МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

О Т Ч Е Т

ПО ЛАБОРАТОРНОЙ РАБОТЕ №**2.2**

«*Командный язык и скрипты Shell. Часть 2 – Скрипты, создание проекта, диалоги»*

по курсу: "Системное программирование"

Выполнил: Сулим И.С., 10702121

Проверил: Давыденко Н.Н.

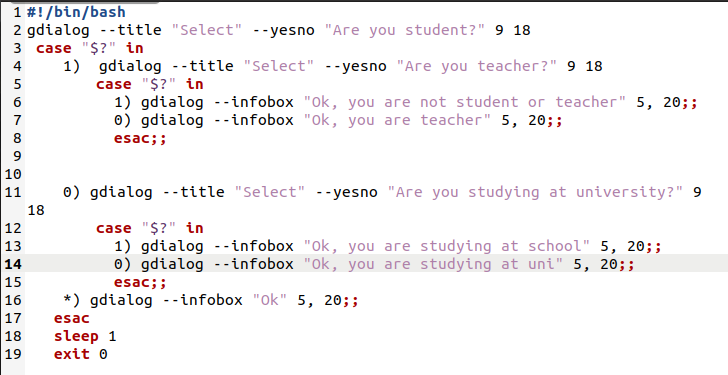
**Цель работы :**

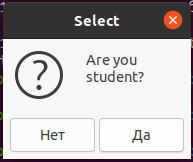
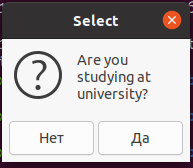
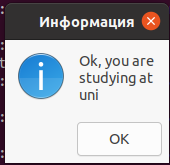
Закрепить на практике принципы создания проектов с помощью скриптов SHELL, освоить средства примитивного графического интерфейса в Linux-скриптах.

**Задание :**

**Задание 1**

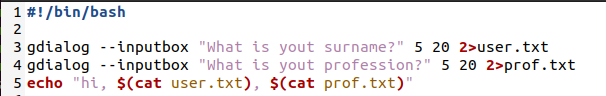
1. Сделать пример с двумя вложенными диалогами типа YesNo.

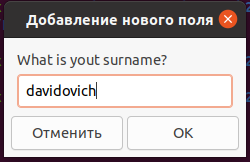
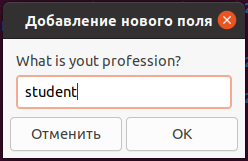


1. Сделать пример, запрашивающий сначала имя человека, а потом профессию. Вывести имя + профессию, прочитанные в диалоге.

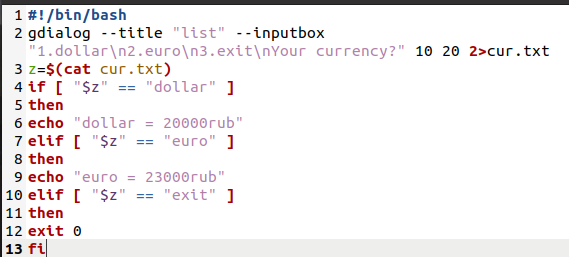


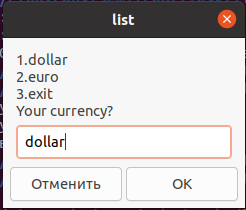


1. Вывести список с названиями валют. После выбора валюты система должна вывести ее котировку.

