## 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 -

Выполнили:студенты группы 10701222 Медведский Е.В Зухта К.М.

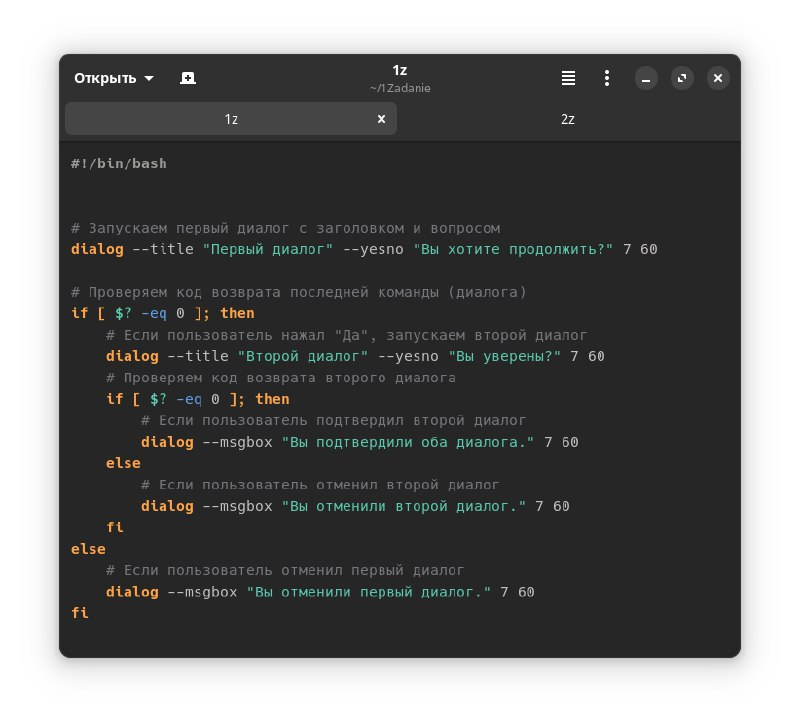
Принял**:** пр. Давыденко Н.В.

