Прохождение внешнего курса. Часть 1. Основы системного администрирования и Linux

Отчёт

Сергеев Даниил Олегович

Содержание

# 1 Цель работы

Разобрать, чем занимается системный администратор, какие задачи выполняет и почему его роль критически важна. Узнать, что такое «система», из чего состоит инфраструктура и как управлять ее элементами.

# 2 Задание

* Модуль 1. Введение в системное администрирование
* Модуль 2. Базовые команды Linux
* Модуль 3. Файлы и каталоги

# 3 Ход выполнения лабораторной работы

## 3.1 Модуль 1. Введение в системное администрирование

### 3.1.1 Тест по теме «Роль системного администратора Linux»



Рис. 1: Подтверждение прохождения теста «Роль системного администратора Linux»

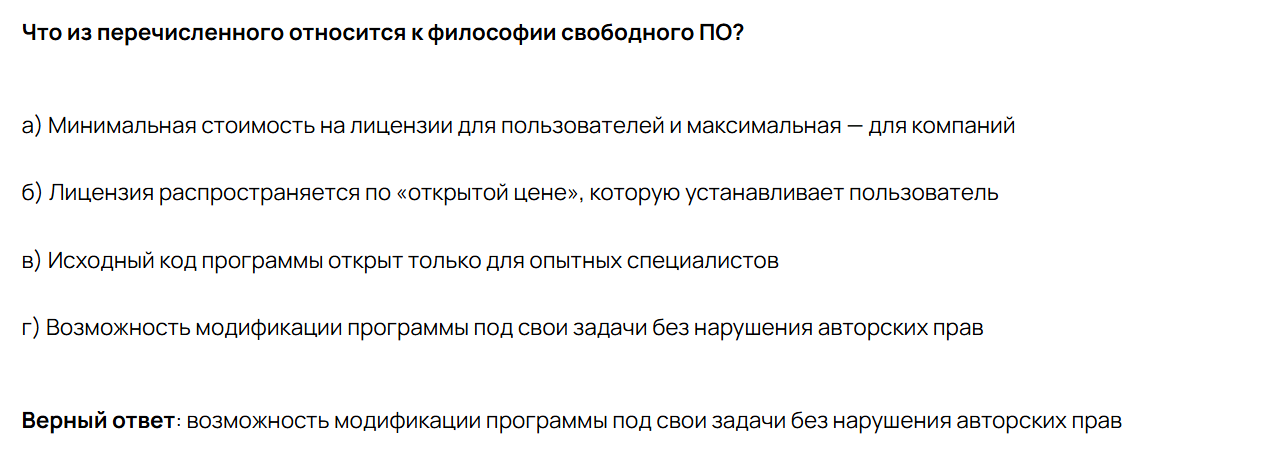


Рис. 2: «Роль системного администратора Linux». Вопрос №1

Выбранный ответ: **возможность модификации программы под свои задачи без нарушения авторских прав**.

Философия свободного ПО - свобода для изучения, применения, изменения и распространения программного обеспечения, собственно открытый исходный код программ.

Все остальные варианты возможны, но они не отражают философию свободного ПО.



Рис. 3: «Роль системного администратора Linux». Вопрос №2

Выбранный ответ: **Zabbix**.

* Zabbix - система мониторинга серверов. Он помогает следить за состоянием серверов, а также сигнализирует об ошибках до того, как они станут критическими.
* Exim - почтовый сервис.
* Puppet - помогает централизованно управлять конфигурацией множества серверов. Работает как инструмент для сокращения рутинных действий, а не как система мониторинга серверов.

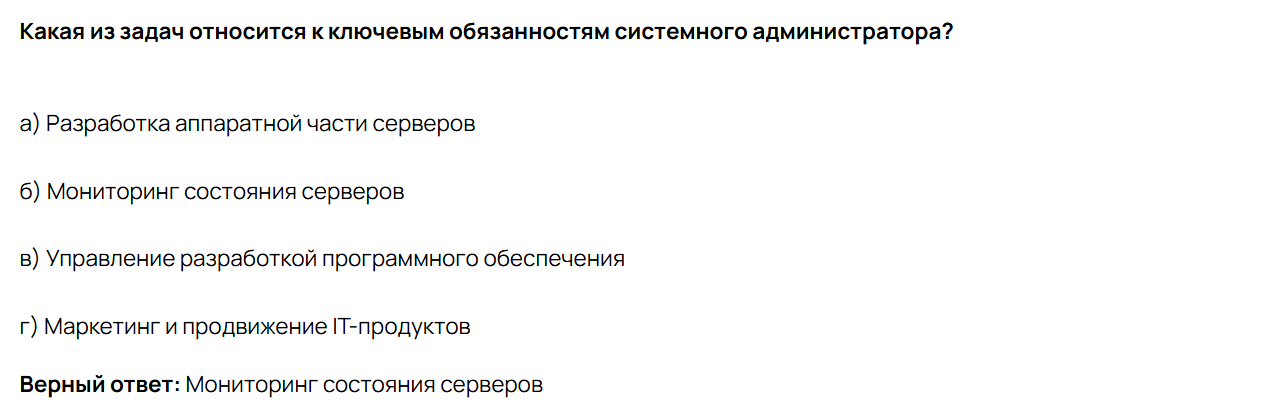


Рис. 4: «Роль системного администратора Linux». Вопрос №3

Выбранный ответ: **Мониторинг состояния серверов**.

Очевидно, что аппаратной частью серверов и управлением разработкой ПО занимаются инженеры и руководители отделов разработки соответственно. За продвижение IT-продуктов отвечает маркетолог. Все перечисленное не входит в обязанности системного администратора.

## 3.2 Модуль 2. Базовые команды Linux

### 3.2.1 Задания по теме «Вводный урок»

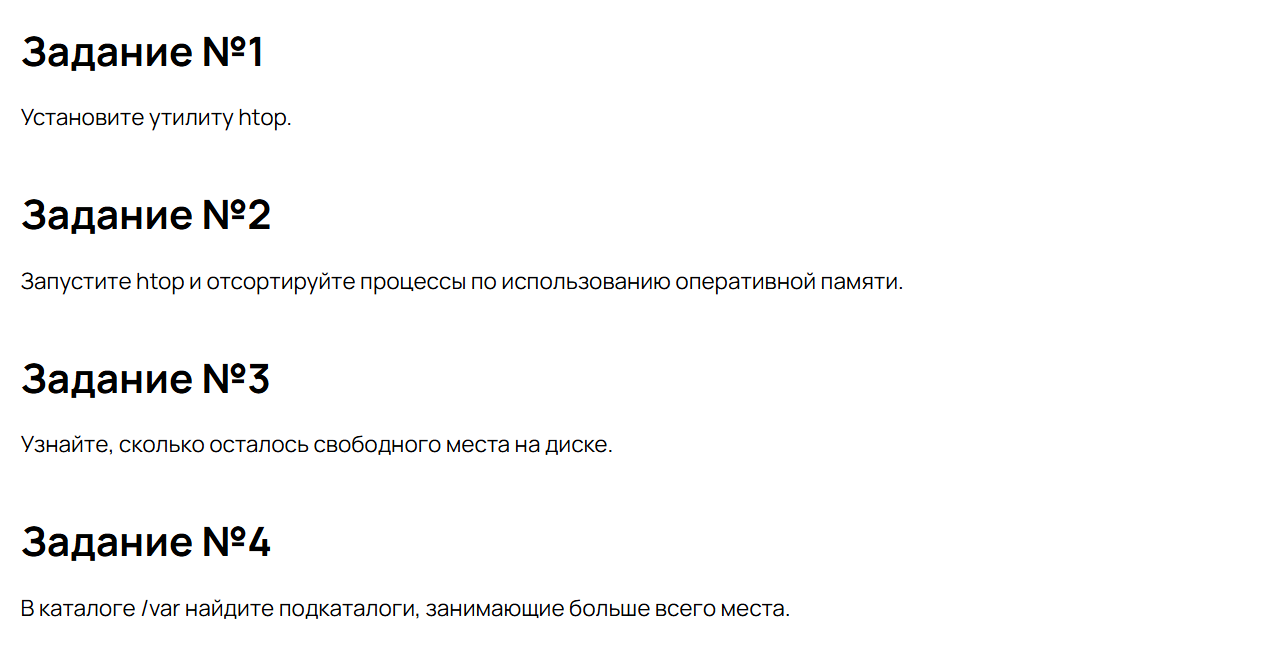


Рис. 5: «Вводный урок». Условия заданий

#### 3.2.1.1 Задание №1

Перейдем в учётную запись root пользователя и установим htop

su -  
dnf install -y htop

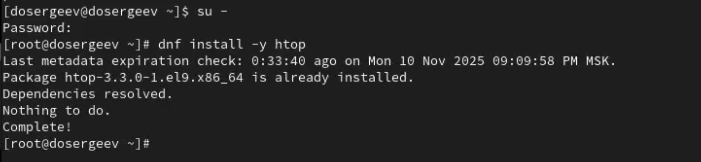


Рис. 6: Установка htop

#### 3.2.1.2 Задание №2

Запустим htop

htop

Чтобы отсортировать процессы по памяти, нажмем на клавишу F6 либо выберем её в нижней панели и в открывшемся окне укажем фильтр по использованию памяти PERCENT\_MEM.

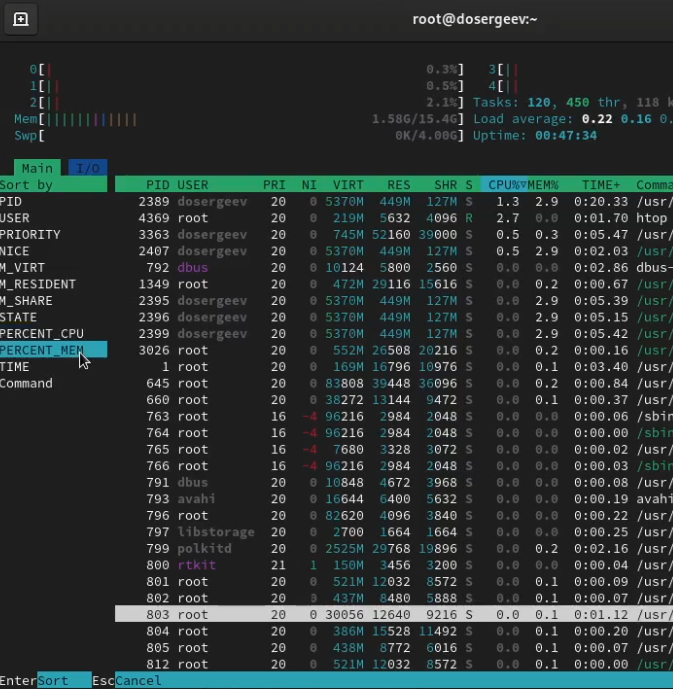


Рис. 7: Фильтрация процессов htop

#### 3.2.1.3 Задание №3

Чтобы узнать, сколько памяти занято в файловых системах, введем команду

df -h

Опция -h выведет размер в удобном для человека виде.

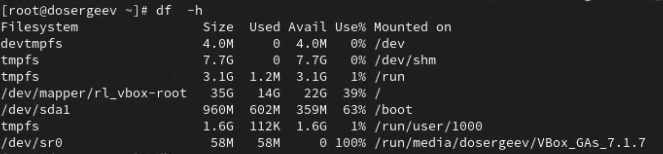


Рис. 8: Размер файловых систем на диске

Отсюда видим, что свободно 22 Гб памяти, или же 39% от всей вместимости файловой системы /dev/mapper/rl\_vbox-root.

#### 3.2.1.4 Задание №4

Выведем суммарный размер каталогов и их подкаталогов по пути /var/\*

# -h удобный для чтения формат размера  
# -s вывод суммы размеров подкаталогов  
du -hs /var/\*

Каталоги, занимающие больше всего места: /var/lib (280 мб), /var/cache (269 мб), /var/log (78 мб).

