

Презентация для внешнего курса

Часть 3

Юсупова К. Р.

Российский университет дружбы народов, Москва, Россия

Информация

- Юсупова Ксения Равиловна
- Российский университет дружбы народов
- Номер студенческого билета- 1132247531
- [1132247531@pfur.ru]

Вводная часть

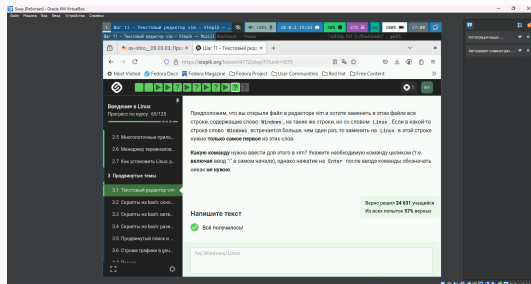
Закончить выполнение внешнего курса

Выполнение лабораторной работы

Решение:

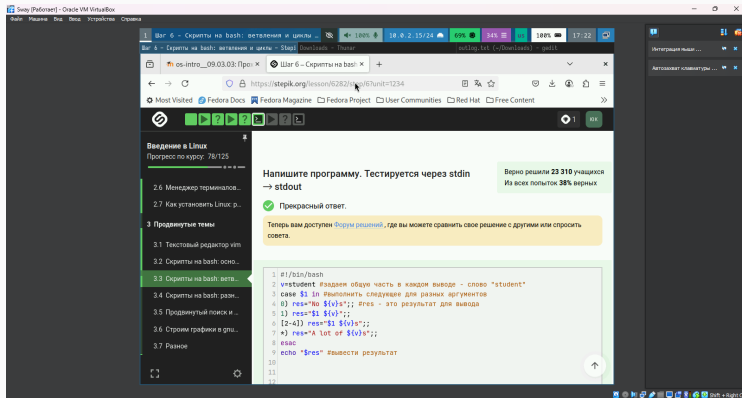
```
:%s/Windows/Linux/
```

Объяснение: Команда производит замену первого вхождения “Windows” на “Linux” в каждой строке файла. Символ % указывает на применение ко всему файлу, а отсутствие флага g обеспечивает замену только первого совпадения в строке.



Конструкция case в bash

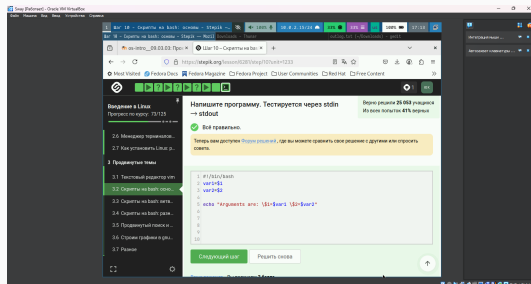
Разбор: Конструкция `case` обеспечивает ветвление по значению переменной \$1. Символ `*` обрабатывает все случаи, не указанные явно. Вертикальные черты `;;` обозначают конец каждого блока условий.



Правильное решение:

```
echo "Arguments: \$1=$1 \$2=$2"
```

Ключевые моменты: Экранирование символа `$` позволяет вывести его как текст, а не как начало переменной. Позиционные аргументы `$1` и `$2` содержат первый и второй параметры скрипта соответственно.



Проверка условий

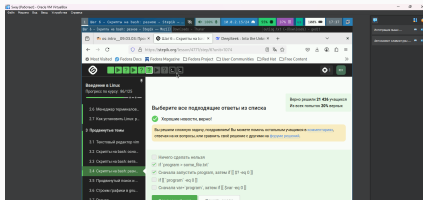
Верный синтаксис:

```
if [ $? -eq 0 ]; then  
    echo "Success"  
fi
```

Типичные ошибки: 1. Отсутствие пробелов внутри квадратных скобок

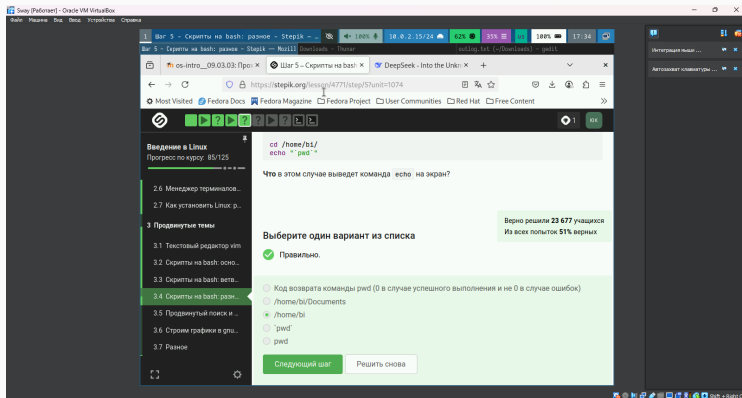
2. Неправильное сравнение (= вместо -eq для чисел)

