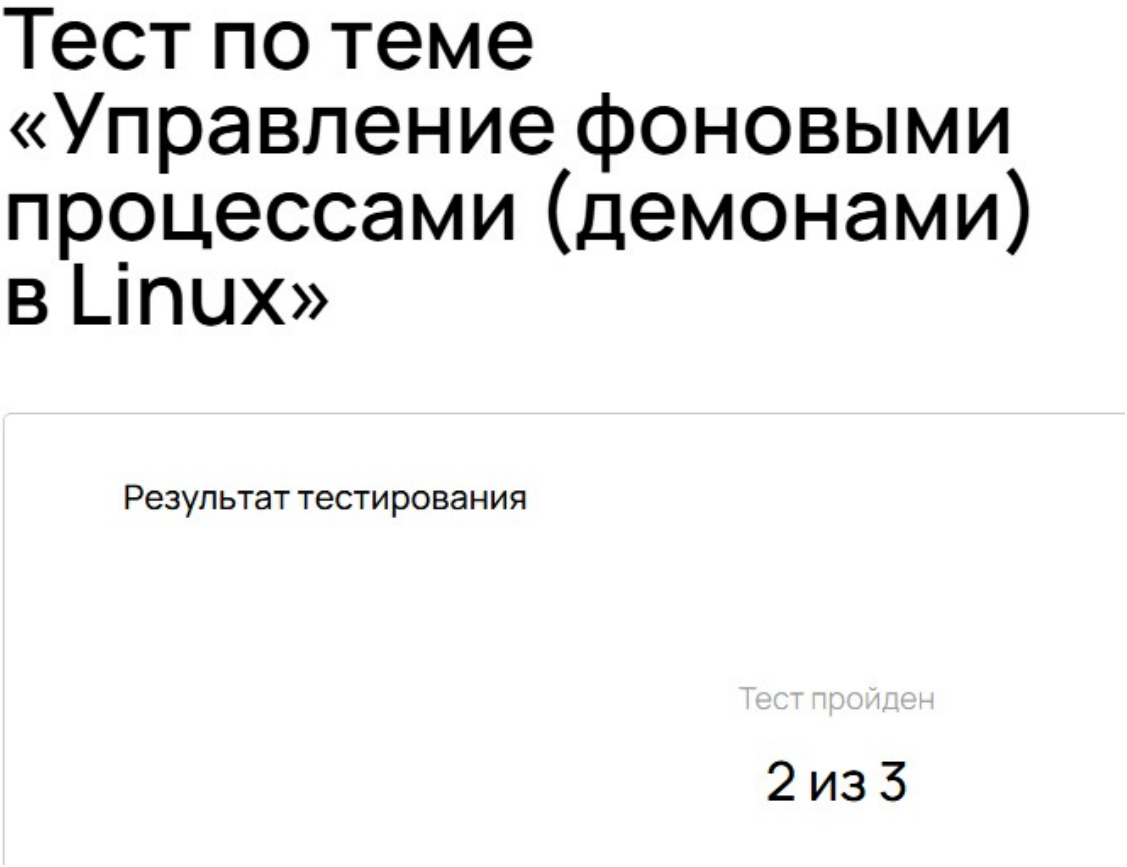

Рисунок 6.15: тест 15

7 Выводы

Мы выполнили второй раздел внешнего курса «Системный администратор Linux с нуля».

Список литературы

1. https://study.selectel.ru/members/courses/course756726784647