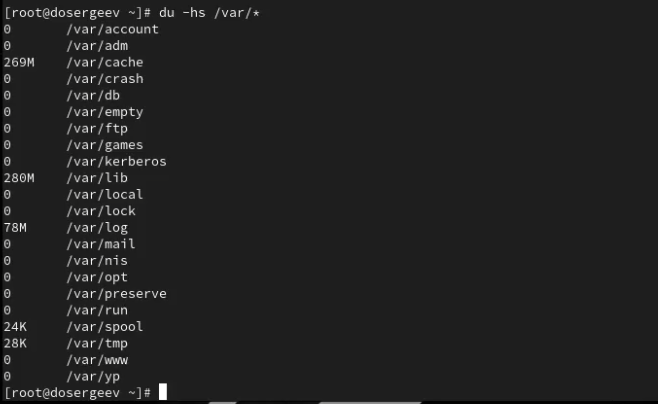


Рис. 9: Размер подкаталогов /var

### 3.2.2 Тест по теме «Вводный урок»

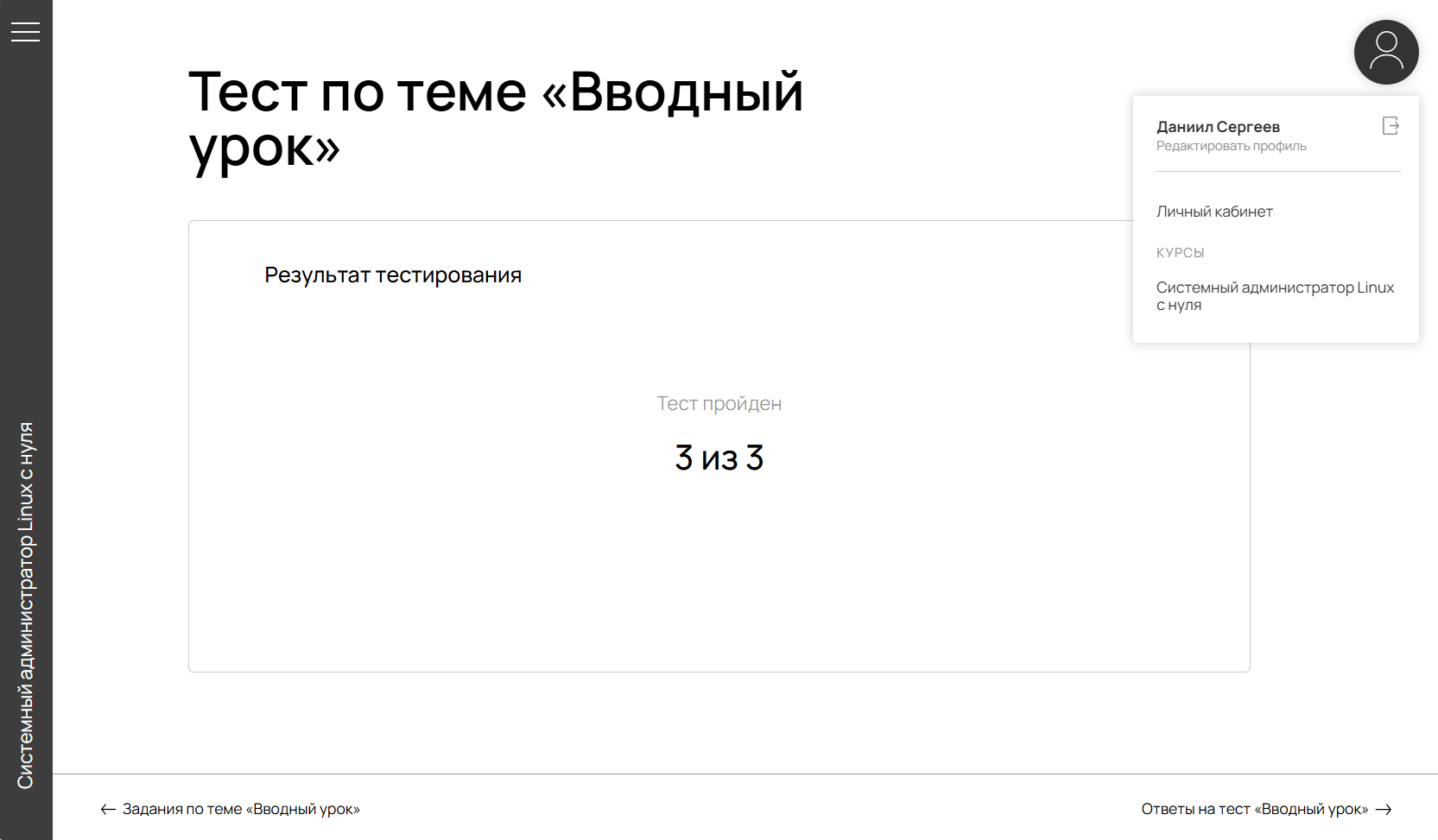


Рис. 10: Подтверждение прохождения теста «Вводный урок»

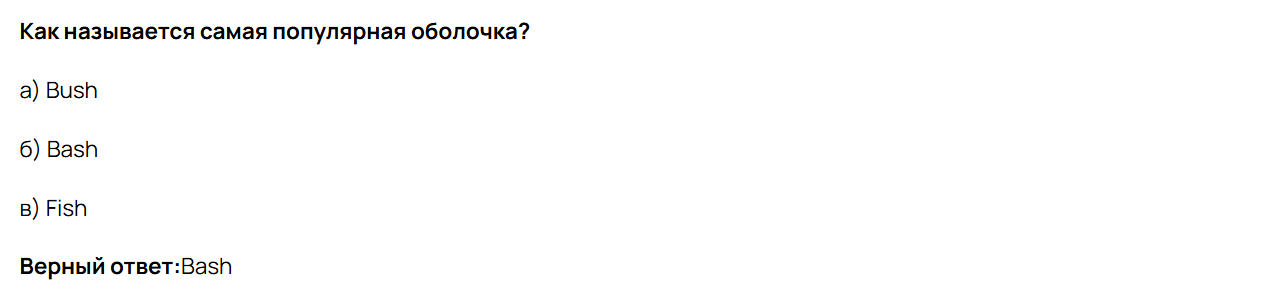


Рис. 11: «Вводный урок». Вопрос №1

Выбранный ответ: **Bash**.

Bash является наиболее известной оболочкой наряду с Zsh, Tcsh и Ksh. Fish менее известен, так как не поддерживает стандарт POSIX.

Bush является опечаткой в слове Bash.

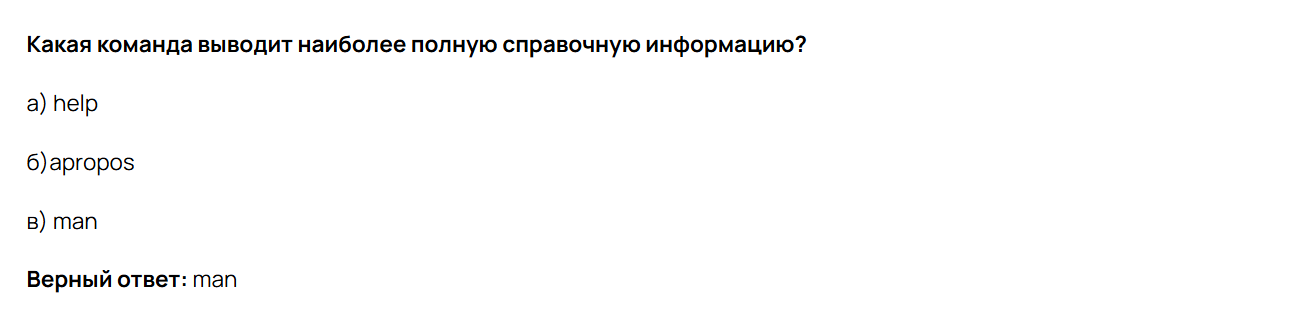


Рис. 12: «Вводный урок». Вопрос №2

Выбранный ответ: **man**.

Самая мощная команда для получения справки. Она отображает подробное руководство: описание, синтаксис, опции, примеры использования и ссылки. help предоставляет в основном описание встроенных команд текущей оболочки и системные переменные.

apropos позволяет искать в файлах с руководствами (man-страницах) указанные строки.

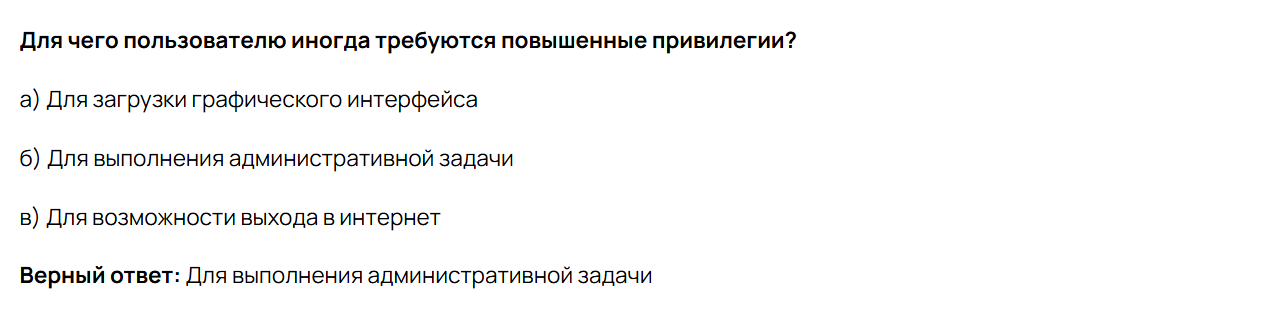


Рис. 13: «Вводный урок». Вопрос №3

Выбранный ответ: **Для выполнения административной задачи**.

Если система специально настроена на доступ к интернету только для супер пользователя, то повышенные привилегии не потребуются для обычного пользователя. Загрузка графического интерфеса также не требует повышенных привелегий.

### 3.2.3 Задания по теме «Роль командной строки в серверной среде»

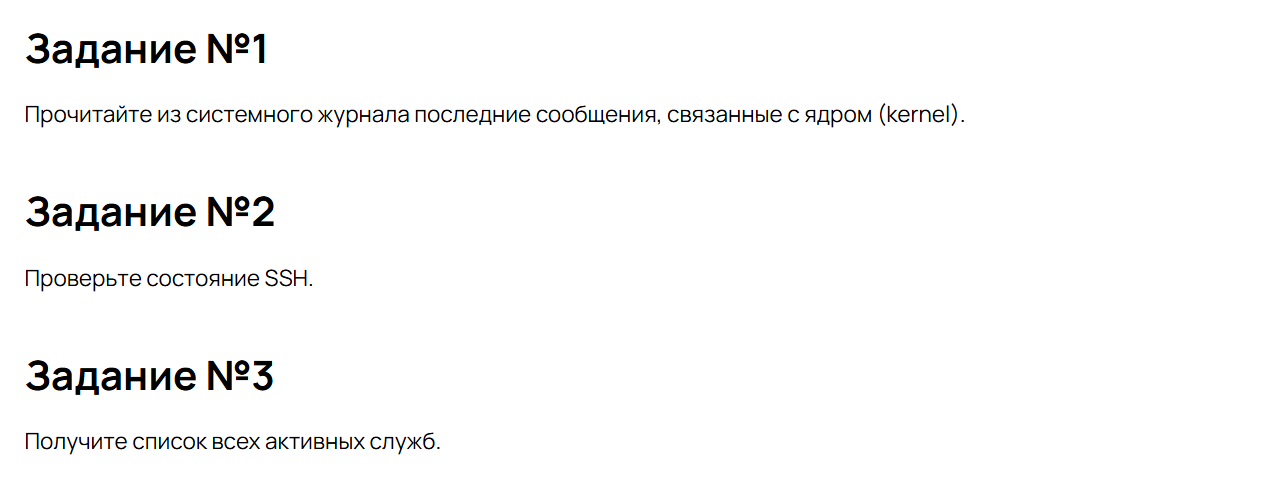


Рис. 14: «Роль командной строки в серверной среде». Условия заданий

#### 3.2.3.1 Задание №1

Воспользуемся утилитой для просмотра системного журнала

# -k сообщения, связанные с ядром  
journalctl -k

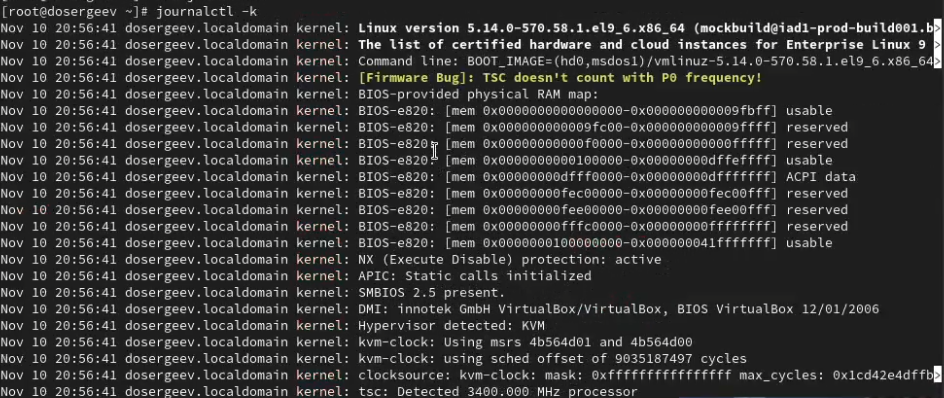


Рис. 15: Журнал с сообщениями ядра

#### 3.2.3.2 Задание №2

Посмотреть статус конкретной службы можно с помощью systemctl status:

systemctl status sshd

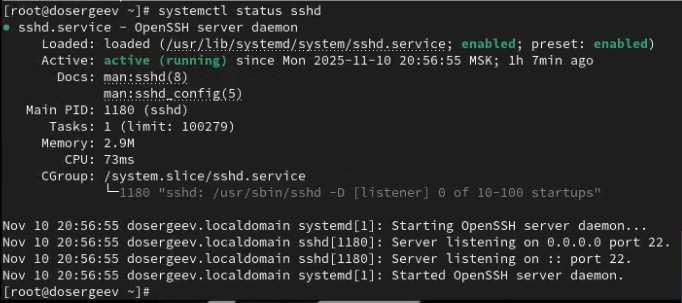


Рис. 16: Статус службы SSH

#### 3.2.3.3 Задание №3

Выведем список всех юнитов и отфильтруем, используя встроенные опции команды:

# list-units выводит состояние всех системных юнитов  
# --type=service указывает тип выводимых системных юнитов, в данном случае - сервис  
# --state=active указывает статус выводимых системных юнитов, в данном случае - активных  
systemctl list-units --type=service --state=active

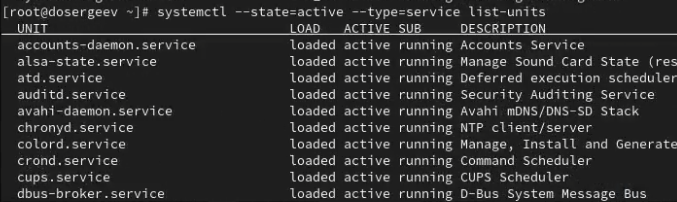


Рис. 17: Список всех активных служб

### 3.2.4 Тест по теме «Роль командной строки в серверной среде»

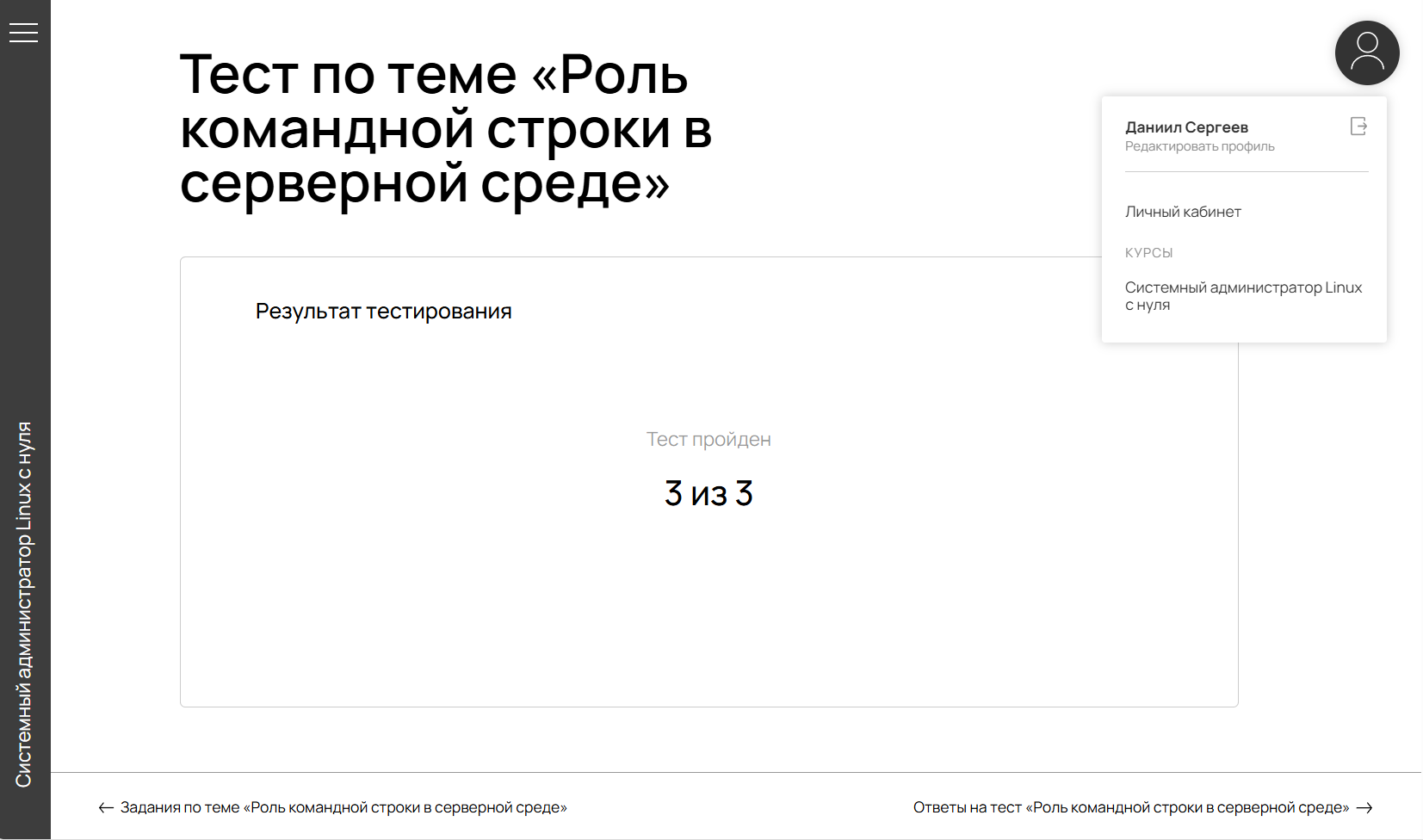


Рис. 18: Подтверждение прохождения теста «Роль командной строки в серверной среде»



Рис. 19: «Роль командной строки в серверной среде». Вопрос №1

Выбранный ответ: **Экономия ресурсов и повышение стабильности**.

