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: ***”*Cкрипт-проект-диалоги*”***

Выполнил**:** студенты группы *10702121* Яковец И.А

Принял**:** пр. Давыденко Н. В.

Минск 2023

# Лабораторная работа №4.

**Цель работы:** Закрепить на практике принципы создания проектов с помощью скриптов Shell, освоить средства примитивного графического интерфейса в сприктах Shell.

## Задание 1

1. Cделать пример с двумя вложенными диалогами типа YesNo.

2. Cделать пример, запрашивающий сначала имя человека, а потом профессию. Вывести имя+профессию, прочитанную в диалоге.

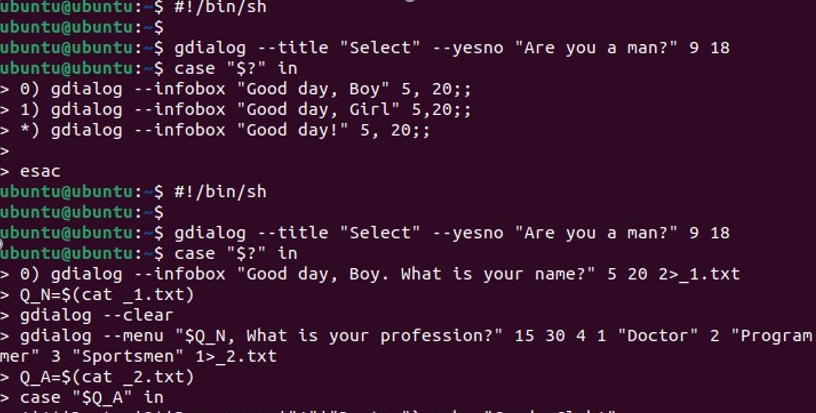
3. Вывести список с названиями валют. После выбора валюты система должна вывести ее котировку.

4. Сделать оконное меню валют к 3 пункту.

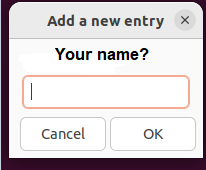
5. Добавить выход из программы.

**Решение**

1.Пишем скрипт с двумя вложенными диалогами типа YesNo сразу вместе с элементами для пункта 2.



2.Запрашиваем на ввод имя человека



3.Запрашиваем у пользователя информацию о его професии.



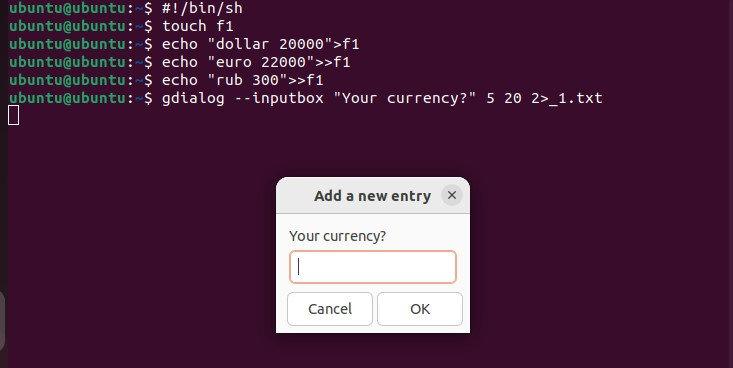
4.Реализовываем систему ответа на выбранный вариант



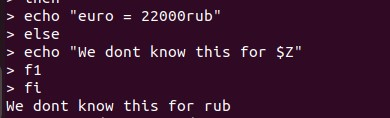
**2.**

1. Прописываем скрипт и спрашиваем у пользователя валюту, с которой он хочет работать.

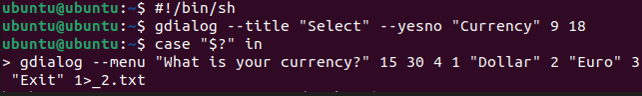


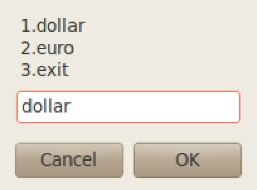


1. При вводе euro получаем котировку, при вводе незаданной валюты, получаем реализованное сообщение о том, что введене не так валюты