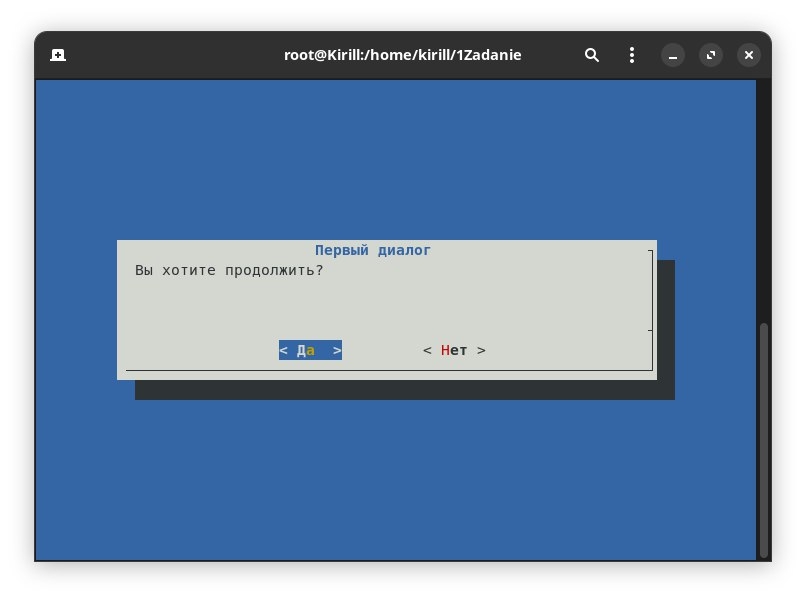
Минск 2024

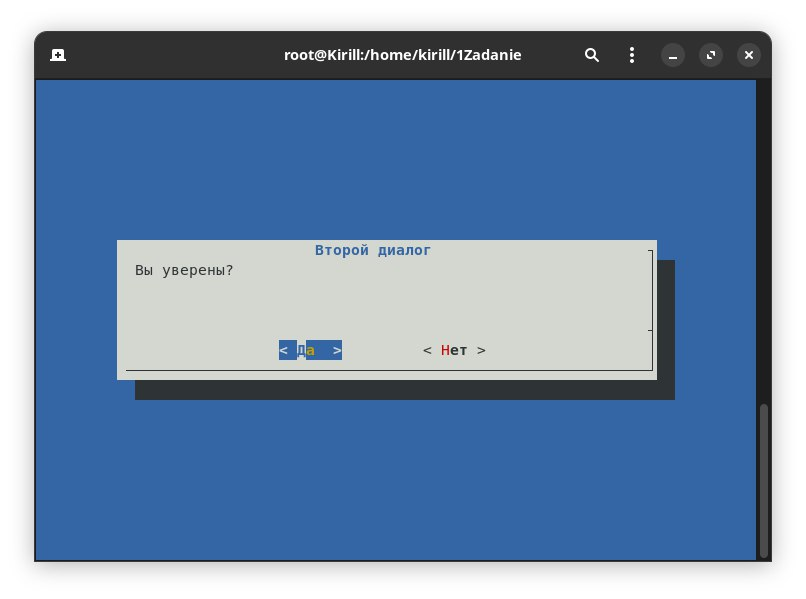
**Цель работы:**Закрепить на практике принципы создания проектов с помощью скриптов SHELL, освоить средства примитивного графического интерфейса в Linux-скриптах.

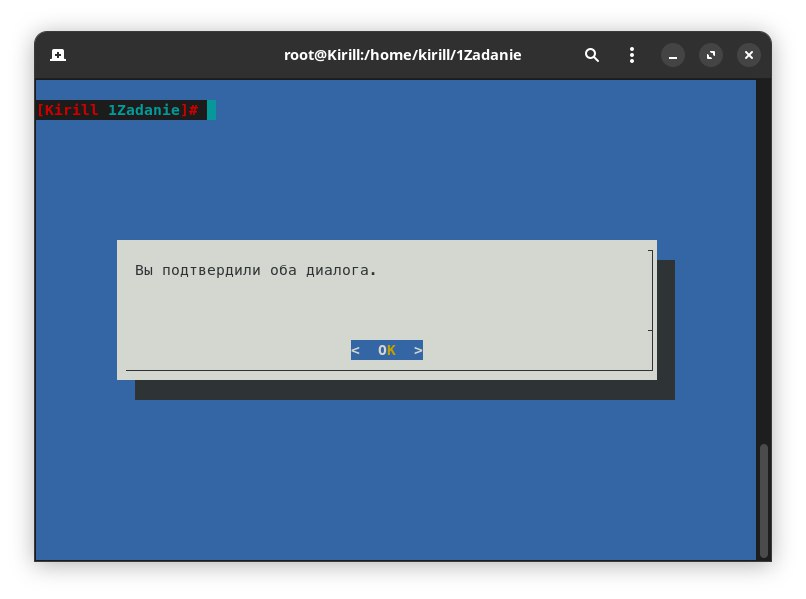
**Задание1**

1. Сделать пример с двумя вложенными диалогами типа YesNo.