Возможность запуска более одного окна существует и без графического интерфейса, например с помощью утилиты tmux. Взаимодействие с командной строкой не имеет ощутимых отличий без графического интерфейса.

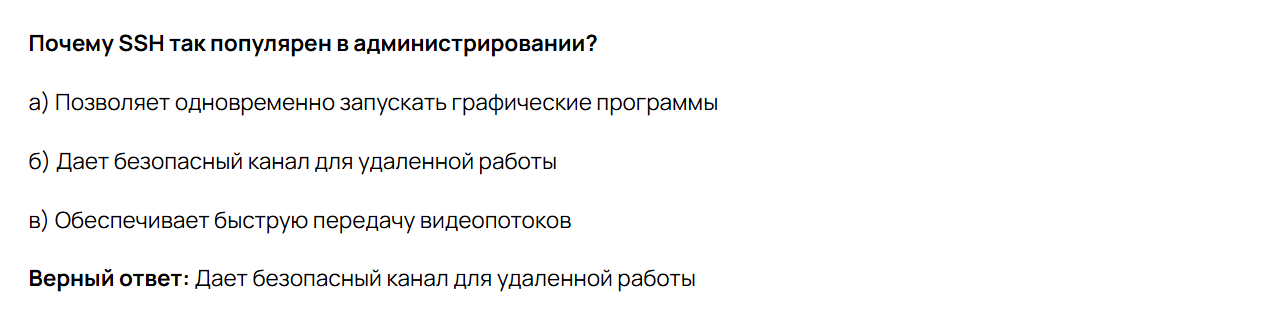


Рис. 20: «Роль командной строки в серверной среде». Вопрос №2

Выбранный ответ: **Дает безопасный канал для удаленной работы**.

SSH - это протокол прикладного уровня для безопасного обмена информацией между двумя устройствами. Он не отвечает за быструю передачу видеопотоков и не позволяет одновременно запускать графические программы.

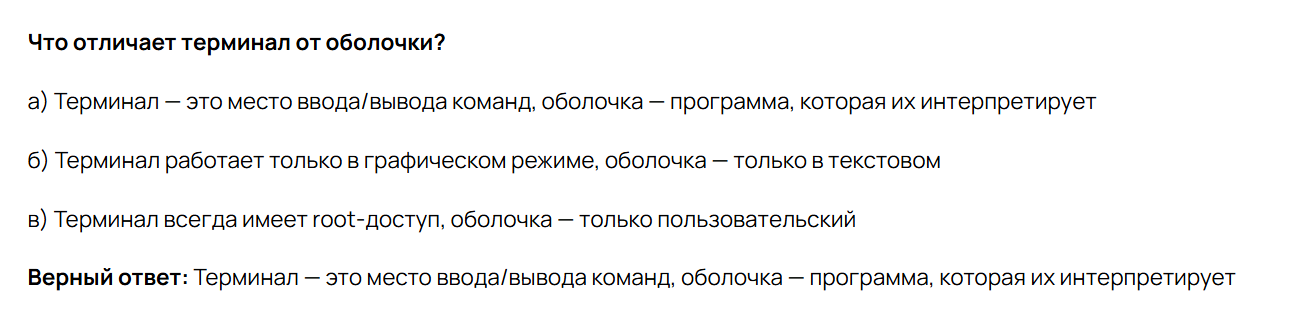


Рис. 21: «Роль командной строки в серверной среде». Вопрос №3

Выбранный ответ: **Терминал - это место ввода/вывода команд, оболочка - программа, которая их интерпретирует**.

Терминал является средством взаимодействие с компьютером посредством ввода текстовых команд.

Оболочка предназначена для неинтерактивной обработки команд.

### 3.2.5 Задания по теме «Файловая структура и ключевые каталоги». Часть 1

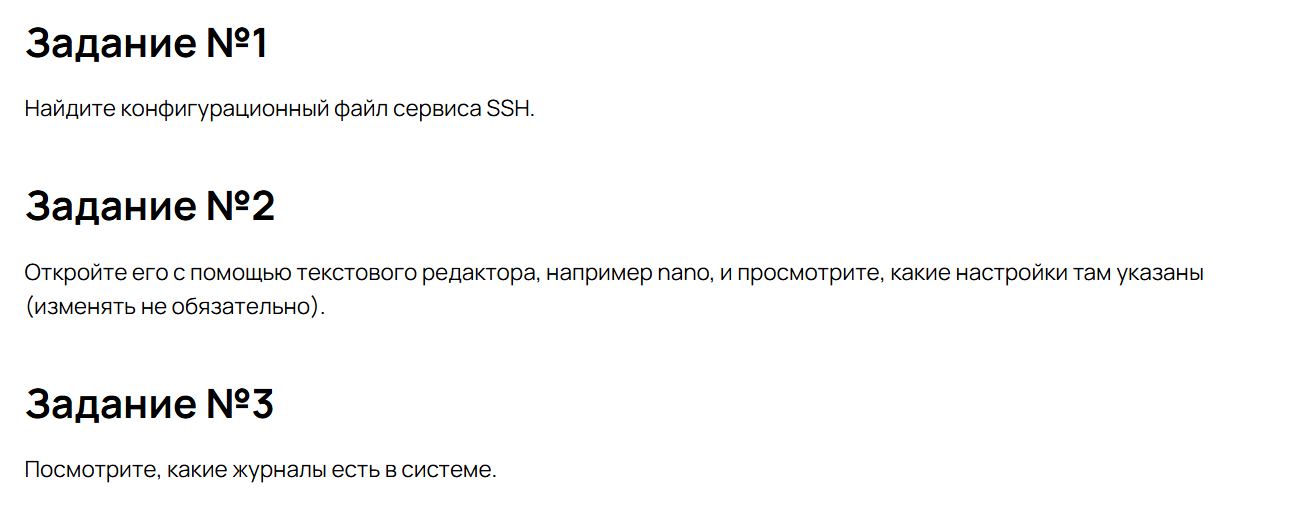


Рис. 22: «Файловая структура и ключевые каталоги». Условия заданий (1)

#### 3.2.5.1 Задание №1

Используем поиск по всем директориям в каталоге /etc, которые содержат в себе слово ssh

ls /etc/\*ssh\*

Опытным путем узнаем, что конфигурация сервиса ssh записана в файле ssh\_config.

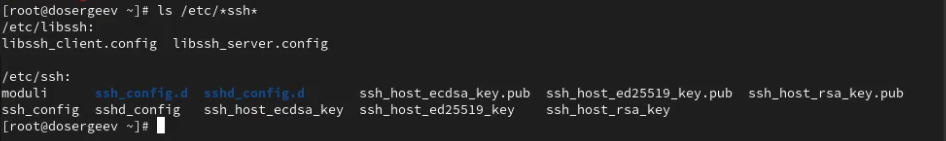


Рис. 23: Поиск конфигурационного файла SSH

#### 3.2.5.2 Задание №2

Откроем найденный файл в редакторе vi:

vi /etc/ssh/ssh\_config

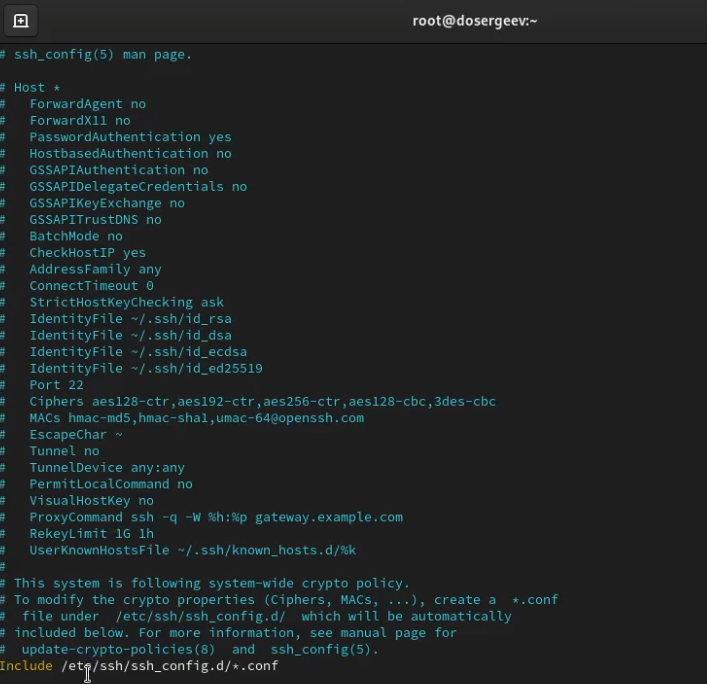


Рис. 24: Файл конфигурации SSH

#### 3.2.5.3 Задание №3

Файлы логов находятся в каталоге /var/log. Посмотрим, какие журналы в нем находятся:

ls /var/log

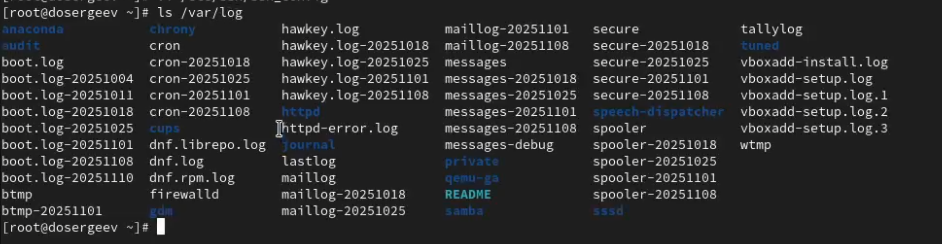


Рис. 25: Журналы каталога /var/log

Из всех перечисленных журналов стоит отметить

* /var/log/messages - содержит глобальные системные сообщения;
* /var/log/secure - хранит сообщения, связанные с безопасностью;
* /var/log/cron - регистрирует сообщения о выполнении заданий crontab;

### 3.2.6 Задания по теме «Файловая структура и ключевые каталоги». Часть 2

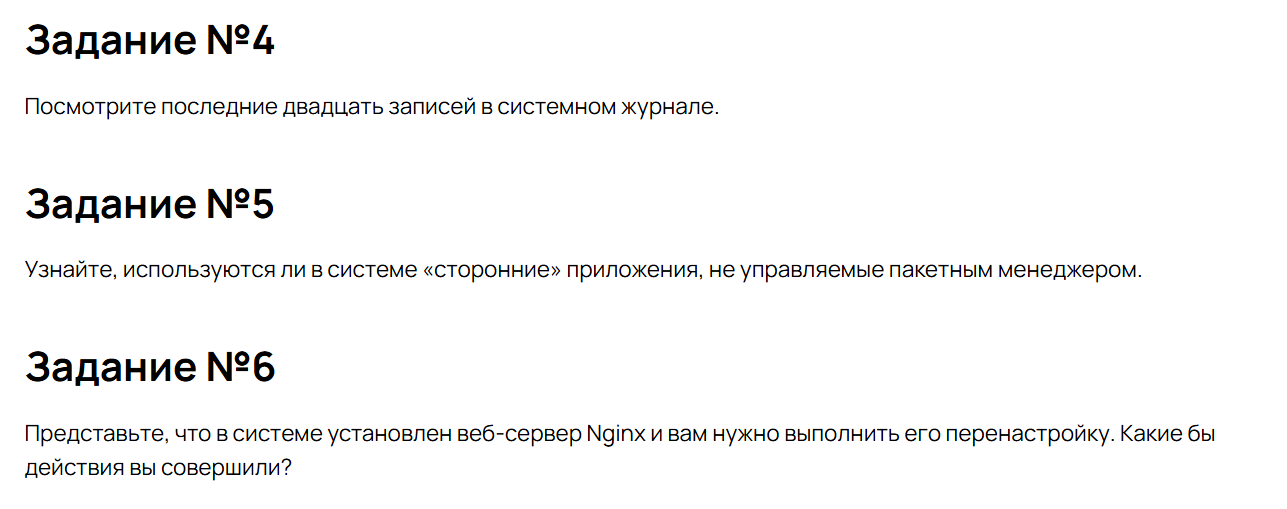


Рис. 26: «Файловая структура и ключевые каталоги». Условия заданий (2)

