МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем и технологий

**Отчет по лабораторной работе № 4**

по дисциплине: “Системное программирование”

на тему:**“Скрипты, диалоги, создание проекта”**

Вариант 5, 12

Выполнил**:** студент группы 10701222 Походня Д. А.

Ерченко К. Д.

Принял**:** ст. пр. Давыденко Н. В.

Минск 2024

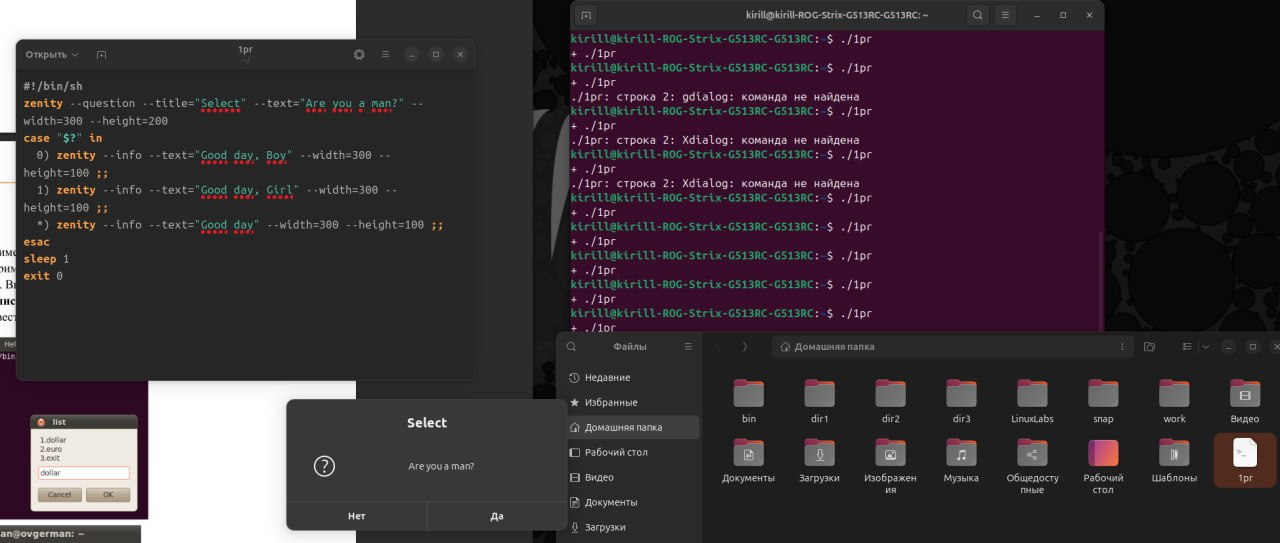
# Лабораторная работа № 4.

Цель работы: закрепить на практике принципы создания проектов с помощью скриптов SHELL, освоить средства примитивного графического интерфейса в Linux-скриптах.

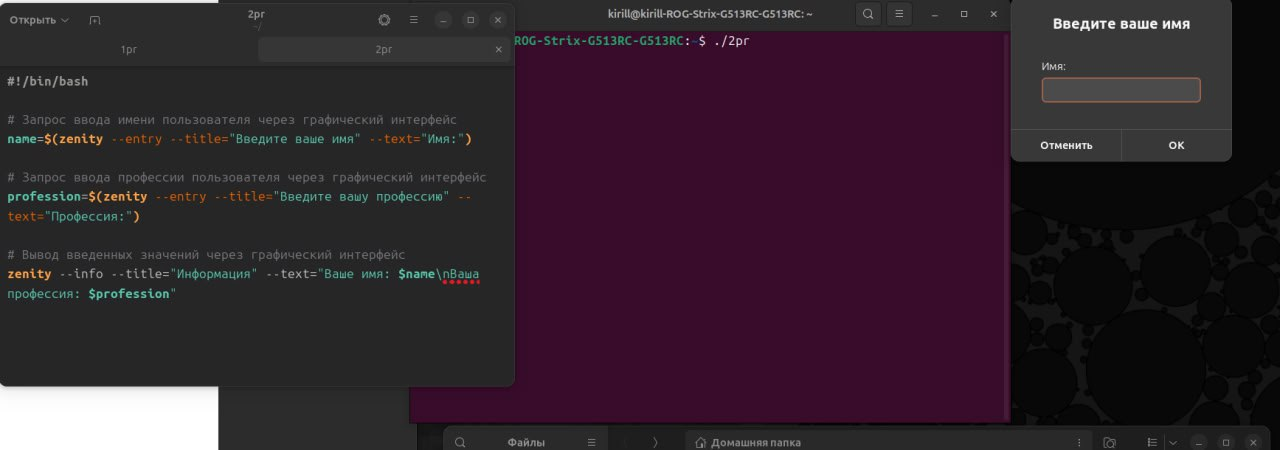
## Задание.

1. Сделать пример с двумя вложенными диалогами типа YesNo.