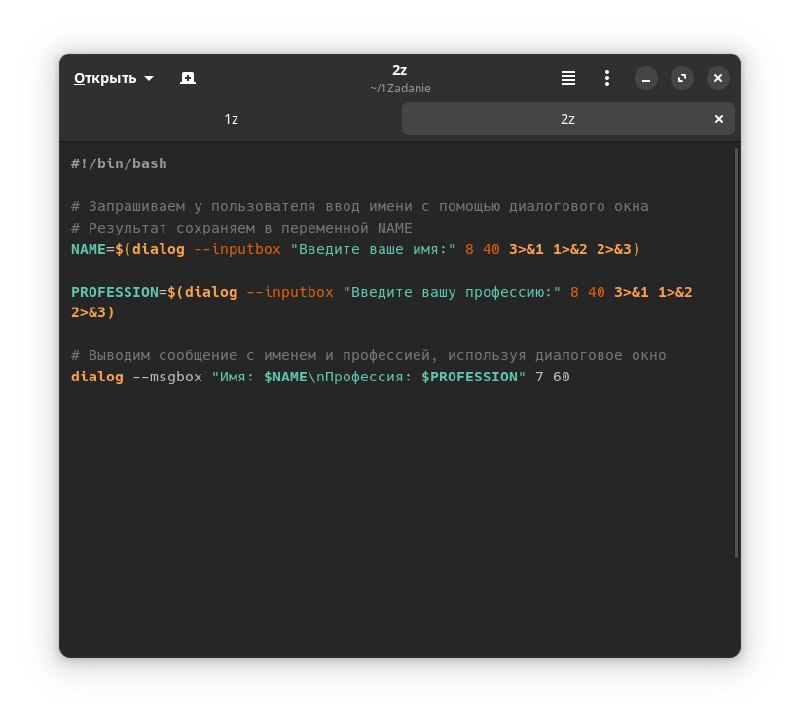
Скрипт первого задания

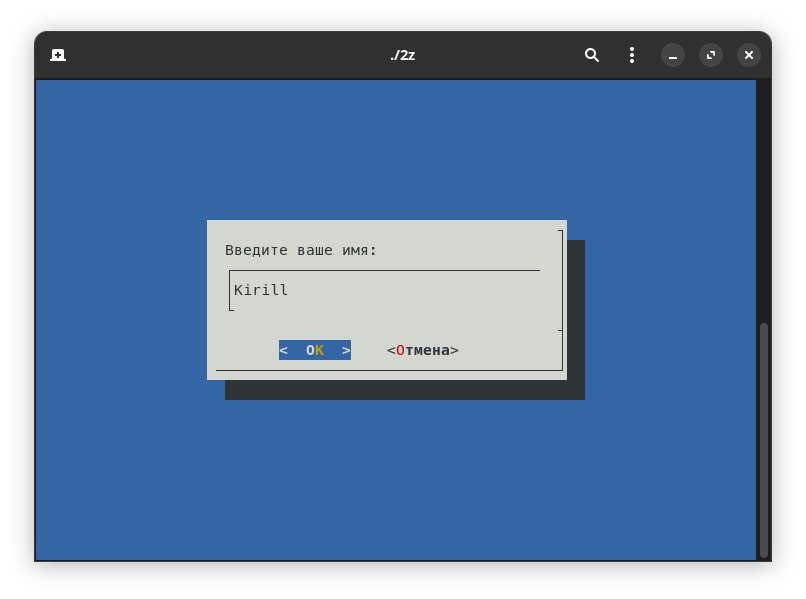


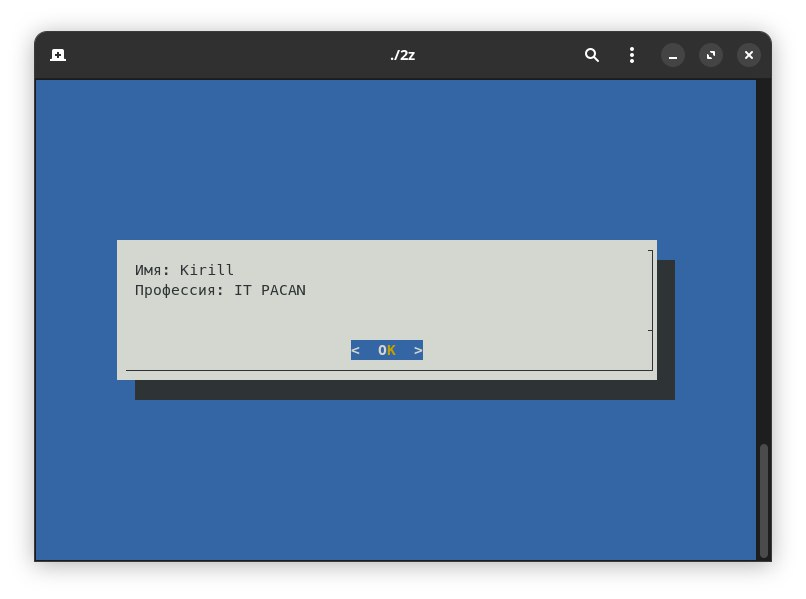
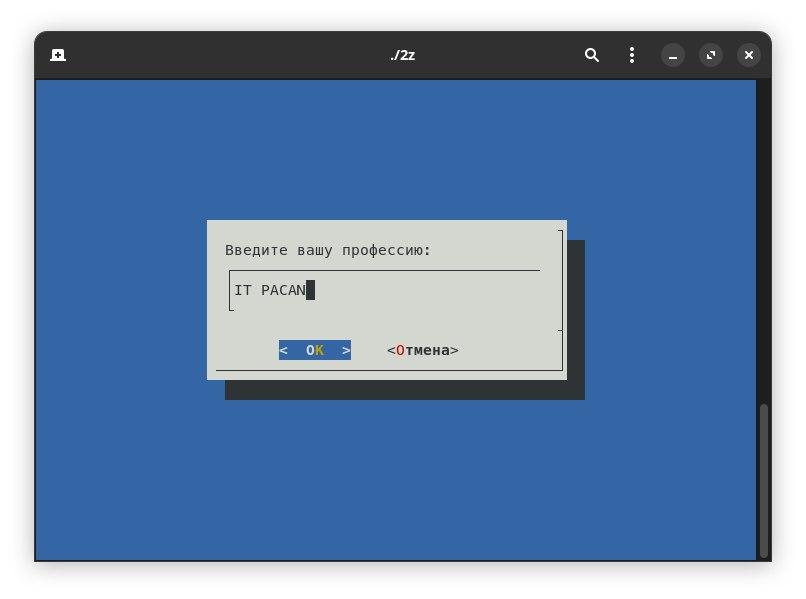