#### 3.2.6.1 Задание №4

Посмотрим последние двадцать записей

# -n 20 последние 20 строк  
journalctl -n 20

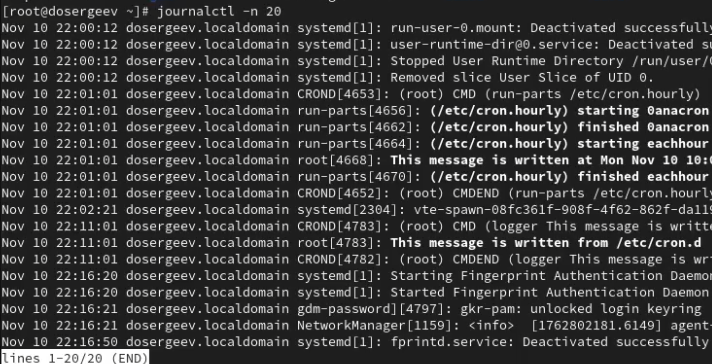


Рис. 27: Последние 20 записей journalctl

#### 3.2.6.2 Задание №5

Сторонние приложения, не используемые пакетным менеджером, находятся в каталоге /opt. Посмотрим его содержимое:

ls -l /opt

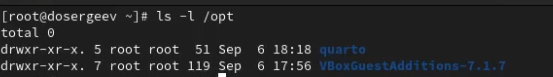


Рис. 28: Сторонние пакеты каталога /opt

В системе используются два сторонних приложения: quarto и VBoxGuestAdditions

#### 3.2.6.3 Задание №6

Необходимо перейти в каталог конфигурации nginx, сделать резервную копию файла с основными настройками и отредактировать его:

cd /etc/nginx  
ls -l  
sudo cp nginx.conf nginx.conf.backup  
sudo vi nginx.conf

Чтобы настройки вступили в силу, перезапускаем сервис

sudo systemctl reload nginx

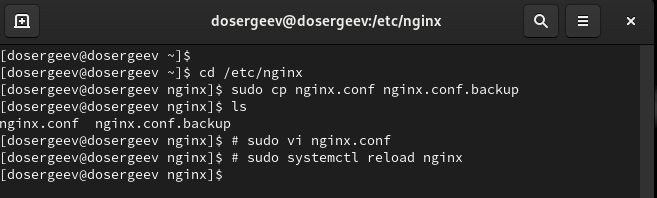


Рис. 29: Имитация процесса настройки nginx

### 3.2.7 Тест по теме «Файловая структура и ключевые каталоги»

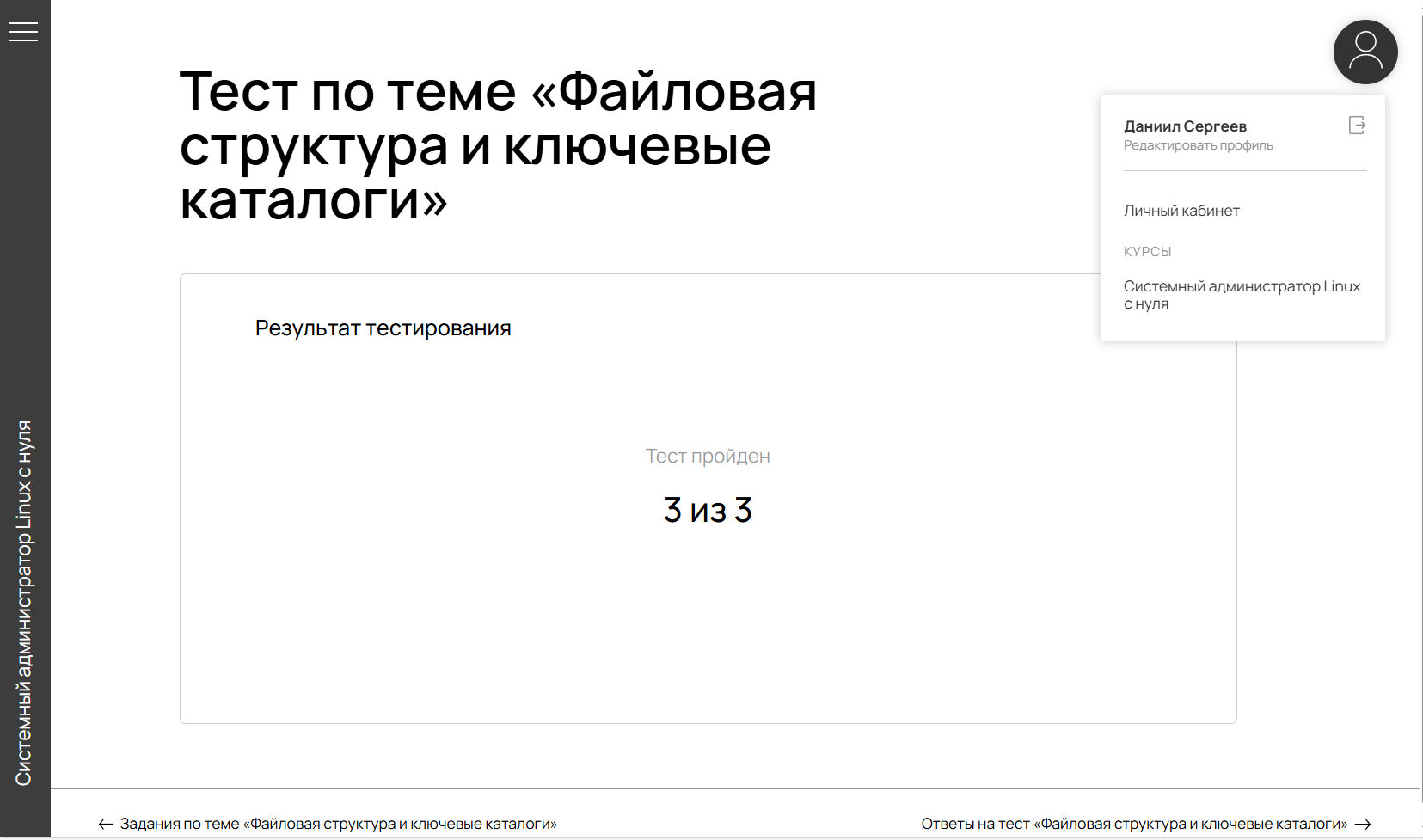


Рис. 30: Подтверждение прохождения теста «Файловая структура и ключевые каталоги»

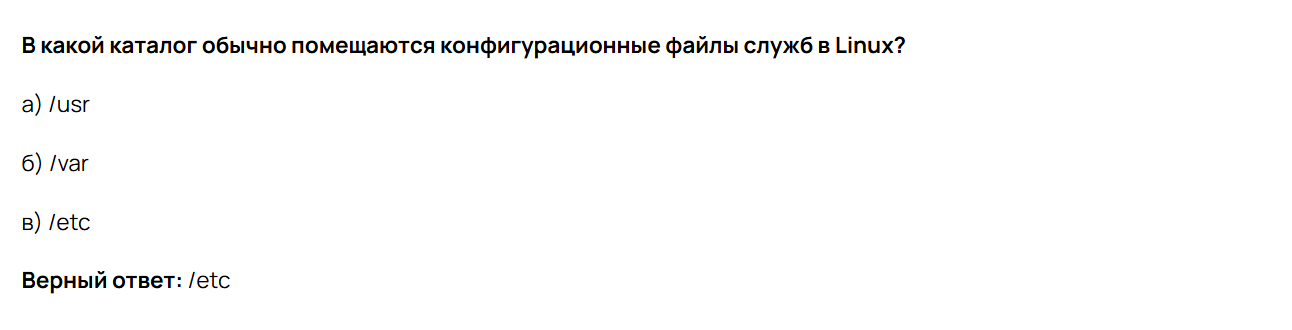


Рис. 31: «Файловая структура и ключевые каталоги». Вопрос №1

Выбранный ответ: **/etc**.

* /usr - каталог для содержания пользовательских приложений и данных, не необходимых для загрузки системы;
* /var - каталог для хранения изменяемых файлов (например, логов);
* /etc - каталог для хранения конфигурационных файлов служб;

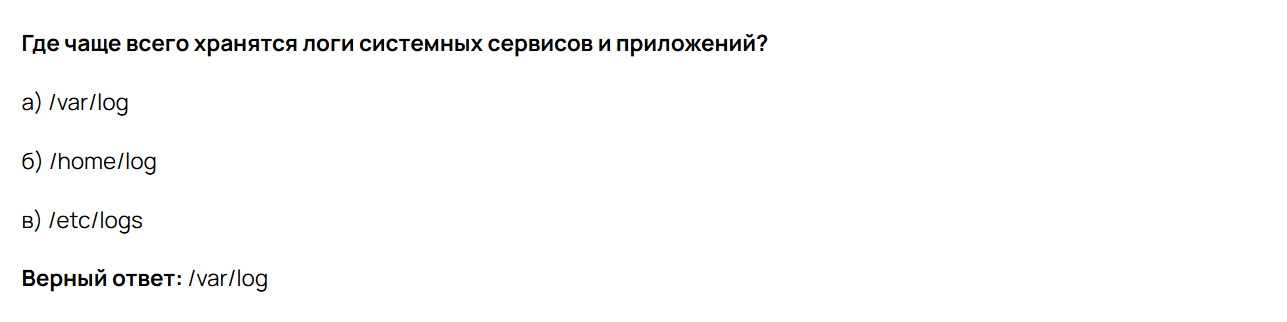


Рис. 32: «Файловая структура и ключевые каталоги». Вопрос №2

Выбранный ответ: **/var/log**.

* /home/log - домашний каталог log (либо для пользователя log, либо для других пользователей)
* /etc/logs - обычно такого каталога не существует, так как логи хранятся в /var.

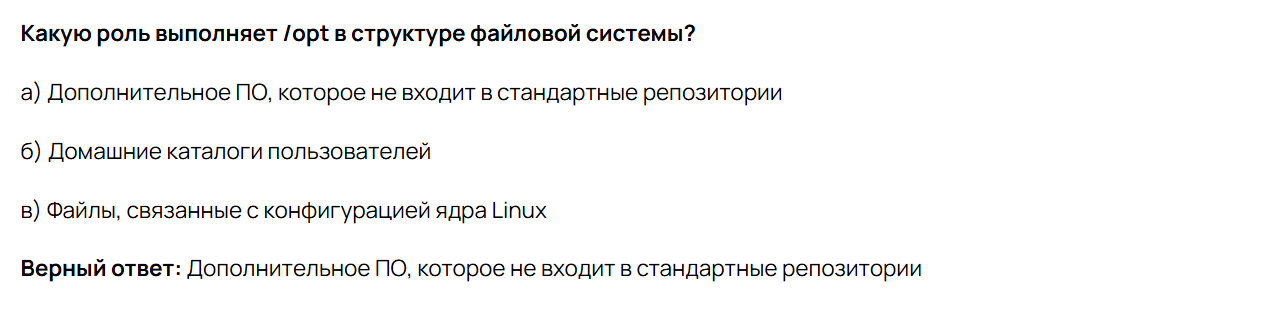


Рис. 33: «Файловая структура и ключевые каталоги». Вопрос №3

Выбранный ответ: **Дополонительное ПО, которое не входит в стандартные репозитории**.

* Домашние каталоги пользователей хранятся в каталоге /home
* Файлы, связанные с конфигурацией ядра Linux хранятся в /etc или /boot

### 3.2.8 Задания по теме «Принцип „Всё есть файл“»

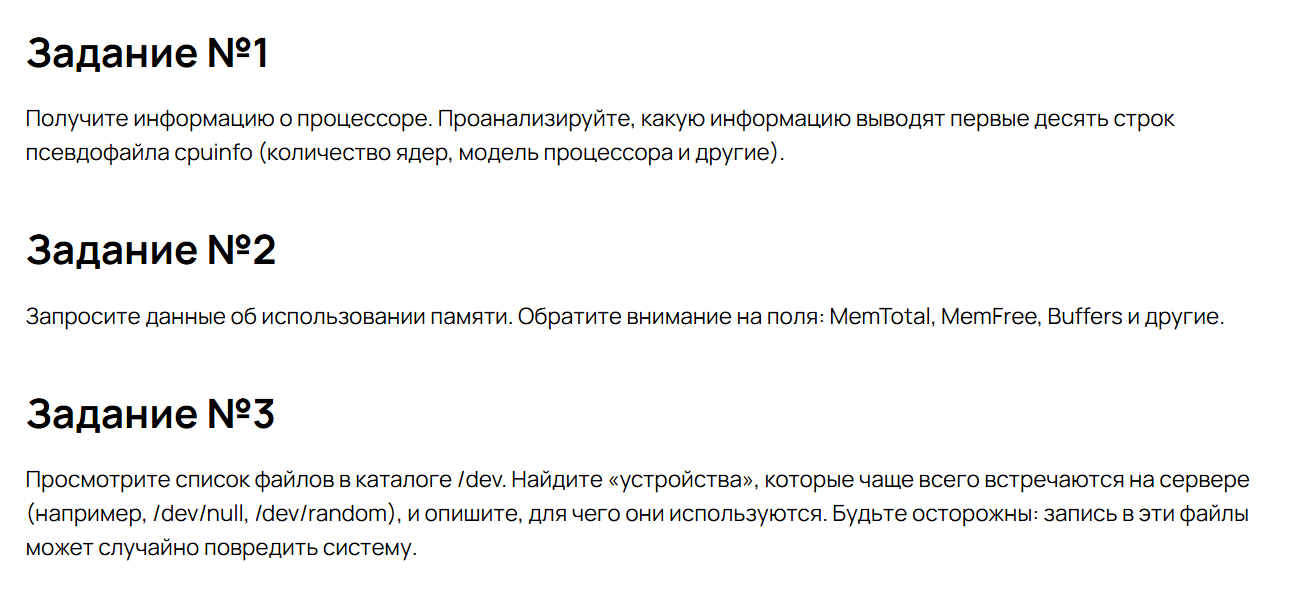


Рис. 34: «Принцип „Всё есть файл“». Условия заданий

#### 3.2.8.1 Задание №1

Выполним команду

cat /proc/cpuinfo | head

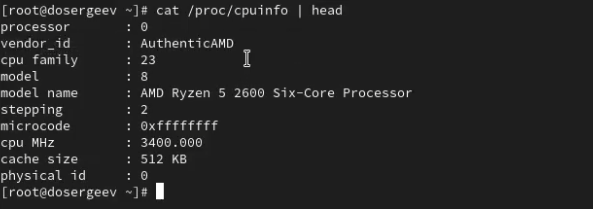


Рис. 35: Просмотр информации о процессоре

* processor : Номер процессора;
* vendor\_id : Идентификатор производителя;
* cpu family : Семейство процессоров;
* model : Модель;
* model name : Название модели;
* stepping : Степпинг процессора (номер версии ядра);
* microcode : Набор инструкций процессора;
* cpu MHz : Частота процессора;
* cache size : Размер КЭШа;
* physical id: Физический идентификатор процессора;

