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 № 2.2**

по дисциплине:” Системное программирование”

на тему: “Командный язык и скрипты Shell”

Выполнил**:** студент группы 10701321

Мацкевич М.Ю.

Принял**:** Давыденко Н.В.

Минск 2023

# Лабораторная работа № 2.2. Командный язык и скрипты Shell

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

Задание 1

1. Сделать пример с двумя вложенными диалогами типа YesNo.

Решение

#!/bin/bash

zenity --question --text="Вы хотите открыть вложенный диалог?" --title="Первый диалог"

if [ $? = 0 ]; then

zenity --question --text="Вы выбрали 'Да'. Продолжить?" --title="Второй диалог"

if [ $? = 1 ]; then

zenity --info --text="Вы выбрали 'Нет'. Завершение..." --title="Завершение"

fi

else

zenity --info --text="Вы выбрали 'Нет'. Завершение..." --title="Завершение"

fi

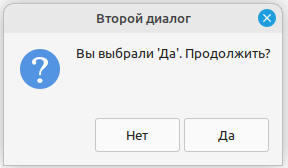
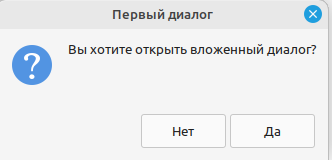


Рис.1 – Диалоговые окна

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

#!/bin/bash

name=$(zenity --entry --text="Введите ваше имя:" --title="Имя")

profession=$(zenity --entry --text="Введите вашу профессию:" --title="Профессия")

zenity --info --text="Ваше имя: $name\nВаша профессия: $profession"

exit 0

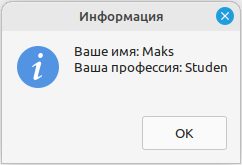
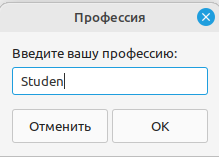
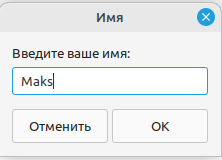


Рис.2 – Результат выполнения задания 2

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

#!/bin/bash

selected\_currency=$(zenity --entry --title="Выберите валюту" --text="Введите название валюты (dollar или euro):")

if [ "$selected\_currency" = "dollar" ]; then

exchange\_rate=3.36

elif [ "$selected\_currency" = "euro" ]; then

exchange\_rate=3.55

else

zenity --info --text="Неверное название валюты. Завершение..."

exit 1

fi

zenity --info --text="Курс $selected\_currency: $exchange\_rate Byn"

exit 0

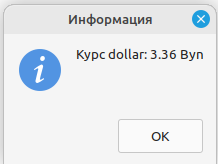
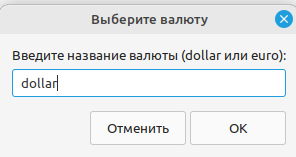


Рис.3 – Котировка валют

4. 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.  
#!/bin/bash

while true; do

selected\_currency=$(zenity --entry --title="Выберите валюту или введите 'exit'" --text="Введите название валюты (dollar или euro):")

if [ "$selected\_currency" = "exit" ]; then

break

fi

if [ "$selected\_currency" = "dollar" ]; then

exchange\_rate=3.36

elif [ "$selected\_currency" = "euro" ]; then

exchange\_rate=3.55

else

zenity --info --text="Неверное название валюты. Попробуйте снова или введите 'exit' для выхода."

continue

fi

zenity --info --text="Курс $selected\_currency: $exchange\_rate BYN"

done

exit

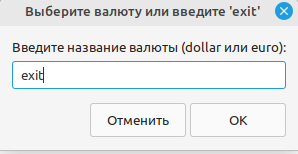


Рис.4 – Изменённое окно

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

#!/bin/bash

while true; do

selected\_currency=$(zenity --list --title="Выберите валюту" --text="Выберите валюту из списка или 'exit' для выхода:" --column="Валюта" dollar euro exit)

if [ "$selected\_currency" = "exit" ]; then

break

fi

if [ "$selected\_currency" = "dollar" ]; then

exchange\_rate=3.36

elif [ "$selected\_currency" = "euro" ]; then

exchange\_rate=3.55

else

zenity --info --text="Неверное название валюты. Попробуйте снова или выберите 'exit' для выхода."