Работа первого скрипта

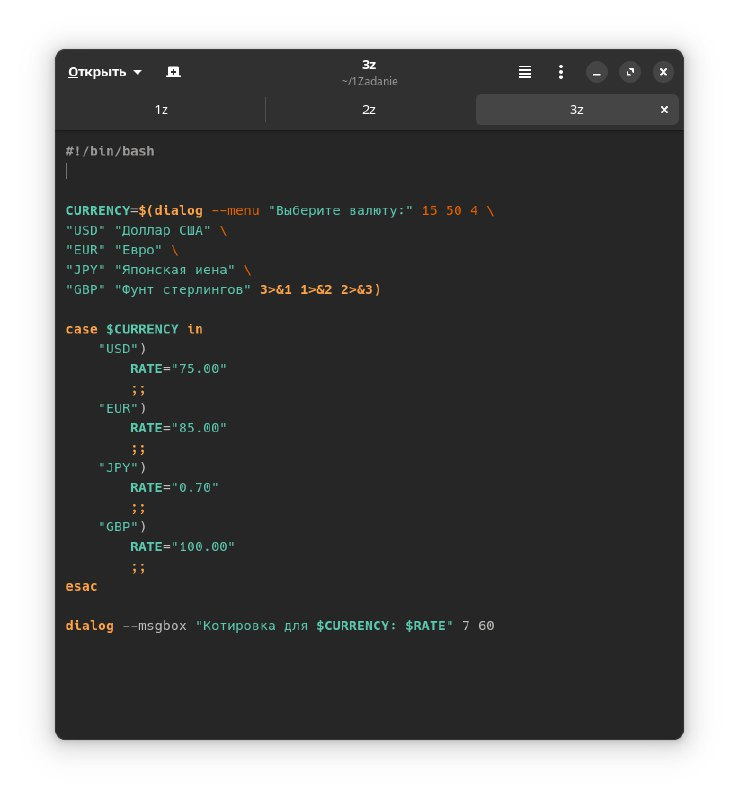
2. Сделать пример, запрашивающий сначала имя человека, а потом профессию. Вывести имя + профессию, прочитанные в диалоге.

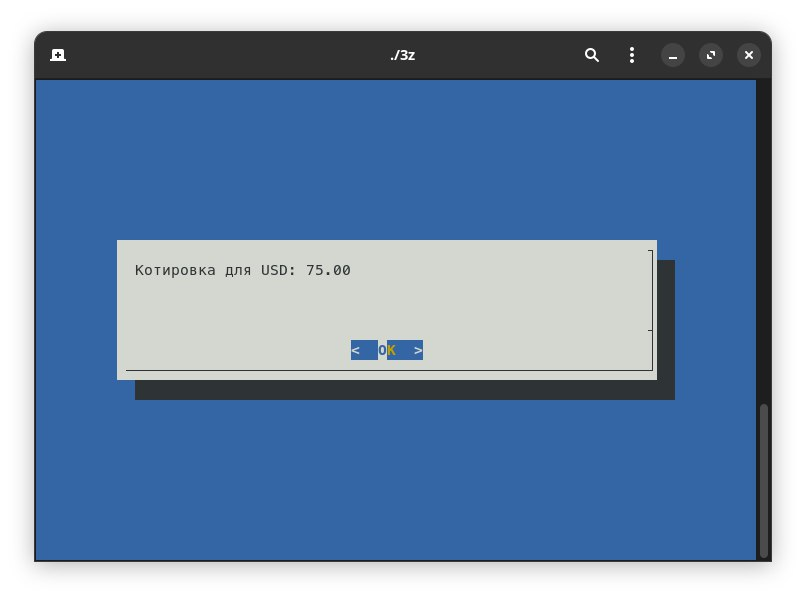
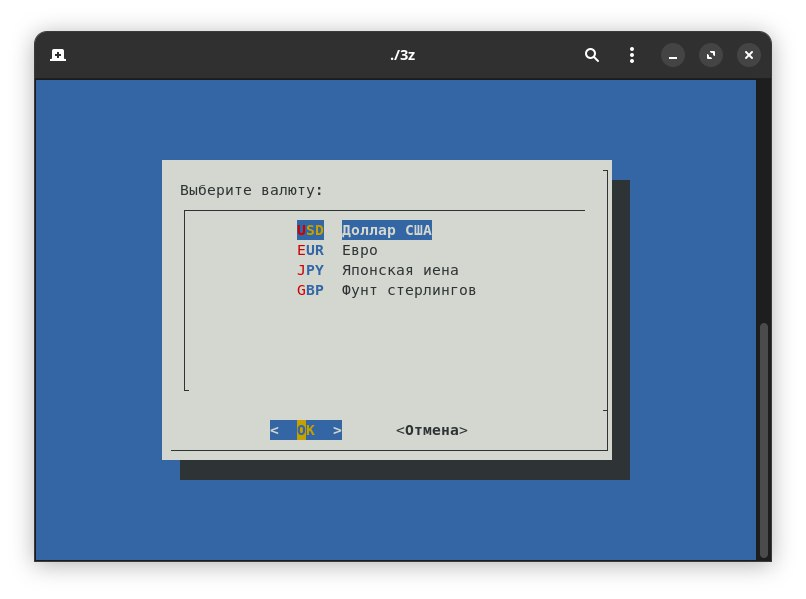
Скрипт 2 задания