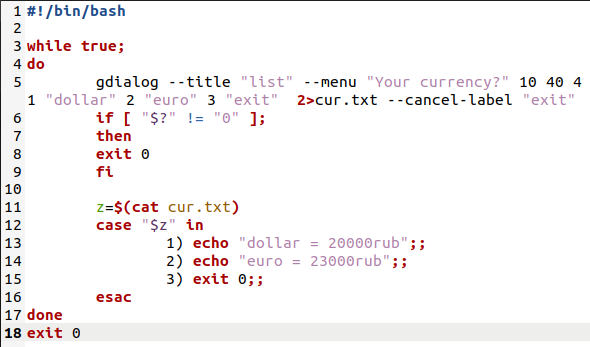


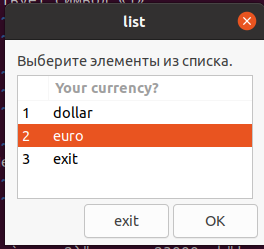




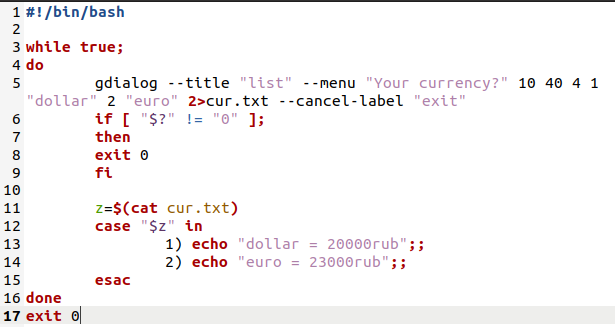
1. Измените предыдущую программу так, вместо списка валют предлагалось оконное меню валют. Чтобы программа работала в цикле. Для выхода из цикла нужно вместо названия валюты вводить exit.

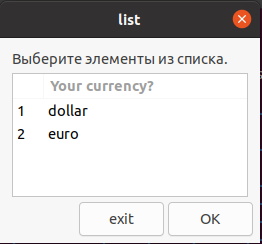






1. Измените предыдущую программу так, для выхода из программы в меню была кнопка закрытия программы «exit».