continue

fi

zenity --info --text="Курс $selected\_currency: $exchange\_rate"

done

exit 0

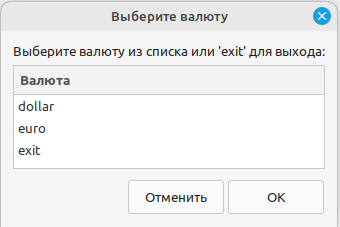


Рис.5 – Изменение типа ввода

Задание 2  
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.

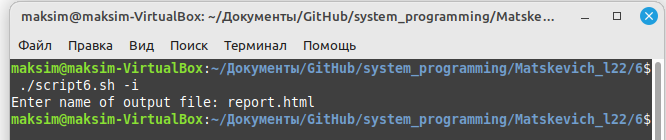


Рис.6 – Выполнение программы

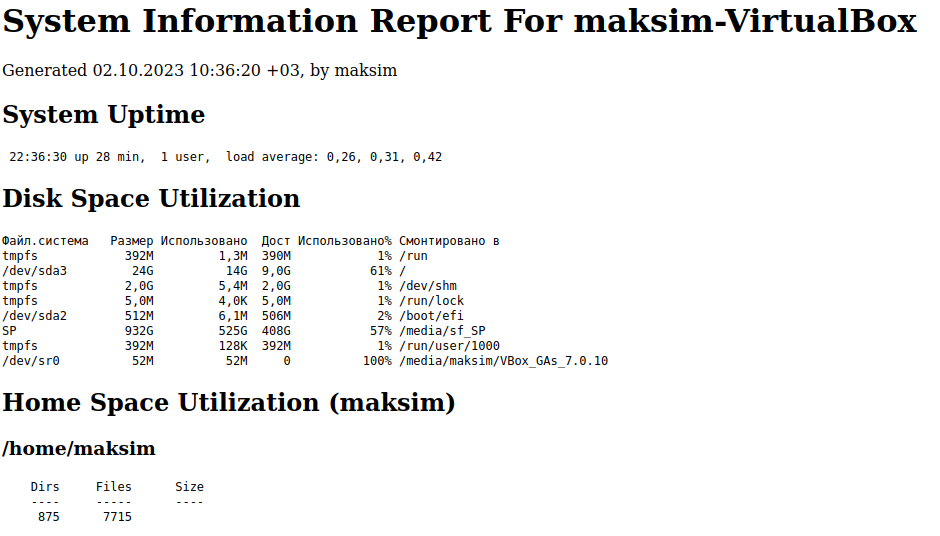
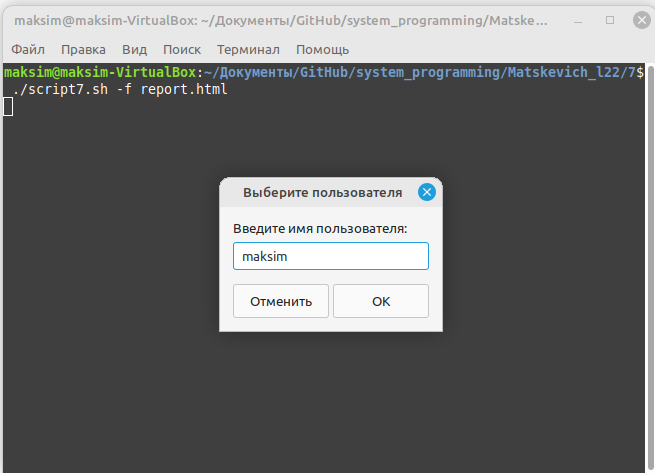
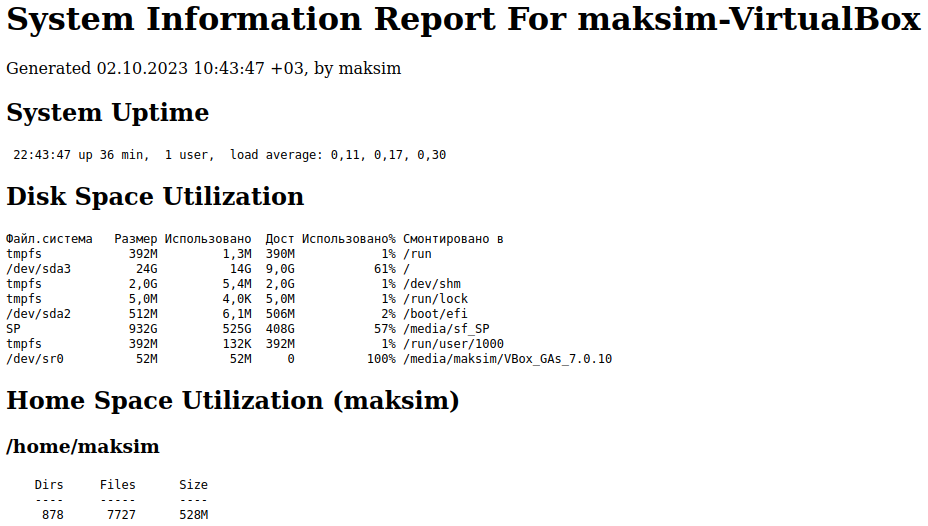


