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 № 04**

по дисциплине: “Системное программирование”

на тему: ***“Командный язык и скрипты SHELL”***

Вариант 12

Выполнил:студент группы *10701321* Трухов С.А.

Принял**:** пр. Давыденко Н.В.

Минск 2023

# Лабораторная работа № 04. 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.

**Задание 1**

1. Сделать пример с двумя вложенными диалогами типа YesNo.

2. Сделать пример, запрашивающий сначала имя человека, а потом

профессию. Вывести имя + профессию, прочитанные в диалоге.

3. Вывести список с названиями валют. После выбора валюты система

должна вывести ее котировку.

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.

5. Измените предыдущую программу так, для выхода из программы в

меню была кнопка закрытия программы «exit».

**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:

report\_home\_space () {

local format="%8s%10s%10s\n"

local i dir\_list total\_files total\_dirs total\_size user\_name

if [[ $(id -u) -eq 0 ]]; then

dir\_list=/home/\*

user\_name="All Users"

else

dir\_list=$HOME

user\_name=$USER

fi

echo "<H2>Home Space Utilization ($user\_name)</H2>"

for i in $dir\_list; do

total\_files=$(find $i -type f | wc -l)

total\_dirs=$(find $i -type d | wc -l)

total\_size=$(du -sh $i | cut -f 1)

echo "<H3>$i</H3>"

echo "<PRE>"

printf "$format" "Dirs" "Files" "Size"

printf "$format" "----" "-----" "----"

printf "$format" $total\_dirs $total\_files $total\_size

echo "</PRE>"

done

return

}

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.

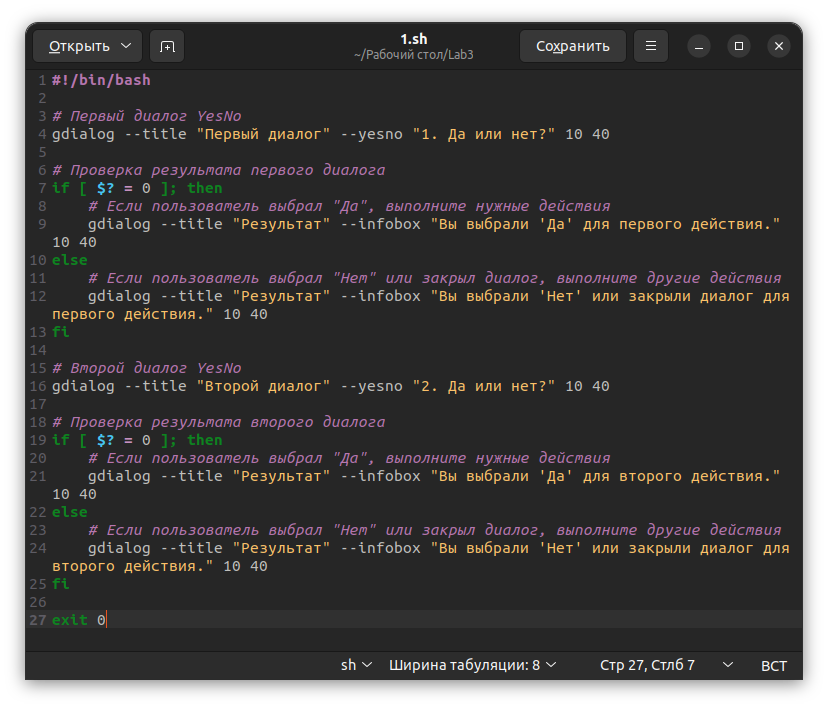
**Задание 3**

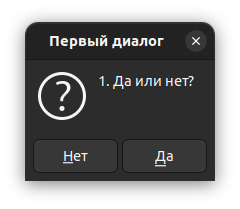
Для сценария выполненного в задании №2 создайте графическое диалоговое

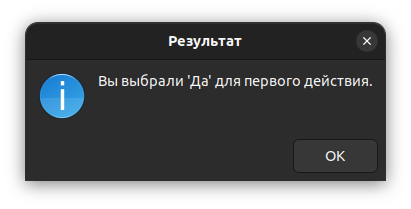
окно.