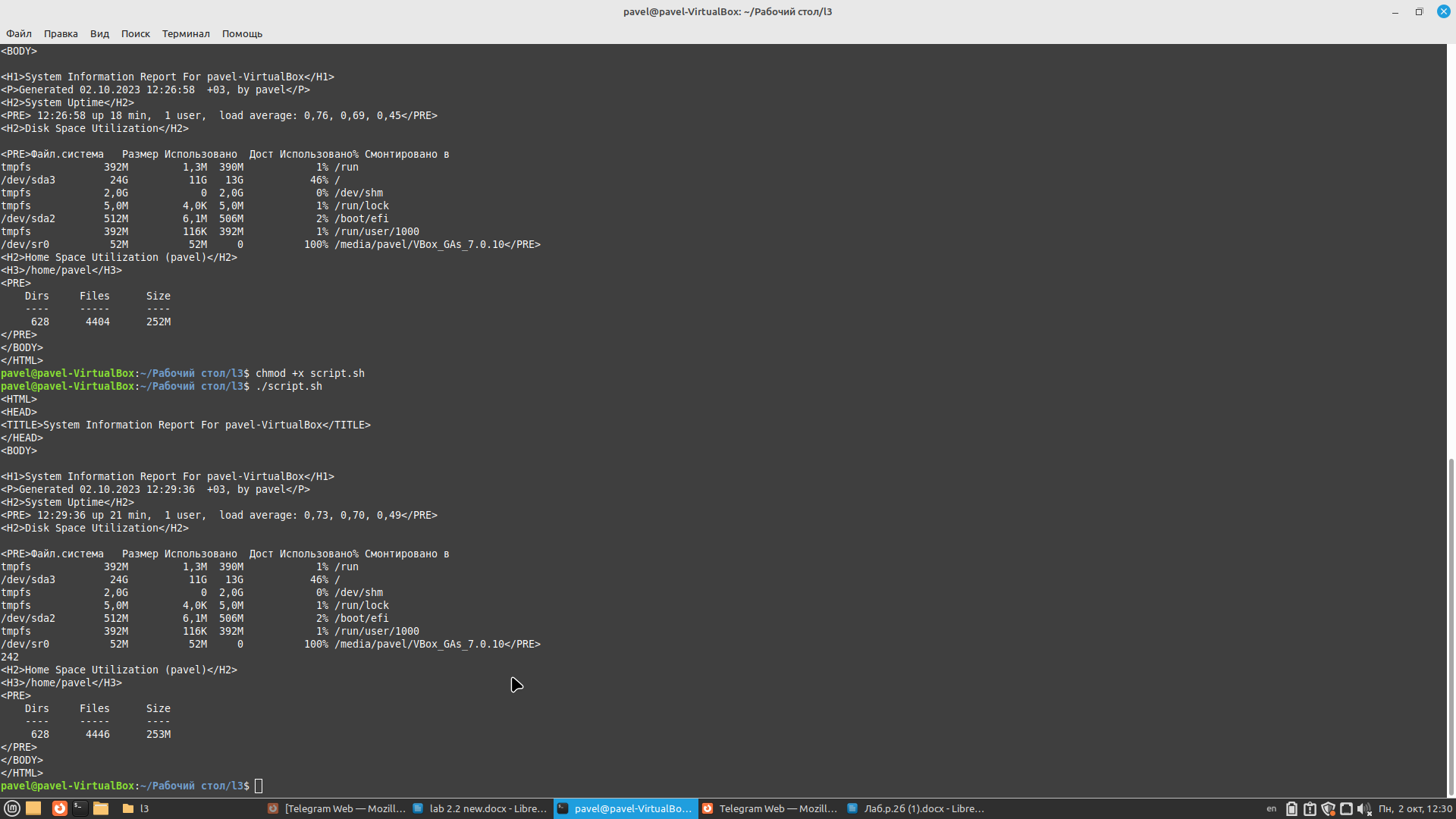




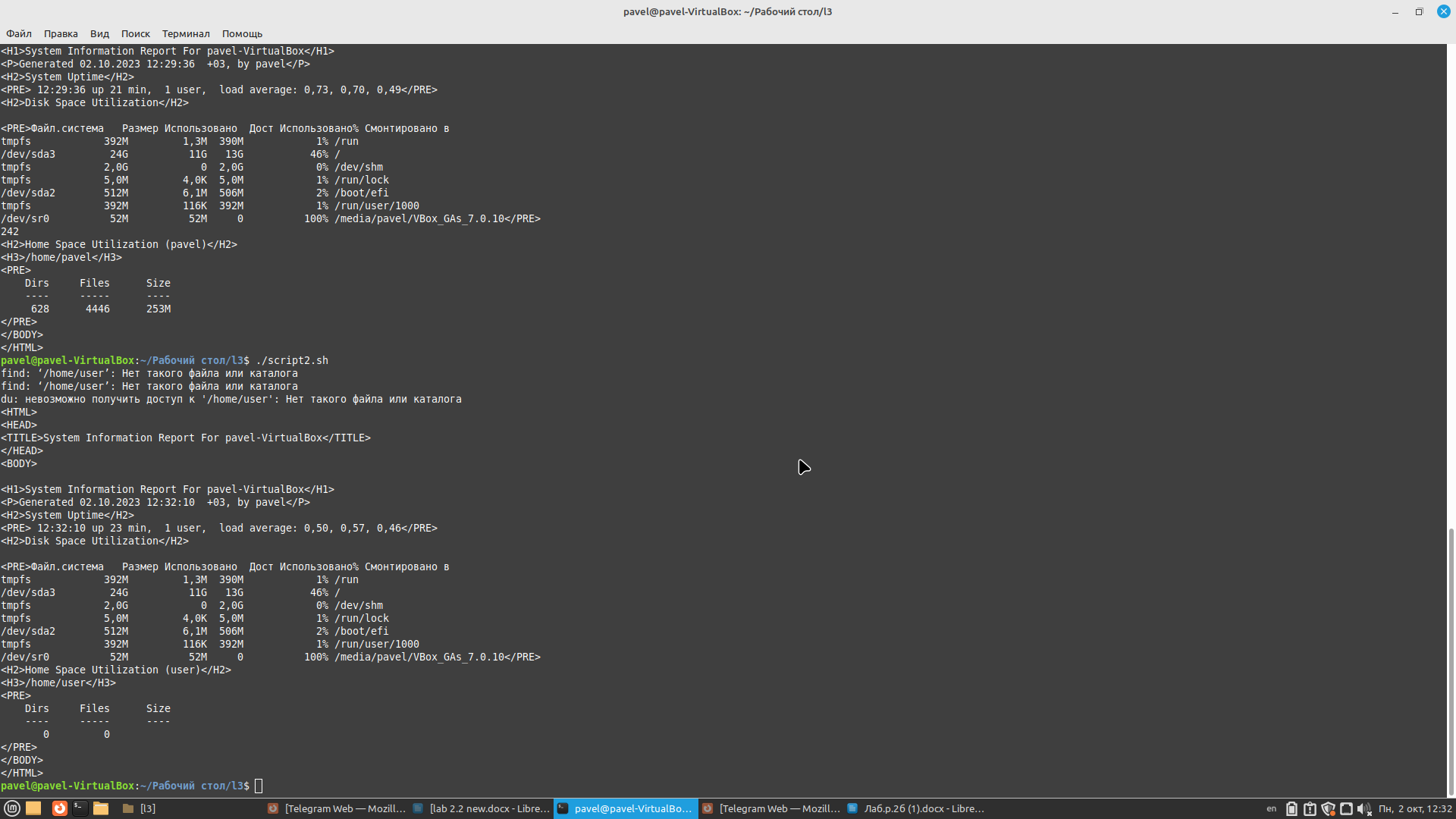
**Задание 2**

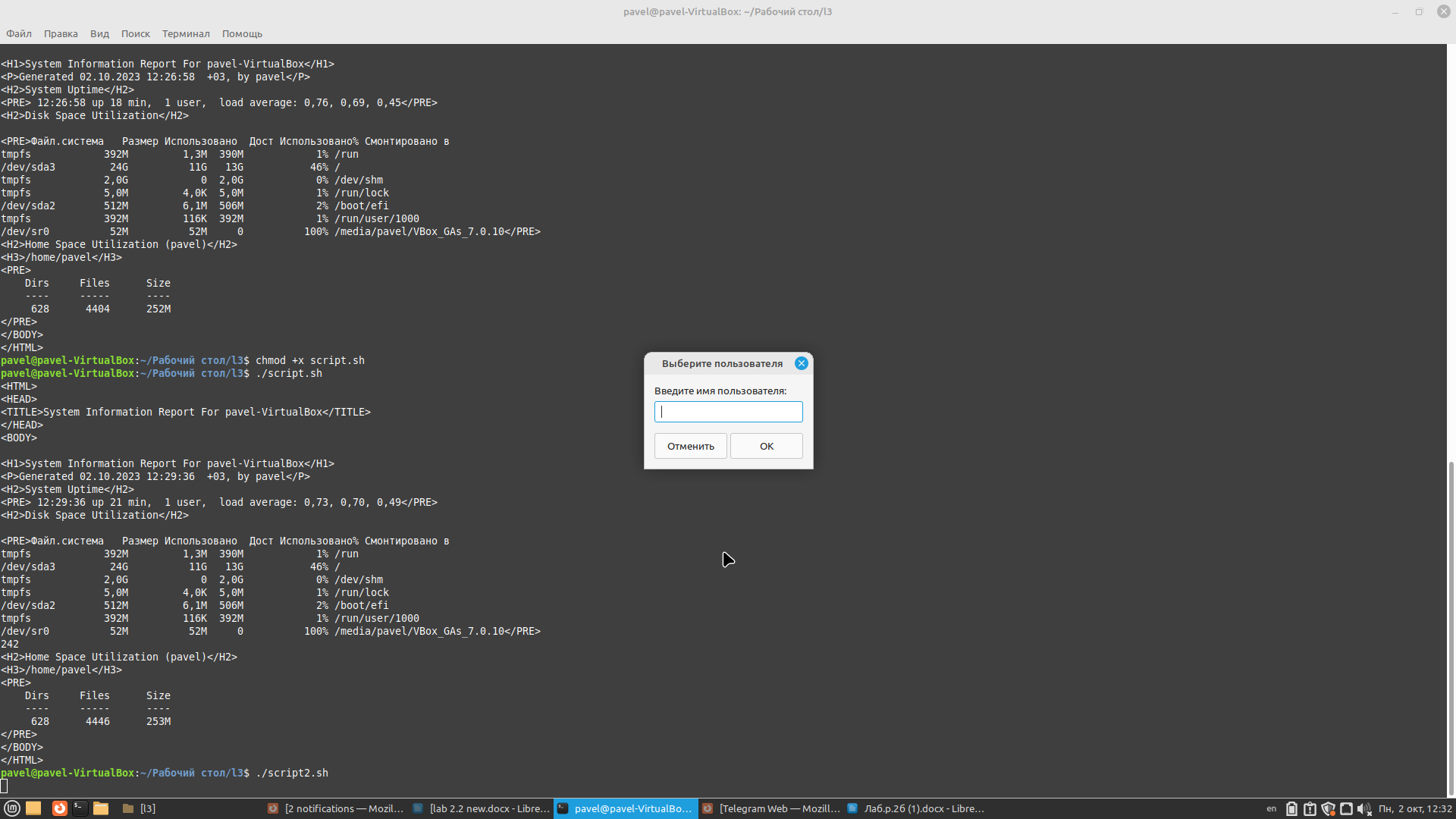
Создайте копию проекта представленного в упражнении №3. Внесем изменения в новом проекте, добавив вывод информации о домашнем каталоге каждого пользователя и включив в вывод общее число файлов и подкаталогов в каждом из них:

В этой новой версии проверяется наличие привилегий суперпользователя, но вместо того, чтобы выполнить полный набор операций в каждой из ветвей if, здесь устанавливаются некоторые переменные, которые затем используются в цикле for. В функции использованы несколько локальных переменных и команда printf для форматирования части вывода.