Работа второго скрипта

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

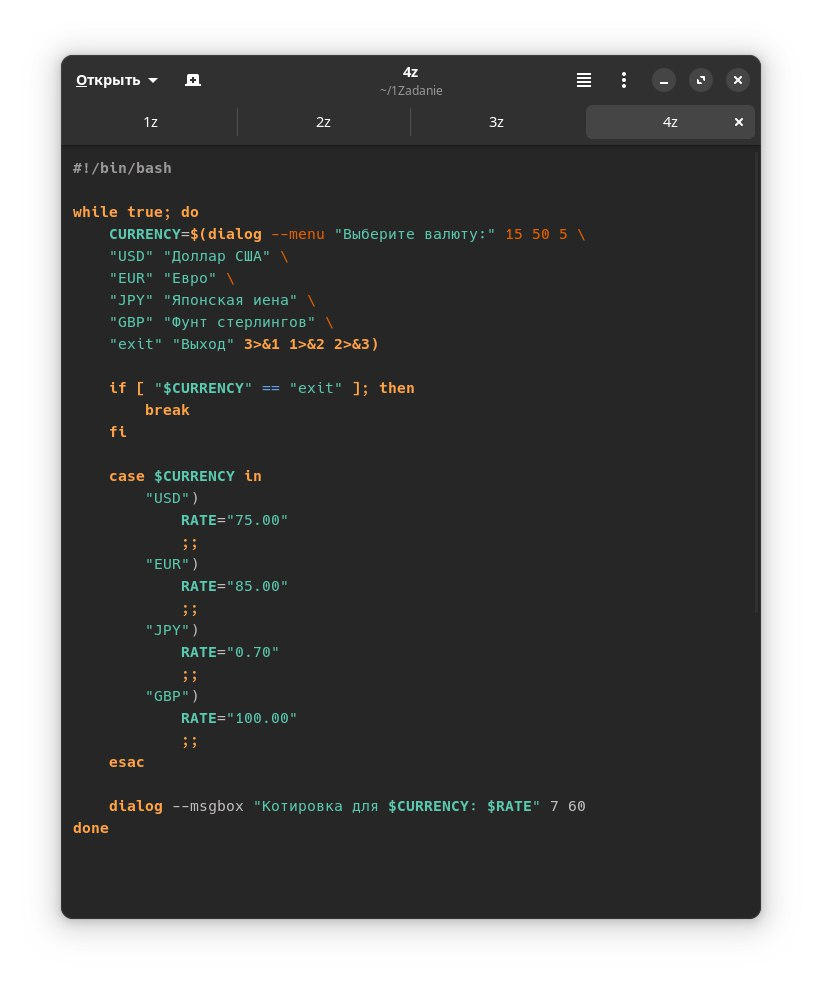
Скрипт третьего задания

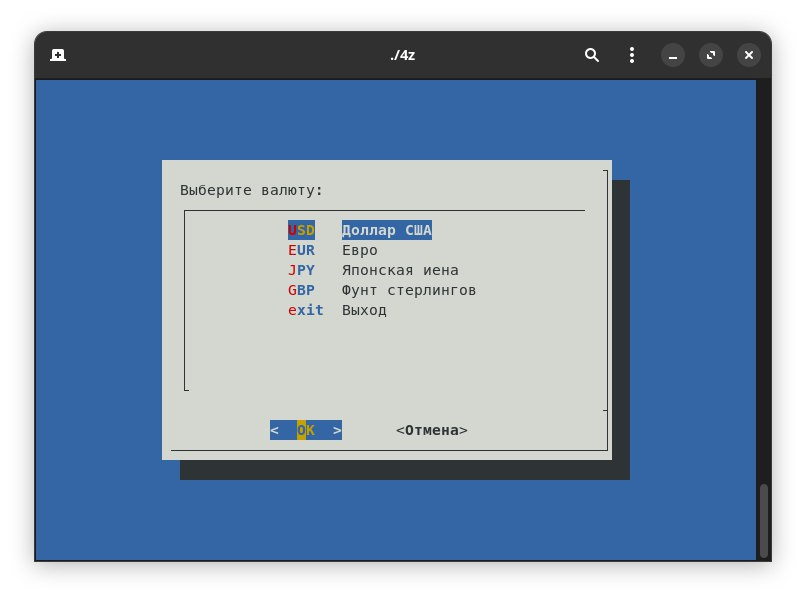