#### 3.2.8.2 Задание №2

Откроем файл meminfo, в котором хранятся данные об использовании памяти

cat /proc/meminfo

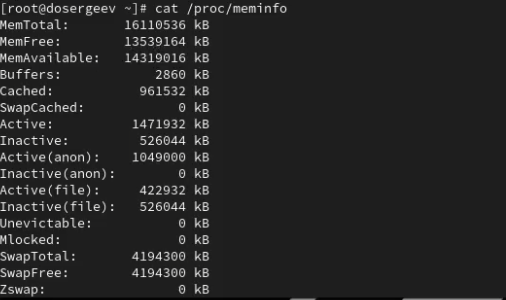


Рис. 36: Просмотр информации о процессоре

Обратим внимание на поля

* MemTotal - Всего памяти, примерно 16 Гб;
* MemFree - Свободно памяти, примерно 13 Гб;
* Buffers - Память, используемая буферами ядра, примерно 2,8 мб;

#### 3.2.8.3 Задание №3

Просмотрим список файлов

ls -l /dev

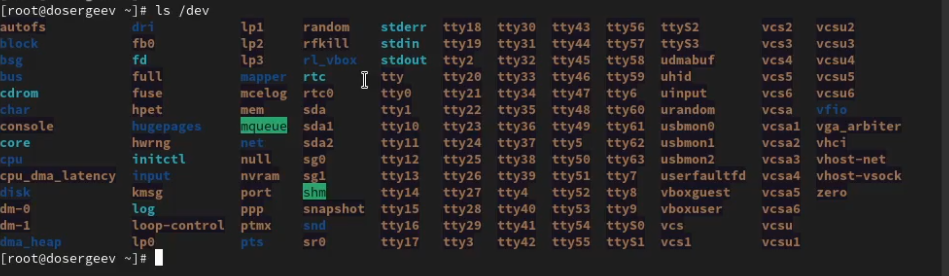


Рис. 37: Просмотр каталога /dev

* /dev/null - корзина для данных, что удобно для отбрасывания ненужного вывода;
* /dev/random - генератор псевдослучайных чисел;

### 3.2.9 Тест по теме «Принцип „Всё есть файл“»

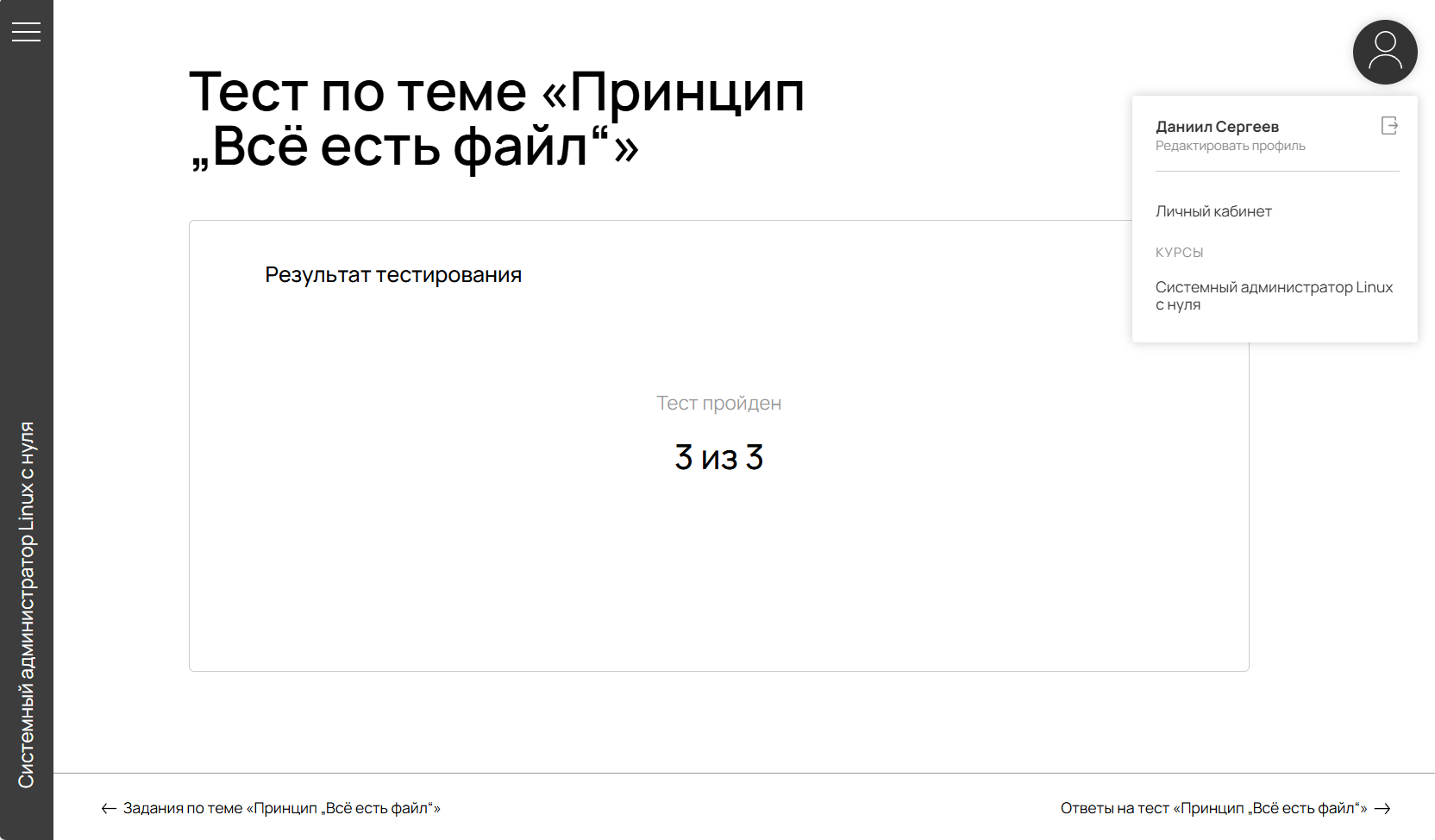


Рис. 38: Подтверждение прохождения теста «Принцип „Всё есть файл“»

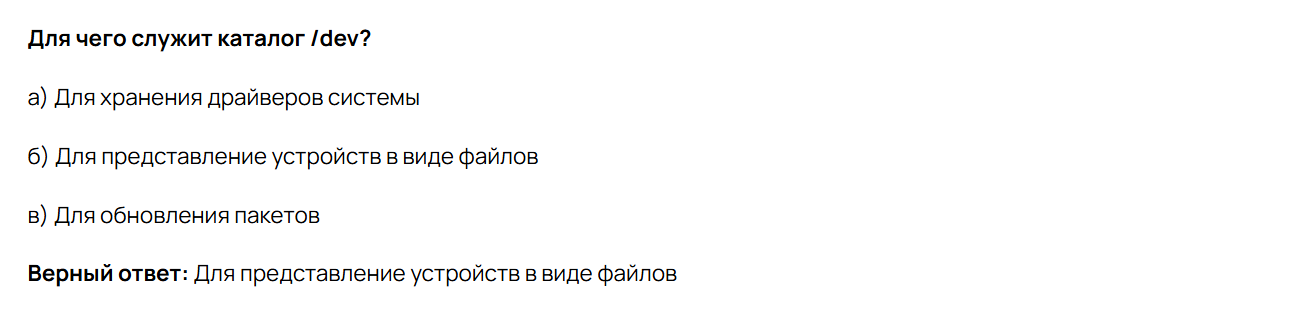


Рис. 39: «Принцип „Всё есть файл“». Вопрос №1

Выбранный ответ: **Для представления устройств в виде файлов**.

В директории /dev расположены псевдофайлы, которые представляют собой интерфейс к аппаратному и виртуальному оборудованию. Драйвера системы вероятно хранятся в /lib/modules.

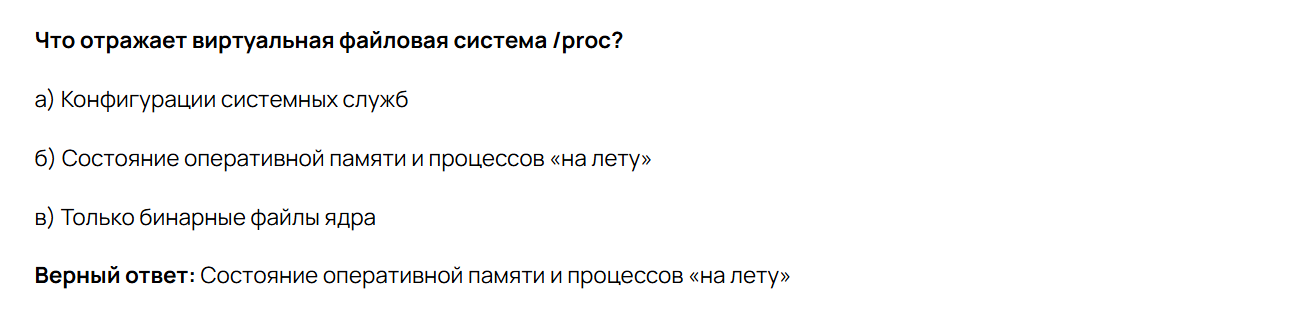


Рис. 40: «Принцип „Всё есть файл“». Вопрос №2

Выбранный ответ: **Состояние оперативной памяти и процессоров «на лету»**.

* Конфигурации системных служб находятся в каталоге /etc
* Бинарные файлы ядра находятся в каталогах /bin и /sbin

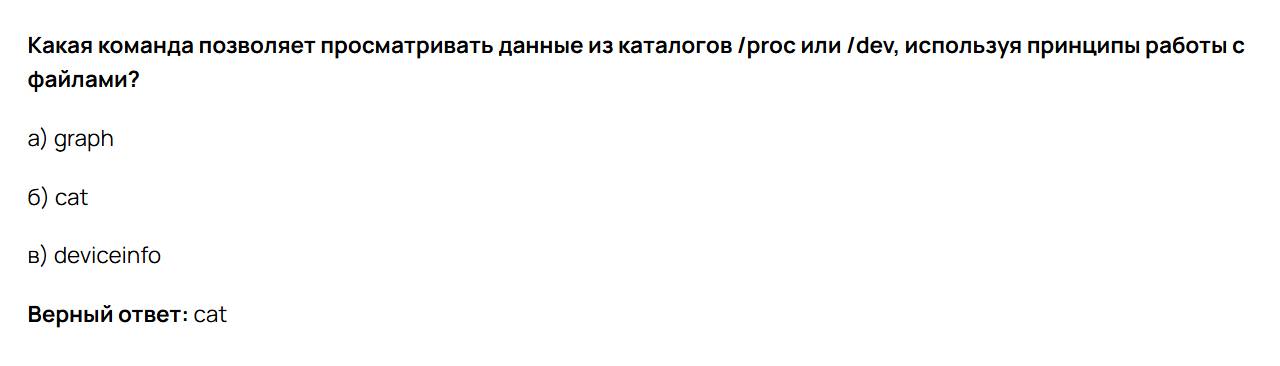


Рис. 41: «Принцип „Всё есть файл“». Вопрос №3

Выбранный ответ: **cat**.

Команда cat олицетворяет принцип работы с файлами, позволяя отфильтровать вывод содержимого файла через конвейер по своему усмотрению.

### 3.2.10 Задания по теме «Работа с базовыми командами в реальных условиях»

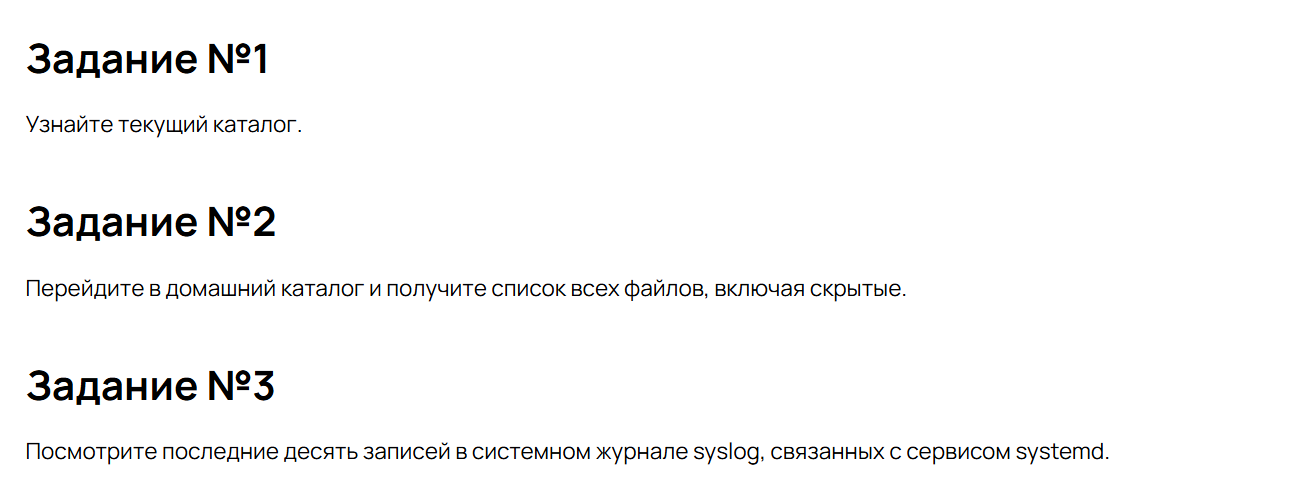


Рис. 42: «Работа с базовыми командами в реальных условиях». Условия заданий

#### 3.2.10.1 Задание №1

Текущей каталог доступен командой pwd.