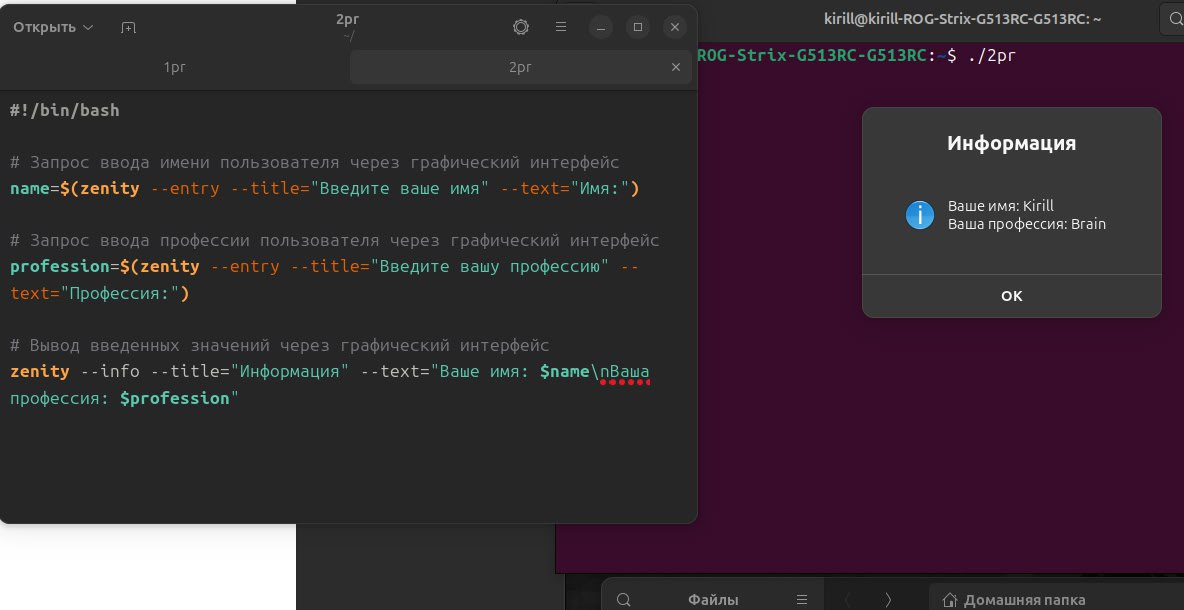
Решение



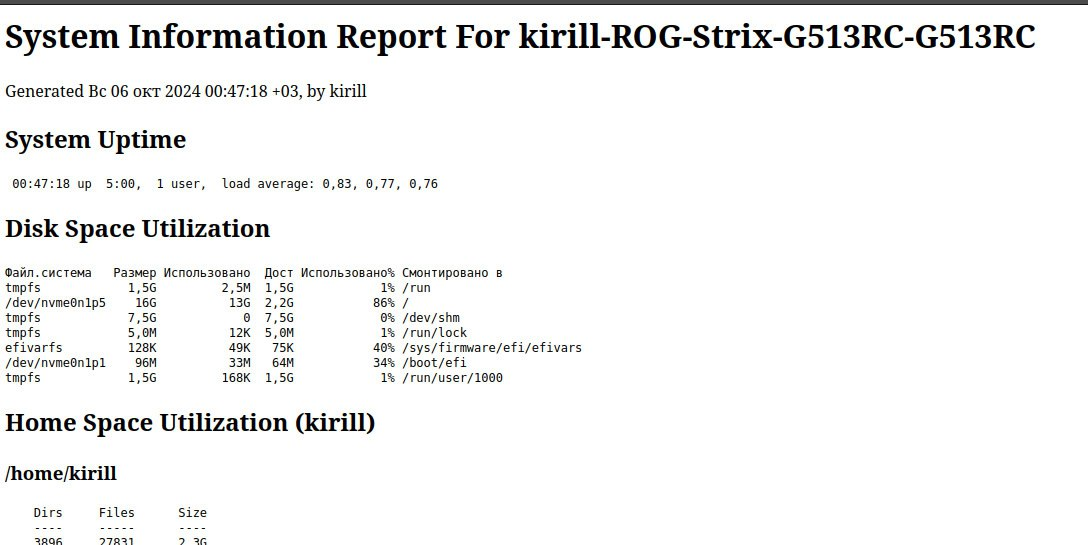
1. Сделать пример, запрашивающий сначала имя человека, а потом профессию. Вывести имя + профессию, прочитанные в диалоге.