**Задание 3**



Для сценария выполненного в задании №2 создайте графическое диалоговое окно

**Вывод :** в ходе выполнения лабораторной работы я закрепила на практике принципы создания проектов с помощью скриптов SHELL, освоила средства примитивного графического интерфейса в Linux-скриптах.

**Контрольные вопросы:**

**Возможности командной оболочки и скриптов Shell**

1. **Что такое область видимости переменных? Какие типы области видимости переменных существуют?**

Область видимости – участок программы, в рамках которого можно использовать переменную. Существуют глобальные и локальные области видимости переменных.

Область видимости переменных определяет, где в вашей программе или скрипте можно получить доступ к определенной переменной и где переменная не видна. Область видимости устанавливает правила доступа к переменным и их значениям. В различных языках программирования и средах могут существовать разные типы областей видимости. Основные типы областей видимости включают:

Глобальная область видимости (Global Scope): Переменные, определенные в глобальной области видимости, видны и доступны во всей программе или скрипте. Это означает, что они могут быть использованы в любой части кода.

Локальная область видимости (Local Scope): Переменные, определенные внутри функции или блока кода, имеют локальную область видимости. Это означает, что они видны только внутри этой функции или блока кода и недоступны вне его. Это способствует изоляции переменных и предотвращает конфликты имен.

Область видимости вложенных функций (Nested Scope): Если функция вложена в другую функцию, она может иметь доступ к переменным внешней функции, но не наоборот. Это называется "цепью областей видимости".

Блочная область видимости (Block Scope): Некоторые языки программирования, такие как JavaScript, поддерживают блочную область видимости. Переменные, определенные внутри блока кода (например, внутри цикла или условного оператора), видны только в этом блоке.

**Графический интерфейс диалога**

1. **С помощью каких утилит реализуется средства примитивного графического интерфейса в Linux-скриптах?**

gdialog, zenity, Xdialog, whiptail.

Примитивные графические интерфейсы Linux-скриптов могут включать различные типы окон и виджетов для взаимодействия с пользователем. Вот некоторые из основных типов окон и виджетов, которые могут использоваться в Linux-скриптах:

Диалоговые окна (Dialog Boxes): Диалоговые окна используются для вывода информации и запроса данных от пользователя. Они могут включать текстовые сообщения, кнопки "ОК" и "Отмена", а также поля ввода.

Меню выбора (Selection Menus): Меню выбора позволяют пользователю выбирать опции из списка. Они могут быть выпадающими или в виде списка с радиокнопками или флажками для выбора одной или нескольких опций.

Формы ввода (Input Forms): Формы ввода предоставляют поля для ввода текста, чисел и других данных. Они могут включать метки для описания полей и кнопку "Отправить" для отправки данных.

Информационные окна (Information Windows): Информационные окна используются для вывода информации без необходимости пользовательского ввода. Они могут содержать текст, изображения и другие элементы.

Графические элементы (Graphical Elements): Примитивные графические интерфейсы могут включать в себя графические элементы, такие как кнопки, иконки, изображения и линии.

Окна выбора файла (File Selection Dialogs): Окна выбора файла позволяют пользователю выбирать файлы или директории на файловой системе. Они обеспечивают доступ к файлам и папкам для выбора или сохранения данных.

Окна предупреждений (Alert/Warning Dialogs): Окна предупреждений используются для вывода важных сообщений или предупреждений пользователю. Они могут содержать информацию о проблемах или действиях, требующих внимания пользователя.

Списки (Lists): Списки могут использоваться для отображения данных в виде списка. Они могут быть вертикальными или горизонтальными и могут поддерживать прокрутку.

Строки состояния (Status Bars): Строки состояния могут содержать информацию о текущем состоянии приложения или процесса.

Таблицы (Tables): Таблицы могут использоваться для отображения данных в табличной форме. Они могут содержать ячейки с данными и заголовки столбцов.