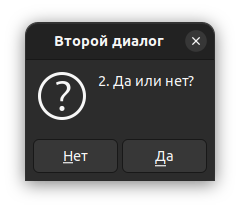
### Решение:

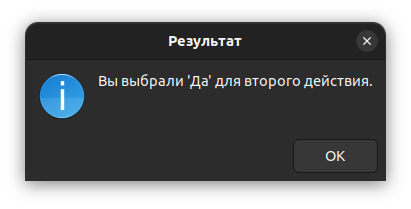
1)1.



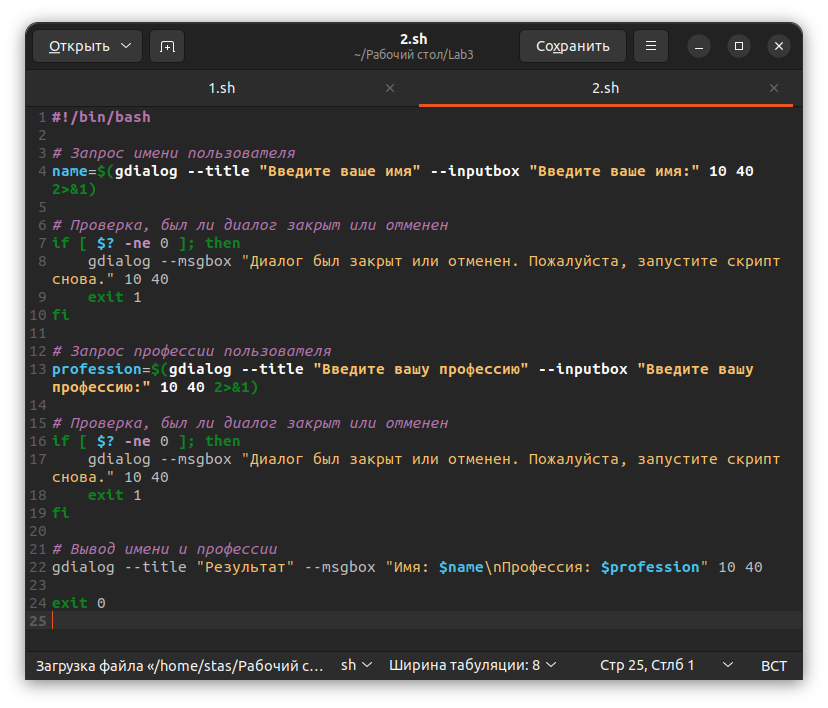


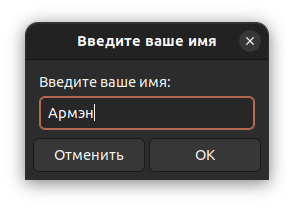


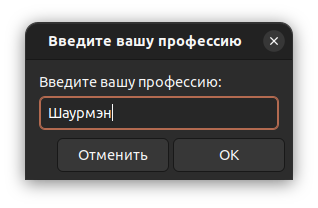


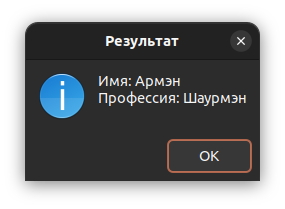


2.

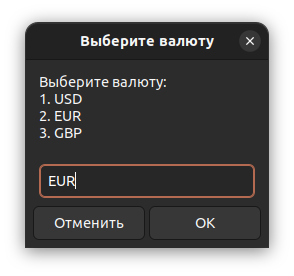


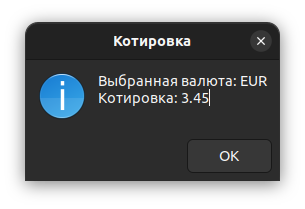




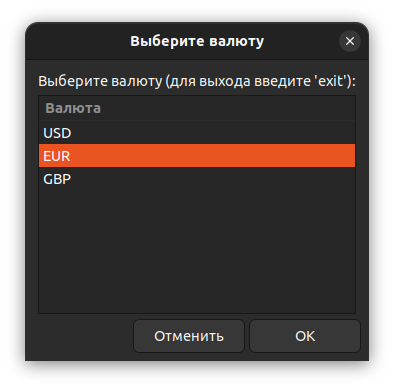


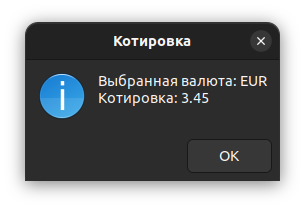
3.