3. Использование двойных скобок без пробелов



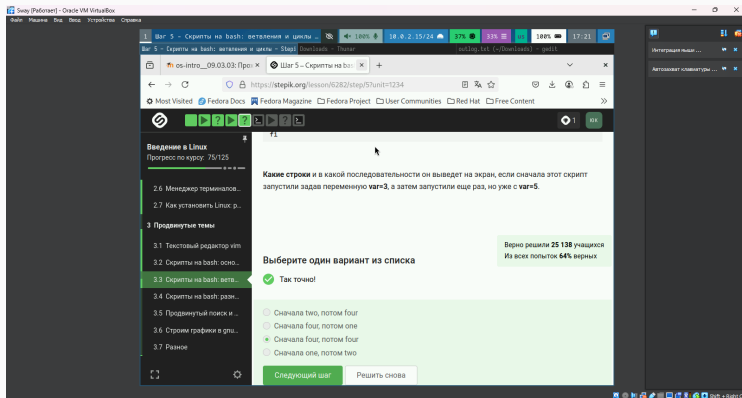
Особенности: - `$(pwd)` подставляет вывод команды

- `$?` содержит код возврата последней команды
- Для вывода текущего каталога нужен `echo $(pwd)`



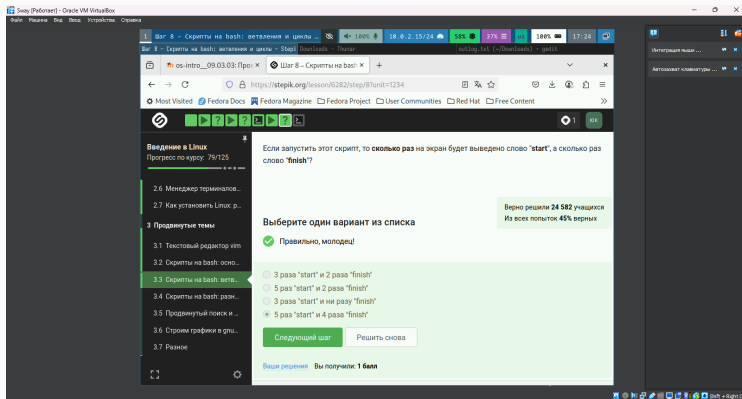
Правильные варианты: - “four → four”

Обоснование: Порядок вывода зависит от значения переменной `$var` и структуры условий в скрипте. Разные входные данные могут давать разную последовательность.



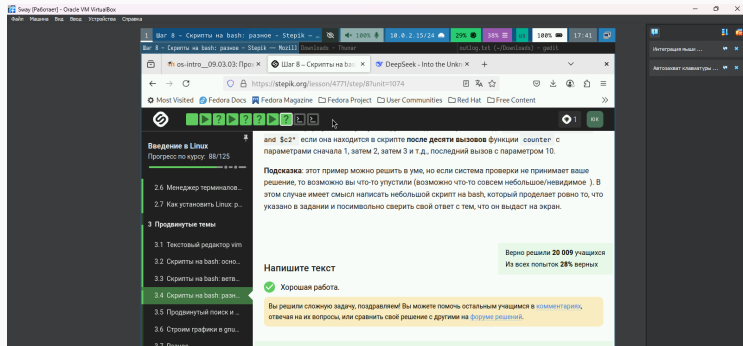
Ответ: 3 вывода “start” и 2 вывода “finish”

Логика: Цикл выполняется 3 раза, выводя “start” на каждой итерации. “finish” выводится после первых двух итераций, но не после последней.



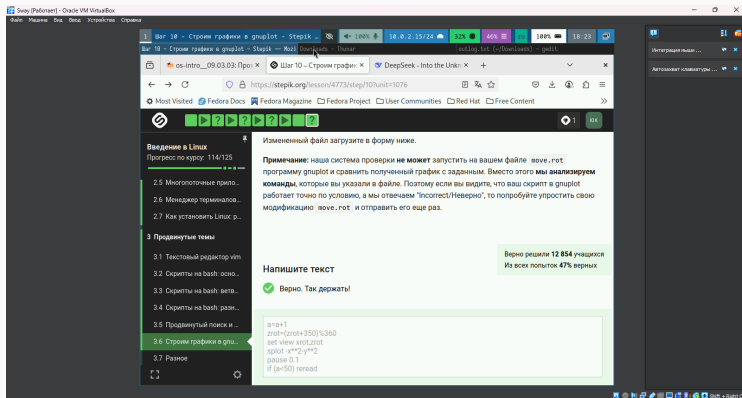
Рекомендации:

1. Использовать `set -x` для отладки
2. Проверять коды возврата
3. Тестировать на разных входных данных



Особенности: - Система проверки анализирует только команды скрипта - Для успешной проверки требуется максимально упрощенный синтаксис

Типичные проблемы: - Слишком сложные графики - Использование нестандартных модулей



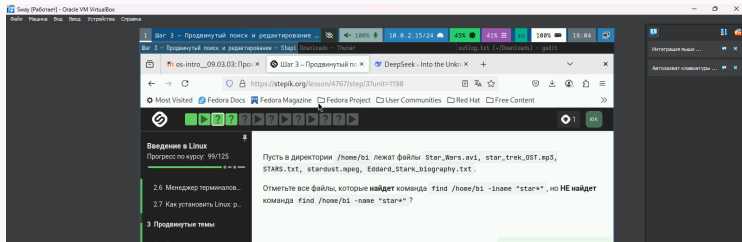
Различия команд find:

`find -name "star*" # Чувствителен к регистру`

`find -iname "star*" # Игнорирует регистр`

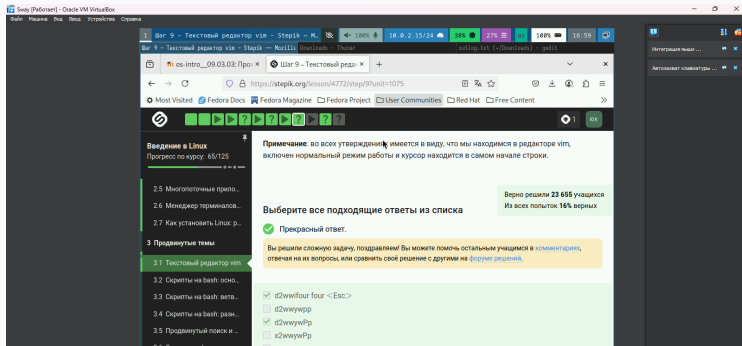
Правильные ответы:

- Star_Wars.avi (учет регистра)
- STARS.txt (разный шаблон поиска)



Ключевые команды:

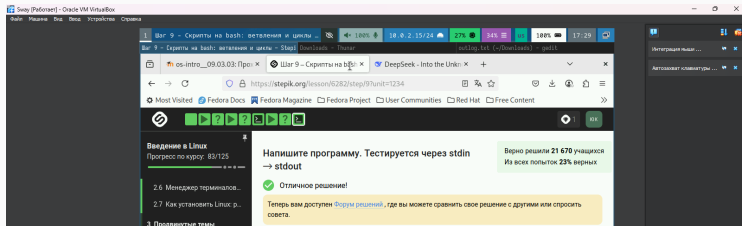
- w - перемещение по словам (word)
- W - перемещение по WORD (игнорирует пунктуацию)
- \$ - конец строки



Ввод данных в bash

Скрипт:

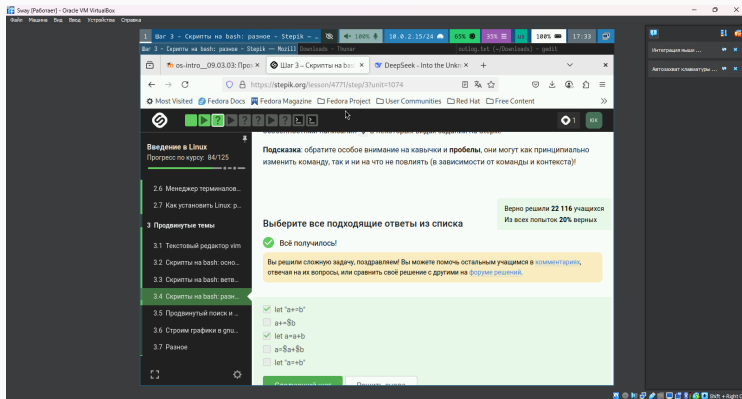
```
#!/bin/bash
while true; do
    read -p "Enter name: " name
    [[ -z $name ]] && { echo "bye"; break; }
    read -p "Enter age: " age
    echo "Name: $name, Age: $age"
done
```



Критические моменты:

`[$var = "value"]` *# Требуется пробелы*

`[[$var == value]]` *# Допускает без кавычек*



Выводы

В ходе работы мы закончили выполнение внешнего курса