Рис.7 – Созданный отчёт

Задание 3 Для сценария выполненного в задании №2 создайте графическое диалоговое окно.

  
  
Рис.8 – Выполнение программы

  
  
  
Рис.9 – Созданный отчёт

Вывод*:* В ходе выполнения лабораторной работы изучили основные команды,   
для написания скриптов.  
  
Контрольные вопросы:  
 1. Что такое область видимости переменных? Какие типы области

видимости переменных существуют?

Область видимости переменных - это контекст, в пределах которого переменная может быть использована или изменена. В разных языках программирования и средах выполнения существуют разные типы областей видимости переменных. Давайте рассмотрим некоторые из них:

Локальная область видимости (Local Scope):

* Переменные с локальной областью видимости доступны только внутри определенной функции или блока кода, где они были объявлены..

Глобальная область видимости (Global Scope):

* Переменные с глобальной областью видимости доступны во всем коде программы, после их объявления.

Область видимости модуля (Module Scope):

* В некоторых языках, таких как Python, каждый файл считается модулем, и переменные, объявленные на верхнем уровне файла, имеют область видимости модуля.

Область видимости блока (Block Scope):

* Некоторые языки, такие как JavaScript (с использованием ключевого слова let или const), имеют блочную область видимости, где переменные видны только внутри блока кода, ограниченного фигурными скобками {}.

Область видимости Shell-скриптов:

В скриптах Shell, таких как bash, область видимости переменных зависит от того, где переменная объявлена:

* Переменные, объявленные внутри функции (с помощью function\_name() { ... }), имеют локальную область видимости.
* Переменные, объявленные за пределами функций, имеют глобальную область видимости и могут быть использованы во всем скрипте.
* Использование ключевого слова local внутри функции делает переменную локальной для этой функции.

2. Какие существуют типы окон примитивного графического интерфейса Linux-скриптов?  
  
В Linux-скриптах для создания примитивных графических интерфейсов можно использовать различные утилиты и библиотеки. Вот некоторые из них:

Dialog: Dialog - это утилита командной строки, которая позволяет создавать диалоговые окна и формы в терминале. Она предоставляет различные виджеты, такие как кнопки, списки, текстовые поля и т. д.

Zenity: Zenity - это утилита, которая позволяет создавать графические диалоговые окна внутри терминала. Она предоставляет простой способ взаимодействия с пользователем через окна с кнопками, полями ввода и другими элементами управления.

Whiptail: Whiptail - это еще одна утилита для создания текстовых диалоговых окон в терминале. Она предоставляет множество опций для настройки диалоговых окон и элементов интерфейса.

ncurses: Библиотека ncurses позволяет создавать текстовые пользовательские интерфейсы в терминале. Она предоставляет более широкие возможности для создания интерфейсов, чем утилиты типа Dialog, Zenity или Whiptail, но требует написания кода на C или других поддерживаемых языках.