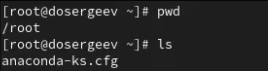


Рис. 43: Текущий каталог

#### 3.2.10.2 Задание №2

Узнаем список всех файлов в домашнем каталоге, включая скрытые:

cd ~  
# -a списох всех файлов, включая скрытые  
ls -a

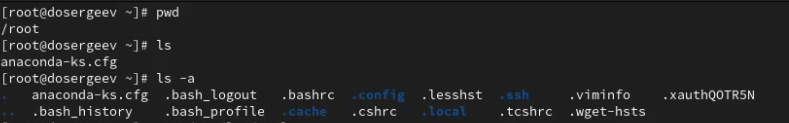


Рис. 44: Скрытые файлы домашнего каталога

#### 3.2.10.3 Задание №3

Так как syslog отсутствует на Rocky Linux, посмотрим в лог messages. Вывод строк передадим сначала в grep, потом в tail (по умолчанию выводит последние 10 строк. Опцию -n использовать не будем).

cat /var/log/messages | grep systemd | tail

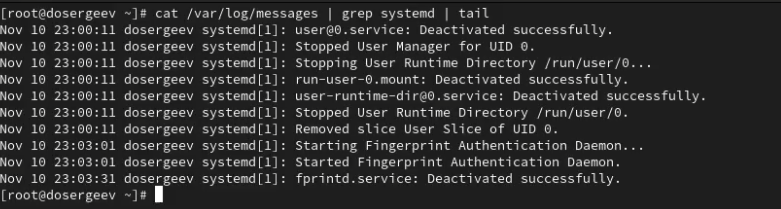


Рис. 45: Последние 10 записей messages, связаных с сервисом systemd

### 3.2.11 Тест по теме «Работа с базовыми командами в реальных условиях»

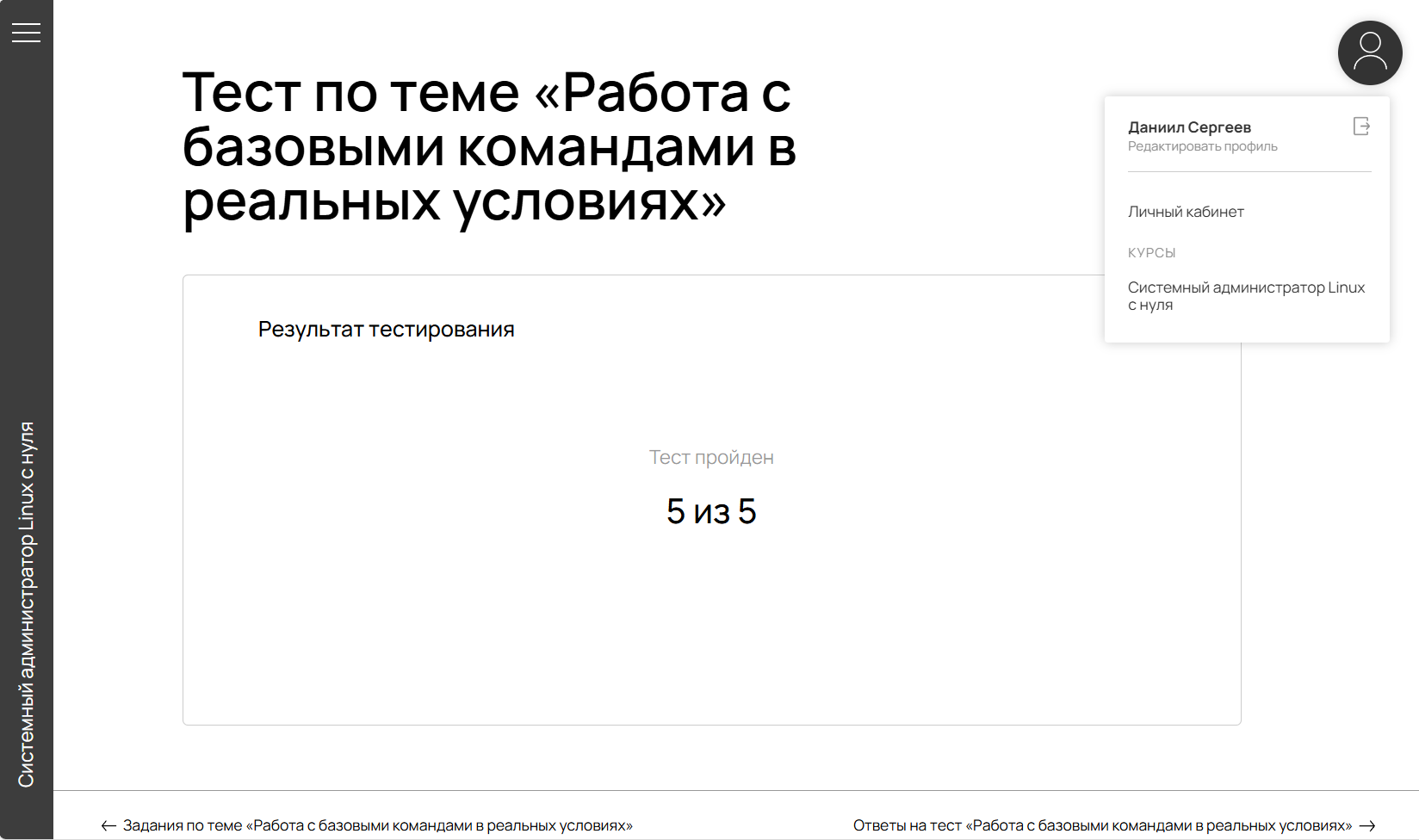


Рис. 46: Подтверждение прохождения теста «Работа с базовыми командами в реальных условиях»

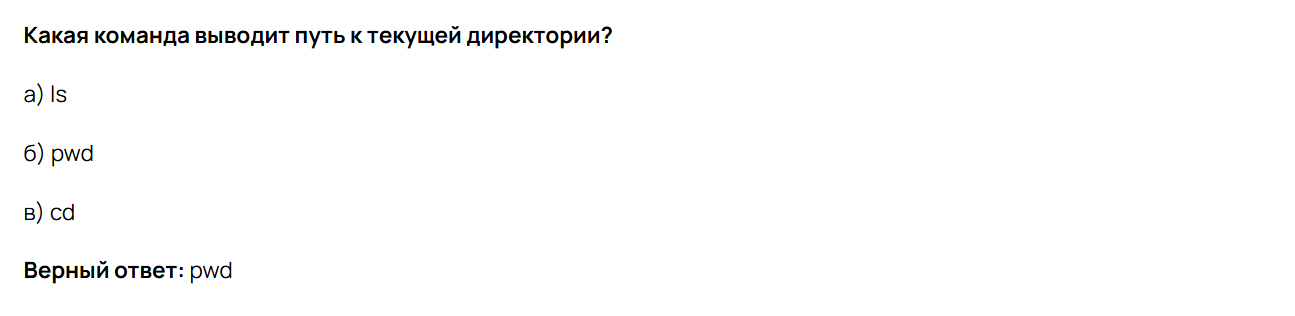


Рис. 47: «Работа с базовыми командами в реальных условиях». Вопрос №1

Выбранный ответ: **pwd**.

* ls - выводит список файлов и каталогов в текущей директории;
* cd - позволяет изменить каталог на другой по указанному пути;

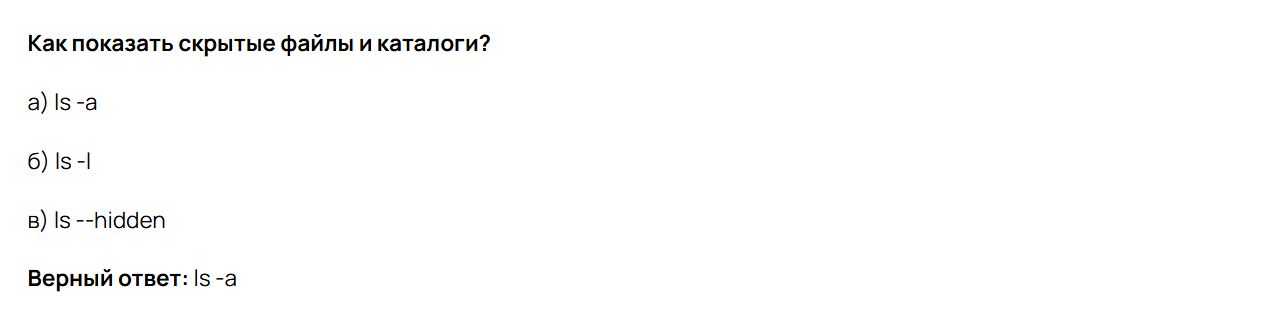


Рис. 48: «Работа с базовыми командами в реальных условиях». Вопрос №2

Выбранный ответ: **ls -a**.

* ls -l - выводит дополнительную информацию о файлах (владелец, группа, права, размер и так далее);
* ls –hidden - данной опции не существует;

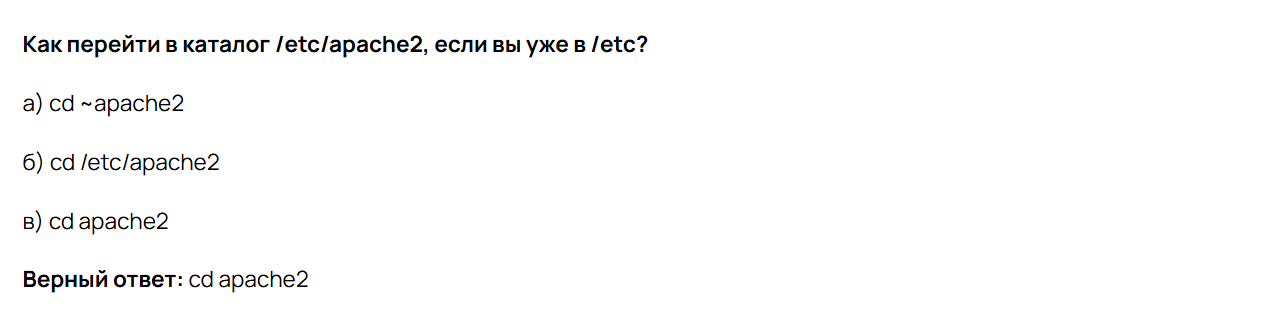


Рис. 49: «Работа с базовыми командами в реальных условиях». Вопрос №3

Выбранный ответ: **cd apache2**.

* cd ~apache2 - перейдет в каталог /etc/~apache2. Отдельный символ ~ обозначает домашнюю директорию;
* cd /etc/apache2 - альтернативный способ с использованием полного пути, но лучше использовать относительный путь;

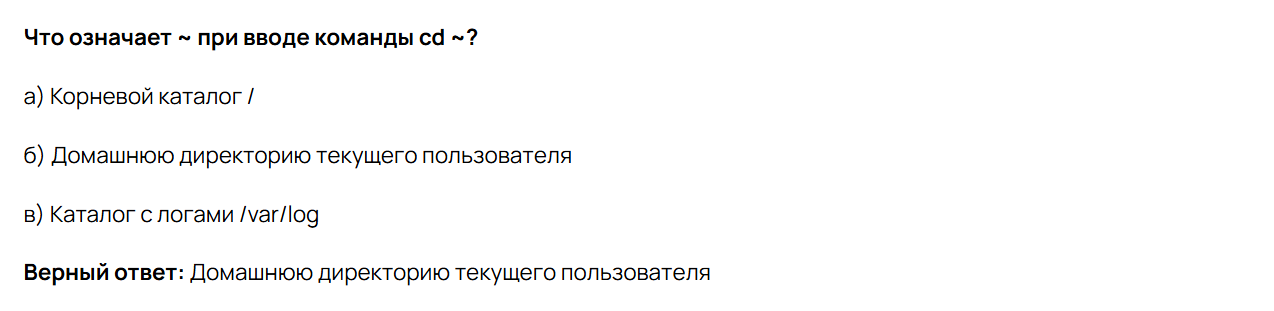


Рис. 50: «Работа с базовыми командами в реальных условиях». Вопрос №4

Выбранный ответ: **Домашнюю директорию текущего пользователя**.

Как было сказано ранее, отдельный символ ~ обозначает домашнюю директорию. Корневой каталог начинается с / (/etc, /home, /usr), а для каталога с логами нет конкретного символа, необходимо писать полный или относительный путь.

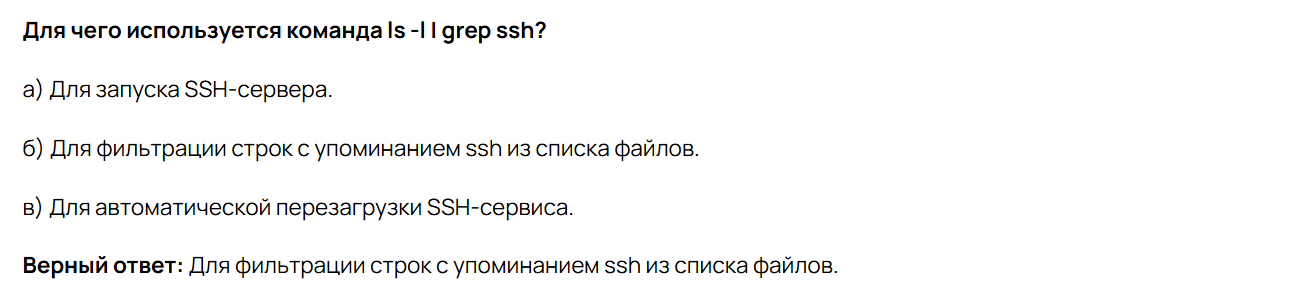


Рис. 51: «Работа с базовыми командами в реальных условиях». Вопрос №5

Выбранный ответ: **Для фильтрации строк с упоминанием ssh из списка файлов**.

Команда ls -l выведет список файлов в текущей директории с дополнительной информацией. Вывод перехватится конвейером, который передаст его в команду grep. В качетсве шаблона в фильтр передается слово ssh, что выведет все строки ls -l, в которых есть упоминание ssh.