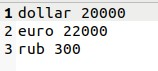


**3.**

1. Делаем оконное меню для валют и сразу реализовываем опцию выхода 



1. Создаем отдельное окно для выхода и добавочное окно катировки



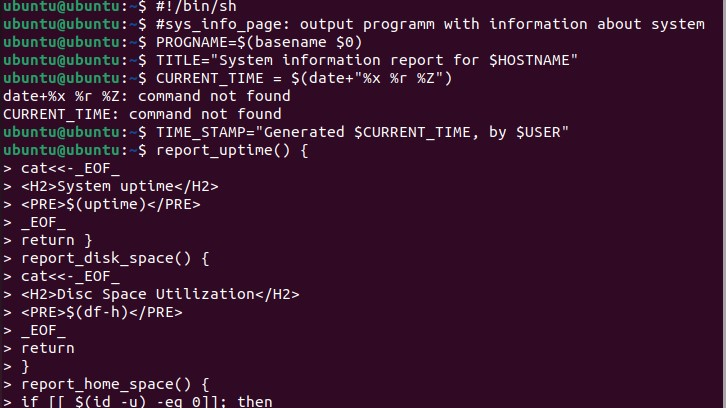


**Задание 2.**

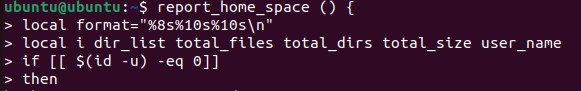
6. Создайте копию проекта из упражнения 3.

7. Добавьте туда файловый вывод

**4.** Создаем копию проекта



**5.** Добавляем туда файловый вывод



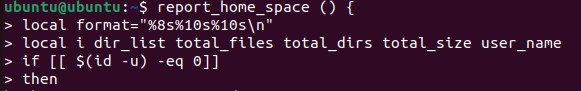
**Задание 3.**

8. Довести проект до совершенства, добавив выбор функций

9. Сделать так, чтобы информация хранилась в html файле и открывалась в браузере

**8.**

1. Добавляем домашнюю страницу для выбора функций



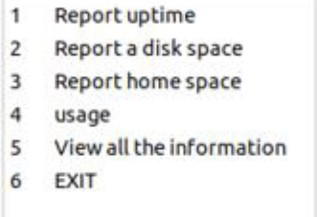
1. Добавляем выбор базовых функций



1. Реализовываем интерфейс для вывод в html документ



1. Получаем итоговое меню для выбора функций



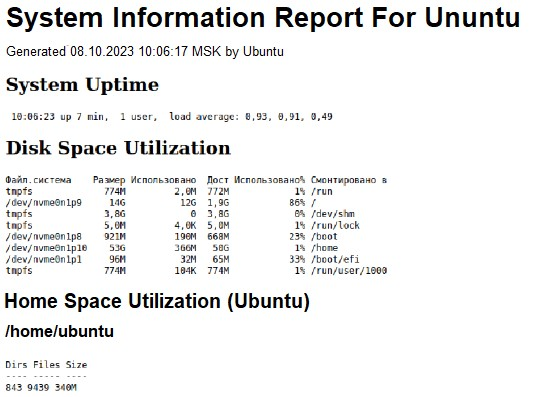
1. Подтверждаем запись в html файл



1. Открываем файл в браузере по нашему пути



1. Наблюдаем за итоговым результатом в браузере



Контрольные вопросы:

1. Какой командой и каким средством в скриптах обеспечивается интерактивность программы?

В Linux для обеспечения интерактивности программы часто используются стандартные утилиты командной строки, такие как read, а также команды для работы с терминалом. Вот несколько способов обеспечения интерактивности в скриптах на Linux:

read:

Команда read позволяет читать ввод с клавиатуры в скрипте. Вы можете использовать ее, чтобы запросить пользователя ввести данные или ответить на вопросы.

Вы также можете использовать флаг -p с командой read для вывода приглашения к вводу непосредственно в командной строке.

Команда select используется для создания меню выбора, где пользователь может выбирать опции.

Вы можете использовать циклы, такие как while или for, для обеспечения интерактивности в скриптах. Например, вы можете создать цикл while, который продолжает выполняться, пока не будет получен определенный сигнал от пользователя.

Использование утилит для работы с терминалом:

Вы также можете использовать специальные утилиты для работы с терминалом, такие как dialog или whiptail, чтобы создавать более сложные интерактивные интерфейсы в скриптах.

2. С помощью каких утилит реализуется средства примитивного

графического интерфейса в Linux-скриптах?

В Linux для создания примитивных графических интерфейсов в скриптах можно использовать следующие утилиты:

1. **Zenity**: Zenity - это утилита для создания диалоговых окон на основе GTK+. Она позволяет создавать окна для ввода текста, выбора файлов, сообщений об ошибке и другие диалоги. Zenity интегрируется хорошо с оконными менеджерами Linux.
2. **KDialog**: KDialog - это аналог Zenity, но ориентированный на среду рабочего стола KDE. Он предоставляет средства для создания диалоговых окон и интерактивных элементов управления в скриптах.
3. **Dialog (ncurses)**: Dialog - это утилита на основе ncurses, которая позволяет создавать текстовые диалоговые окна в терминале. Она предоставляет простые способы создания меню выбора, ввода данных и других элементов интерфейса в текстовом режиме.
4. **Whiptail**: Whiptail - еще одна утилита на основе ncurses для создания текстовых диалоговых окон. Она поддерживает различные виды диалогов, такие как формы, списки выбора и т. д.