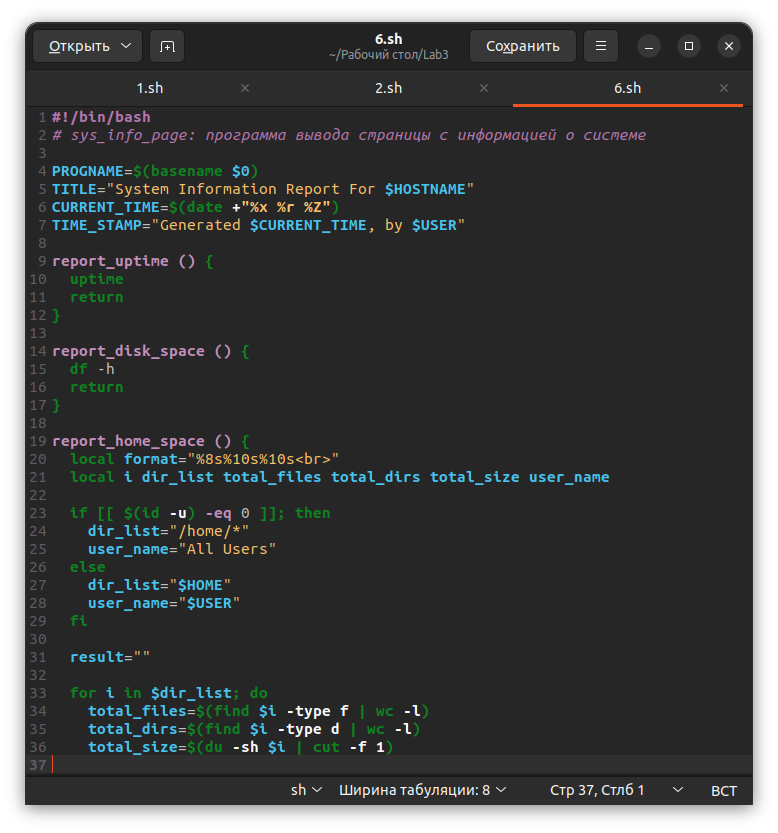


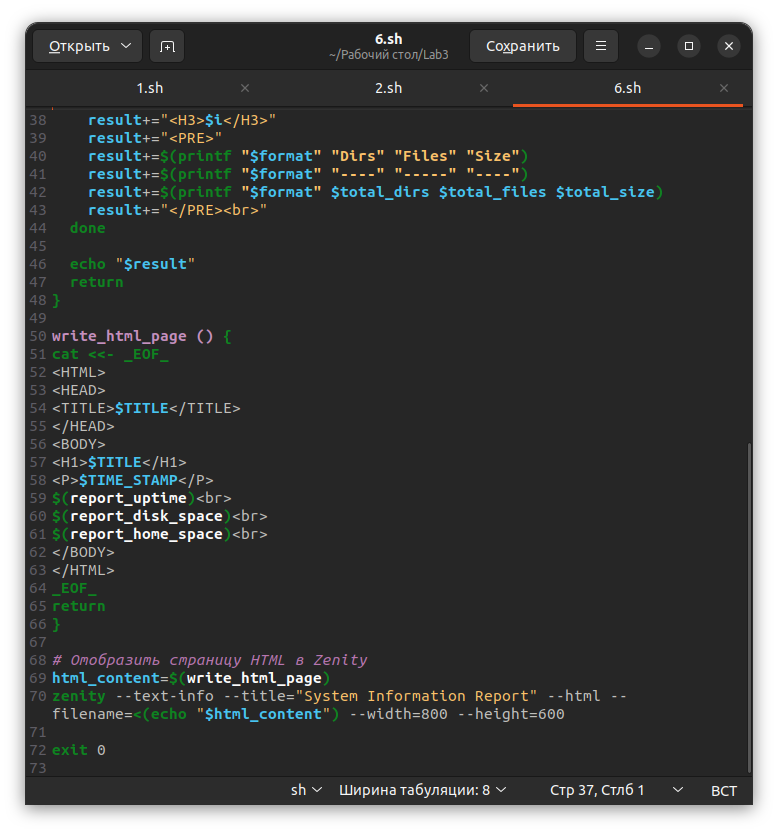
4-5.