## 3.3 Модуль 3. Файлы и каталоги

### 3.3.1 Задания по теме «Базовые команды для управления файлами и каталогами»

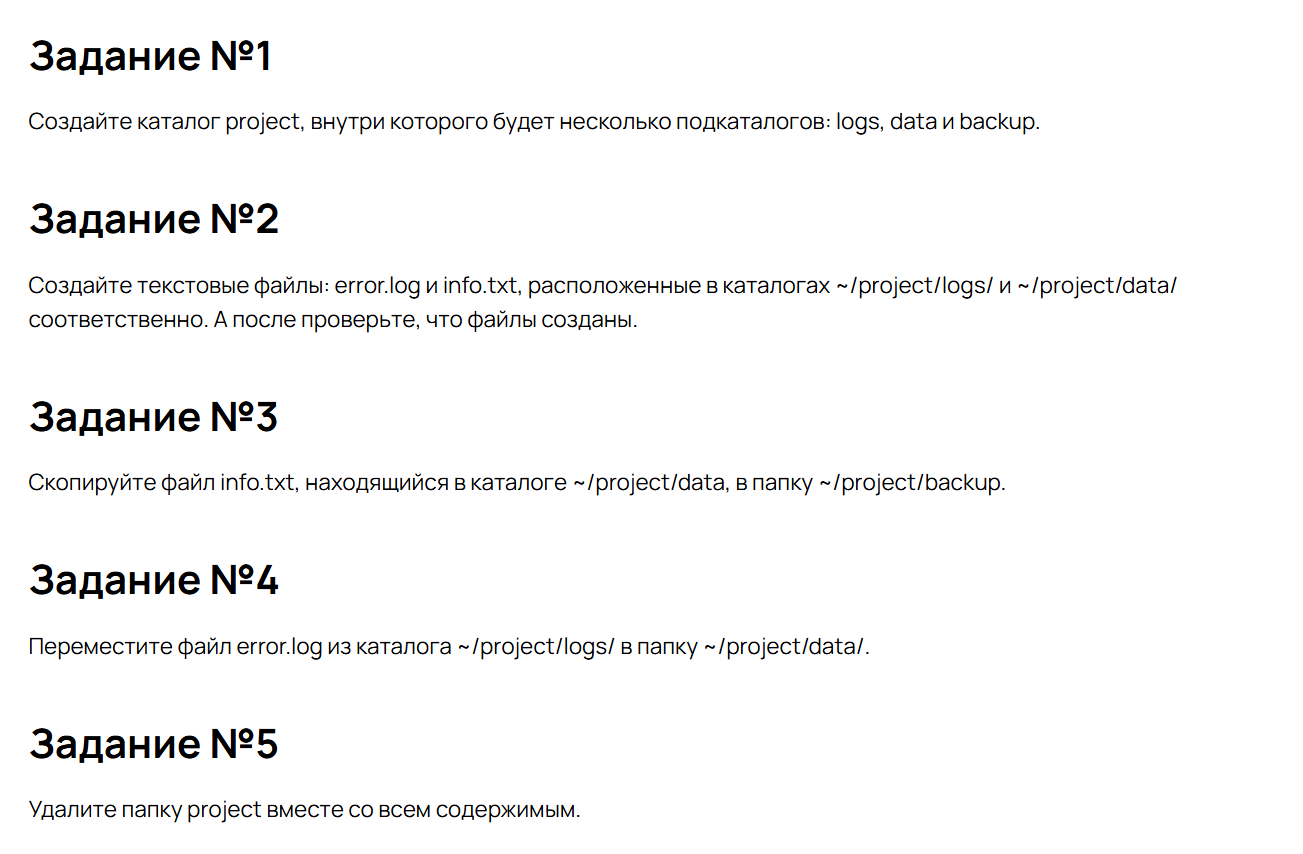


Рис. 52: «Базовые команды для управления файлами и каталогами». Условия заданий

#### 3.3.1.1 Задание №1

Создадим каталоги и сделаем проверку

mkdir -p ~/project/{logs,data,backup}  
ls

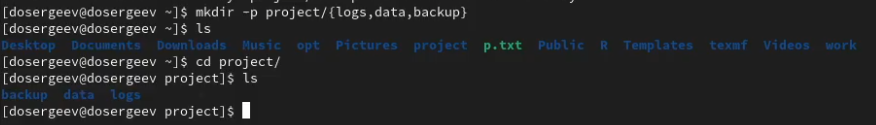


Рис. 53: Создание каталога и его подкаталогов одной командой

#### 3.3.1.2 Задание №2

Создадим файлы и проверим их наличие.

touch ~/project/logs/error.log ~/project/data/info.txt  
ls ~/project/\*

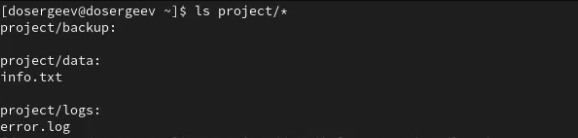


Рис. 54: Проверка наличия error.log и info.txt

#### 3.3.1.3 Задание №3

Скопируем файл в каталог logs (опечатка), проверим наличие файла.

cp ~/project/data/info.txt ~/project/logs/

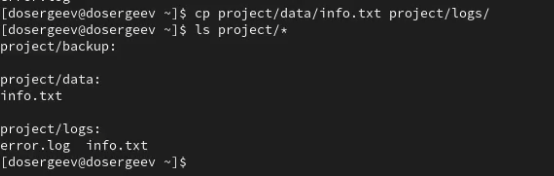


Рис. 55: Копирование файла из одного каталога в другой

#### 3.3.1.4 Задание №4

Переместим файл error.log из каталога project/logs/ в папку project/data/.

mv ~/project/logs/error.log ~/project/data/

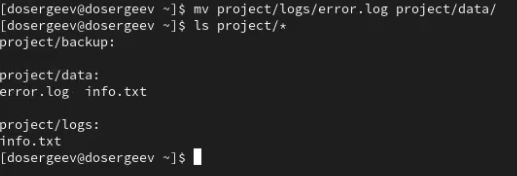


Рис. 56: Перемещение файлов

#### 3.3.1.5 Задание №5

Удалим все созданные файлы

rm -r ~/project

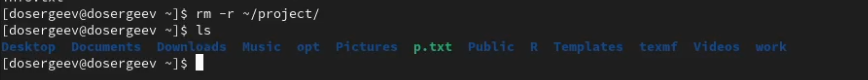


Рис. 57: Удаление файлов и самого каталога project

### 3.3.2 Задания по теме «Работа с архивами и передача данных между серверами»

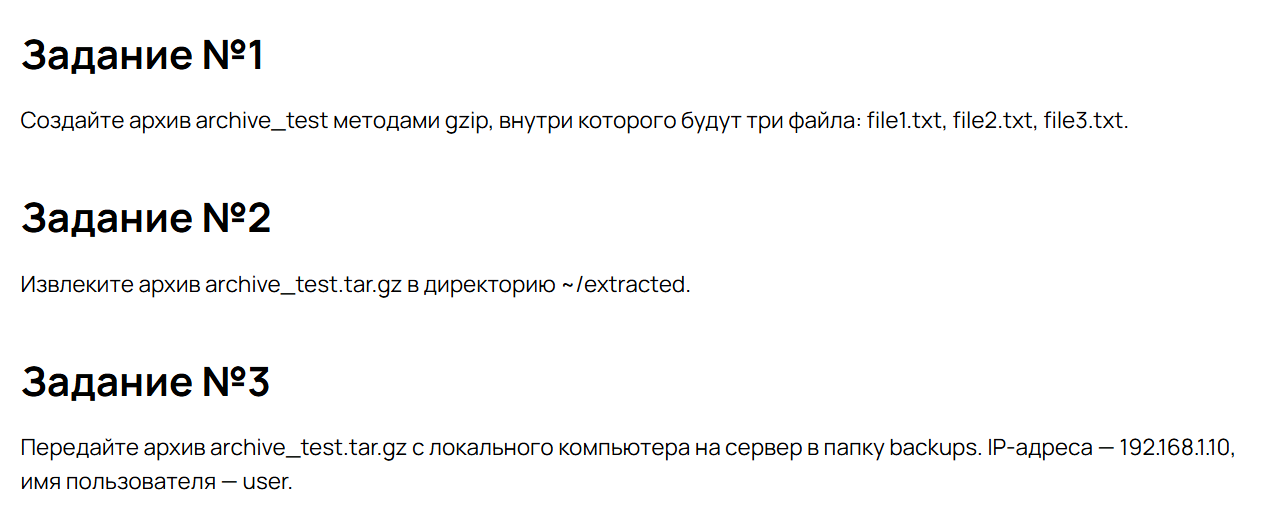


Рис. 58: «Работа с архивами и передача данных между серверами». Условия заданий

#### 3.3.2.1 Задание №1

Создаем временный каталог для выполнения задания и файлы для архивации

mkdir course  
cd course  
touch file{1..3}.txt

Архивируем все файлы в каталоге course

# -c создание архива  
# -z способ архивации gzip  
# -v подробный вывод выполнения команды  
# -f указание названия архива  
tar -czvf archive\_test.tar.gz ./\*

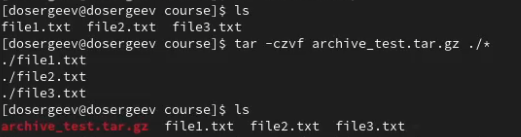


Рис. 59: Архивация файлов file

#### 3.3.2.2 Задание №2

Создадим каталог extracted и извлечем туда файла архива

mkdir extracted   
tar -xzvf archive\_test.tar.gz -C extracted/  
ls extracted/

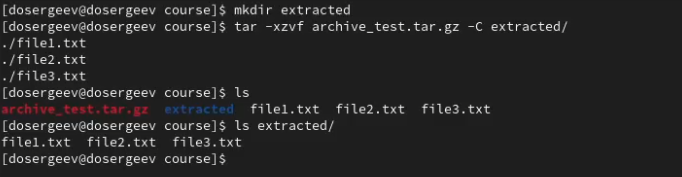


Рис. 60: Извлечение файлов из архива

#### 3.3.2.3 Задание №3

В связи с отсутствием доступа к серверу selectel, запустим локальный сервер apache через службу httpd.

systemctl start httpd  
systemctl status httpd

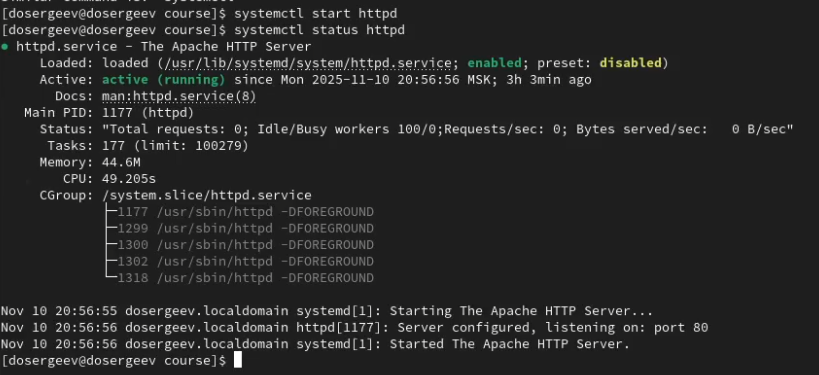


Рис. 61: Запуск и проверка службы httpd

Теперь попробуем скопировать архив “на сервер” и обратно

scp archive\_test.tar.gz root@localhost:/backups  
rm archive\_test.tar.gz  
ls  
scp root@localhost:/backups/archive\_test.tar.gz  
ls

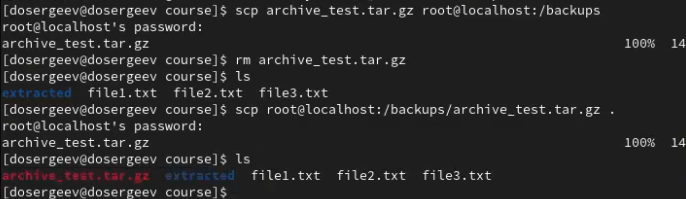


Рис. 62: Практика команды scp

### 3.3.3 Тест по теме «Работа с архивами и передача данных между серверами»

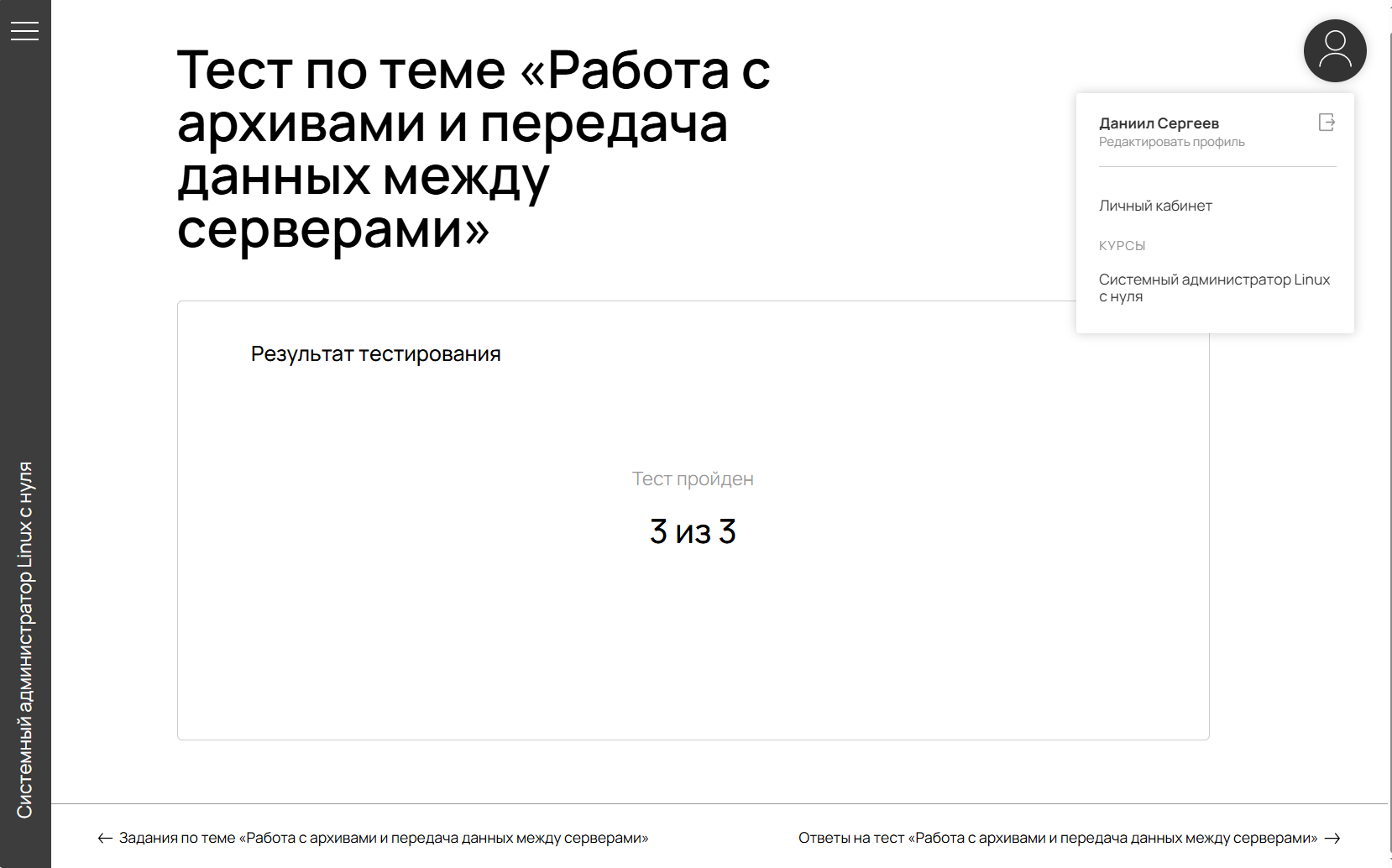


Рис. 63: Подтверждение прохождения теста «Работа с архивами и передача данных между серверами»

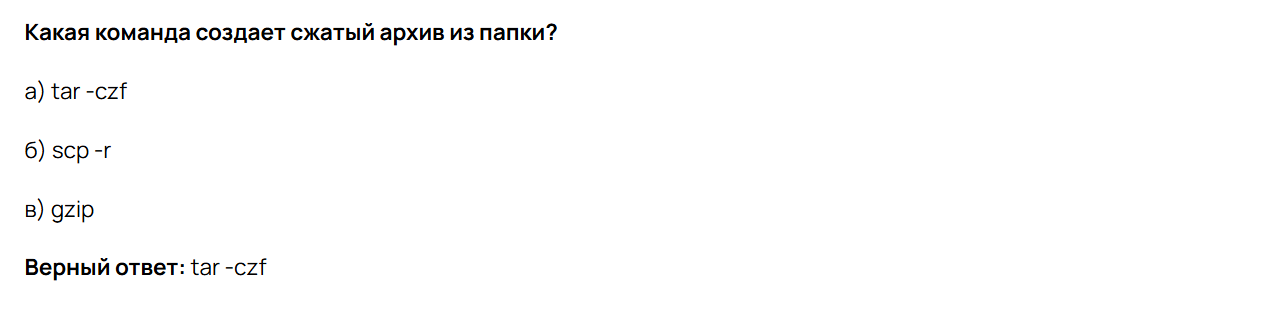


Рис. 64: «Работа с архивами и передача данных между серверами». Вопрос №1

Выбранный ответ: **tar -czf**.