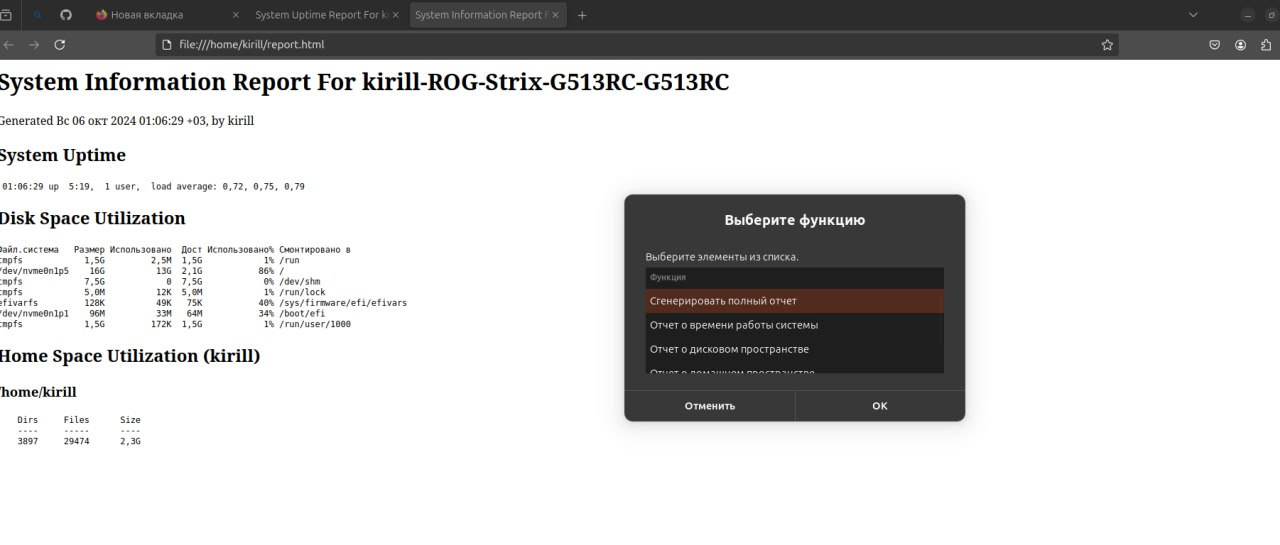
1. Вывести список с названиями валют. После выбора валюты система должна вывести ее котировку.



1. Создайте копию проекта представелнного в упражнении №3. Внесем изменеия в новом проекте, добавив вывод информации о домашнем каталоге каждого пользователя и включив в вывод общее число файлов и подкаталогов в каждом из них



1. Доведите проект до совершенства. Для сценария выполненного в задании №2 создайте графическое диалоговое окно, с интерактивным меню выбора функций. Информация должна автоматически передаваться в HTML-файл и открываться в браузере.



Контрольные вопросы

Вариант 12

Как командная оболочка определяет тип значения переменных?

**Решение**

В командной оболочке Linux, такой как Bash, тип значения переменных определяется по контексту их использования, а не явно. Вот основные моменты, которые стоит учитывать:

1. **Строки по умолчанию**: Все переменные в Bash по умолчанию воспринимаются как строки. Например, если вы присвоите переменной значение, она станет строкой.

bash

Copy

my\_var="Hello, World!"

1. **Арифметические операции**: Когда вы используете переменные в арифметических контекстах (например, в арифметических выражениях), Bash автоматически интерпретирует их как числа.

bash

Copy

num1=5

num2=10

sum=$((num1 + num2)) # Здесь num1 и num2 воспринимаются как числа

1. **Проверка типов**: Вы можете использовать команды, такие как [[ и ((, чтобы проверить, как Bash интерпретирует значение переменной.

bash

Copy

if [[ $my\_var =~ ^[0-9]+$ ]]; then

echo "Это число."

else

echo "Это строка."

fi

1. **Типы данных**: В Bash нет строгой типизации, и переменные могут менять свой тип в зависимости от того, как вы их используете. Например, вы можете сначала присвоить строку, а затем использовать ее в арифметическом выражении.
2. **Массивы и ассоциативные массивы**: Bash поддерживает массивы, которые могут содержать несколько значений. Однако, массивы также не имеют строгих типов.

bash

Copy

my\_array=(1 "two" 3.0)

1. **Хранение значений**: Значения переменных хранятся в виде строк, и при необходимости Bash автоматически преобразует их в числа или другие форматы, основываясь на контексте.

Таким образом, тип значения переменной в Bash определяется неявно и зависит от того, как вы планируете использовать эту переменную в вашем скрипте.

2. Что такое пользовательские переменные и для чего они используются?

Пользовательские переменные — это переменные, которые создаются пользователем в программировании или в контексте определённых приложений и систем. Они позволяют хранить данные, которые могут изменяться в процессе выполнения программы или работы приложения.

**Основные аспекты пользовательских переменных:**

1. **Создание и использование**:
   * Пользовательские переменные создаются с помощью присвоения значений, например, x = 10 в языках программирования.
   * Они могут хранить разные типы данных: числа, строки, массивы и т. д.
2. **Контекст использования**:
   * В веб-разработке пользовательские переменные могут использоваться для хранения данных, передаваемых между страницами или для настройки поведения приложения.
   * В базах данных они могут использоваться для хранения временных значений в ходе сессии.
3. **Применение**:
   * Пользовательские переменные часто применяются для динамической генерации контента, обработки пользовательского ввода и управления логикой программы.
   * Они помогают организовать код, делая его более понятным и гибким.

Таким образом, пользовательские переменные играют важную роль в разработке программного обеспечения, позволяя разработчикам управлять данными и логикой приложения.