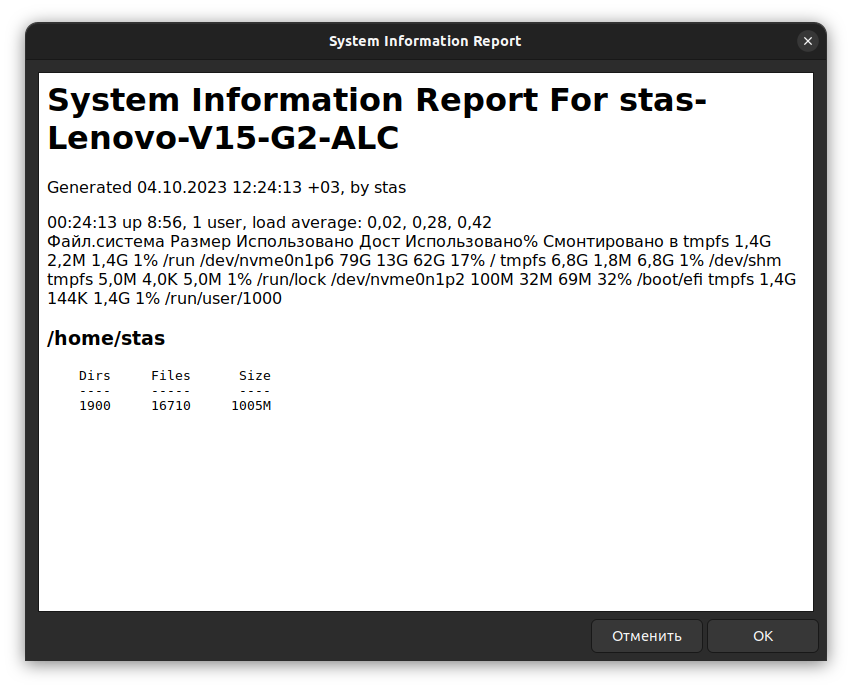




2-3)







4) 1. Как командная оболочка определяет тип значения переменных?

В отличие от некоторых других языков программирования, командная оболочка не заботится о типах значений, присваиваемых переменным, она все значения интерпретирует как строки.

2. Какие существуют типы окон примитивного графического интерфейса Linux-скриптов?

В Linux существует несколько типов окон примитивного графического интерфейса (GUI), которые можно использовать в скриптах. Некоторые из наиболее популярных типов включают:

1. X11:
   * X11 (X Window System) является основным оконным системным протоколом в большинстве дистрибутивов Linux. Вы можете использовать X11-совместимые библиотеки и инструменты для создания GUI-приложений и диалоговых окон в скриптах.
2. GTK:
   * GTK (GIMP Toolkit) представляет собой набор библиотек и инструментов для создания графических интерфейсов. Он широко используется в Linux-приложениях, и вы можете создавать оконные интерфейсы с его помощью в скриптах, используя языки программирования, такие как Python (с библиотекой PyGTK) или Perl (с библиотекой Gtk2).
3. Qt:
   * Qt - это еще один набор библиотек для создания GUI-приложений. Он также популярен и используется в таких средах, как KDE. Вы можете создавать GUI-приложения в Linux-скриптах с использованием Qt и языков программирования, таких как Python (с библиотекой PyQt) или C++.
4. FLTK:
   * FLTK (Fast, Light Toolkit) - это компактный набор библиотек для создания легких и быстрых GUI-приложений. Он может быть использован в скриптах для создания простых оконных интерфейсов.
5. Zenity и Yad:
   * Zenity и Yad - это утилиты командной строки, которые позволяют создавать простые диалоговые окна и оконные интерфейсы из скриптов. Они основаны на библиотеке GTK и могут использоваться для создания оконных интерфейсов в скриптах Bash.
6. Dialog:
   * Dialog - это еще одна утилита командной строки, которая предоставляет различные виды диалоговых окон, такие как окна выбора, ввода и т. д. Она основана на библиотеке ncurses и может быть использована в скриптах.

Эти инструменты и библиотеки предоставляют разные уровни сложности и функциональности для создания GUI в Linux-скриптах, и выбор зависит от ваших потребностей и предпочтений.

**Вывод:** Закрепил на практике основы работы со сценариями командной оболочки и перенаправление ввода/вывода.