* scp -r - рекурсивная (-r) передача файлов между серверами;
* gzip - сжатие одного файла;



Рис. 65: «Работа с архивами и передача данных между серверами». Вопрос №2

Выбранный ответ: **tar -xzf**.

* scp -xzf - команда scp не отвечает за сжатие и не имеет данных опций либо они выполняют другие действия;
* gzip -d - распаковать один файл;

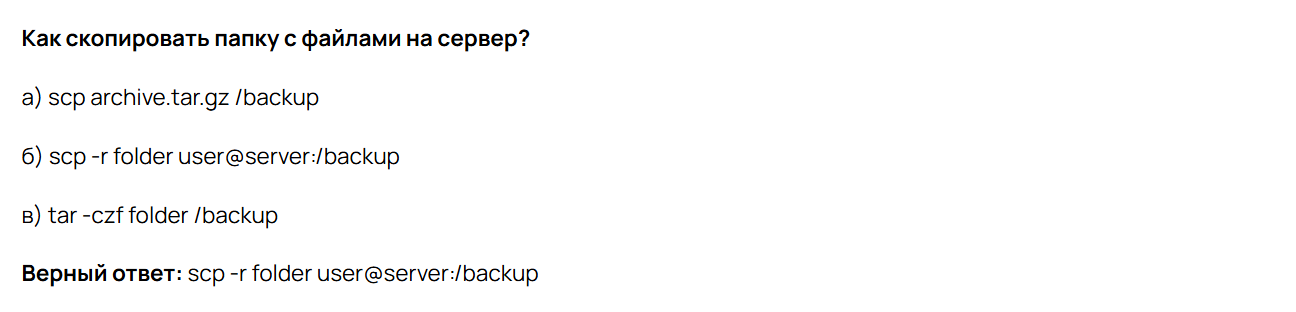


Рис. 66: «Работа с архивами и передача данных между серверами». Вопрос №3

Выбранный ответ: **scp -r folder user@server:/backup**.

* tar -czf folder /backup - архивирование каталога /backup в архив folder;
* scp archive.tar.gz /backup - копирование архива, а не папки с файлами;

### 3.3.4 Задания по теме «Основы автоматизации с помощью Bash-скриптов»

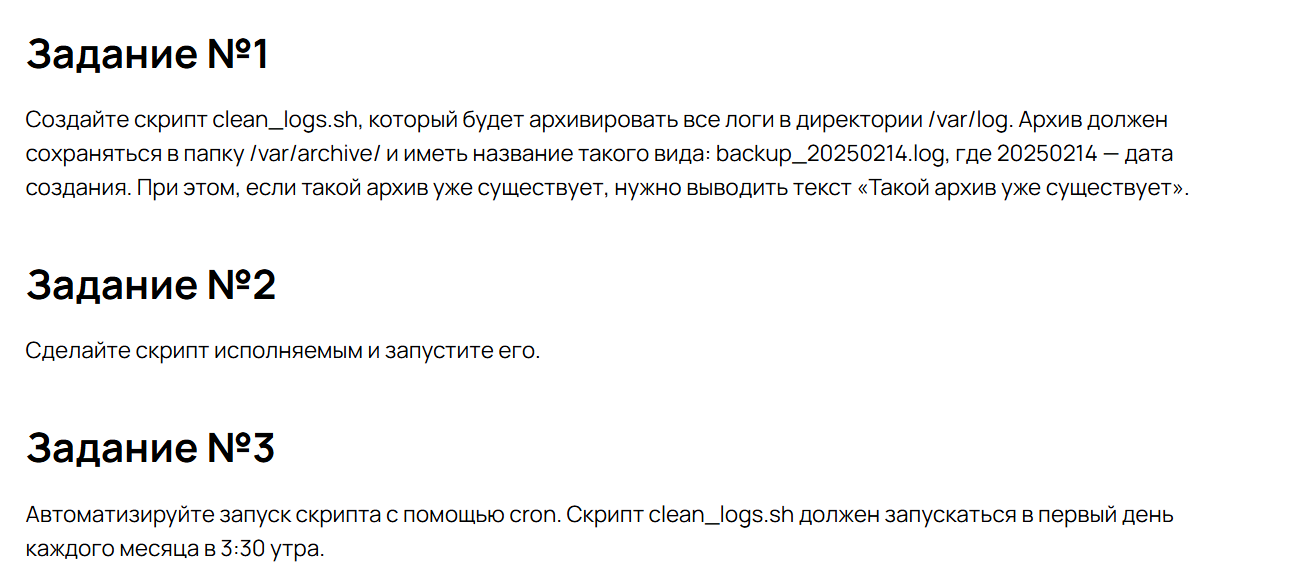


Рис. 67: «Основы автоматизации с помощью Bash-скриптов». Условия заданий

#### 3.3.4.1 Задание №1

Напишем скрпт, который будет архивировать все логи в директории /var/log, согласно условию задания

#!/bin/bash  
  
LOG\_DIR=/var/log  
ARCHIVE\_DIR=/var/archive  
  
DATE=$(date +%Y%m%d)  
ARCHIVE\_NAME="backup\_${DATE}.log"  
  
if [ -e "$ARCHIVE\_DIR/$ARCHIVE\_NAME" ]; then  
 echo "Такой архив уже существует"  
 exit 0  
fi  
  
tar -czf $ARCHIVE\_DIR/$ARCHIVE\_NAME $LOG\_DIR

#### 3.3.4.2 Задание №2

Теперь сделаем наш скрипт исполняемым и запустим его.

chmod +x bin/clean\_logs.sh  
./clean\_logs.sh

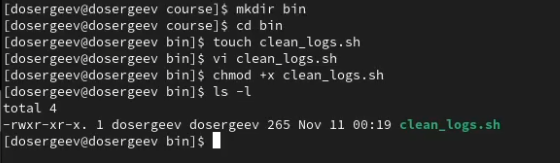


Рис. 68: Редактирование прав доступа clean\_logs.sh

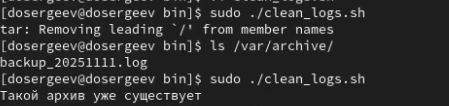


Рис. 69: Выполнение файла clean\_logs.sh

#### 3.3.4.3 Задание №3

Автоматизируем запуск скрипта с помощью cron.

crontab -e

Зададим срок выполнения: 3:30 утра каждого первого дня месяца bash 30 3 1 \* \* /home/dosergeev/course/bin/clean\_logs.shbash



Рис. 70: Настройка crontab -e

### 3.3.5 Тест по теме «Основы автоматизации с помощью Bash-скриптов»

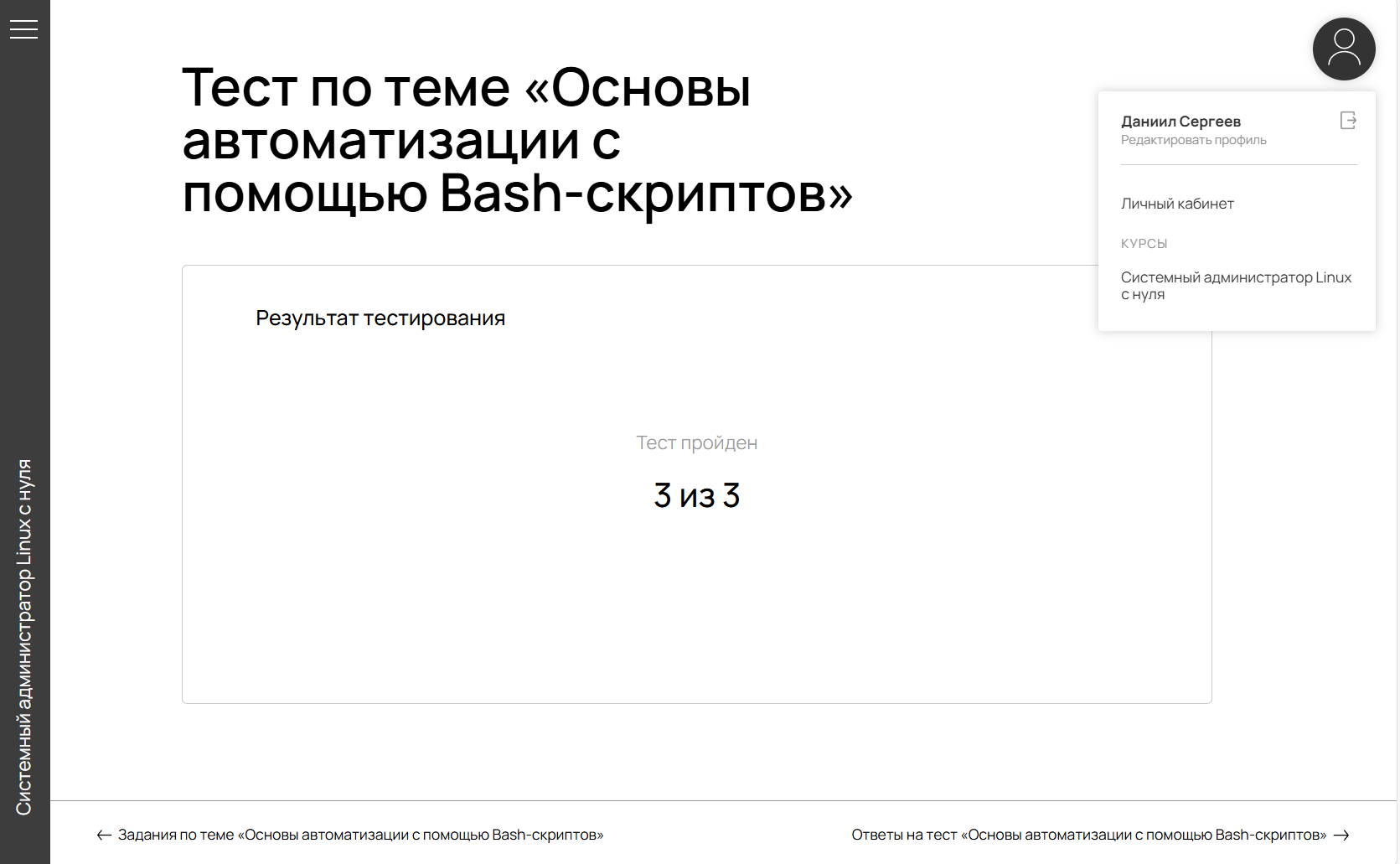


Рис. 71: Подтверждение прохождения теста «Основы автоматизации с помощью Bash-скриптов»



Рис. 72: «Основы автоматизации с помощью Bash-скриптов». Вопрос №1

Выбранный ответ: **chmod +x**.

* chmod +r - добавить право на чтение файла;
* chmod +w - добавить право на запись в файл;



Рис. 73: «Основы автоматизации с помощью Bash-скриптов». Вопрос №2

Выбранный ответ: **#**.

Чтобы добавить комментарий, достаточно поставить знак решетки перед нужной строкой. Однако если в начале прописать заголовок #!/bin/bash, то это укажет скрипту, что нужно использовать Bash.

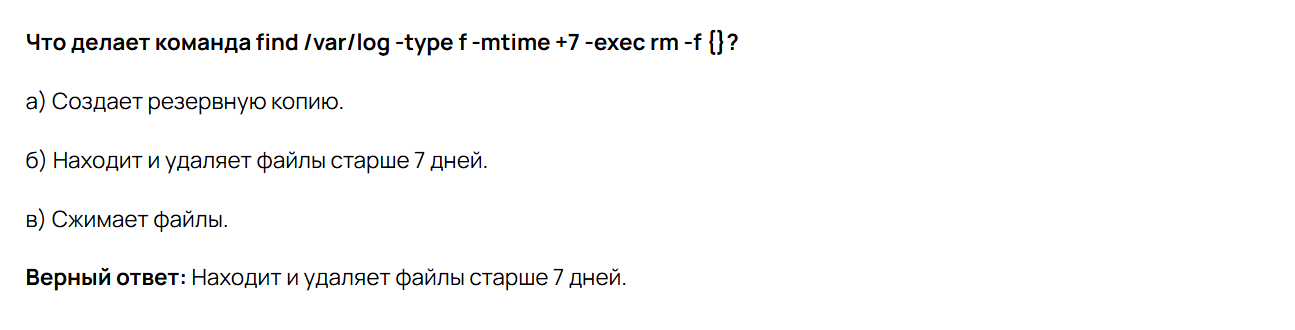


Рис. 74: «Основы автоматизации с помощью Bash-скриптов». Вопрос №3

Выбранный ответ: **Находит и удаляет файлы старше 7 дней**.

Команда find ищет в каталоге /var/log файлы (-f), которые имеют срок более 7 дней (-mtime +7). Для каждого из этих файлов выполняется (-exec) команда rm -f {}, где вместо фигурных скобок подставляется путь к найденному файлу.

# 4 Вывод

В результате прохождения первой части внешнего курса «Системный администратор Linux с нуля» я разобрал, чем занимается системный администратор, узнал о базовых командах, необходимых для управления системой, и выполнил практические занятия по управлению файлами и каталогами.