Работа скрипта 3

4. Измените предыдущую программу так, вместо списка валют

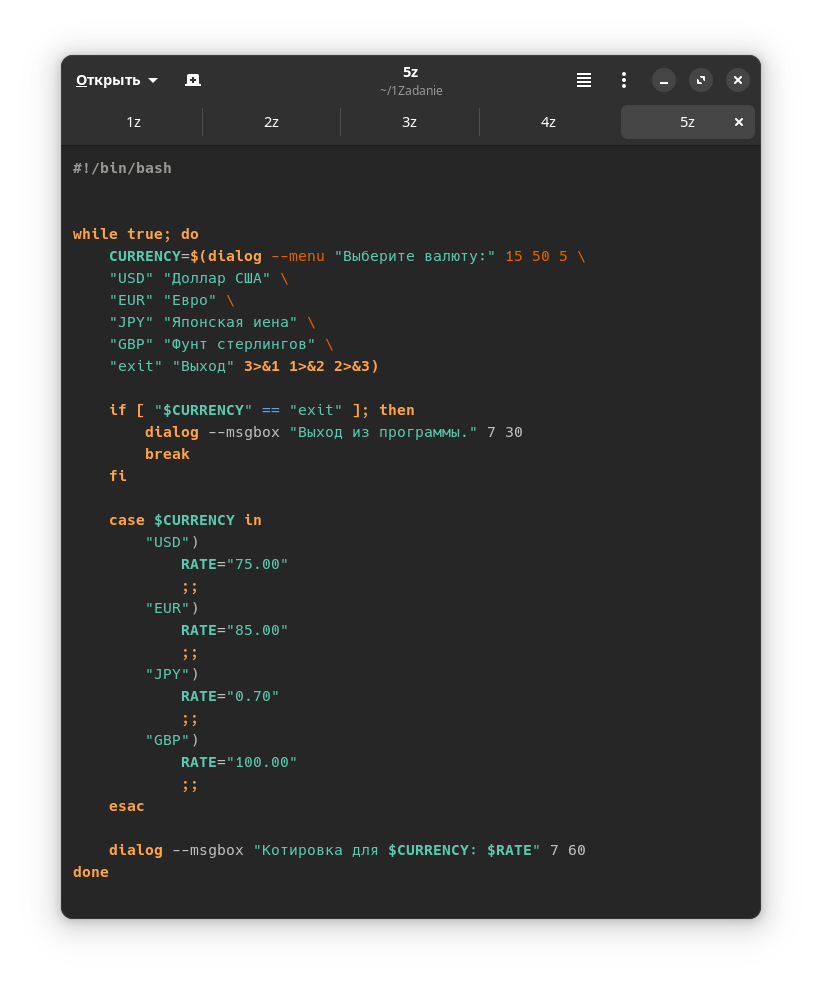
предлагалось оконное меню валют. Чтобы программа работала в цикле.

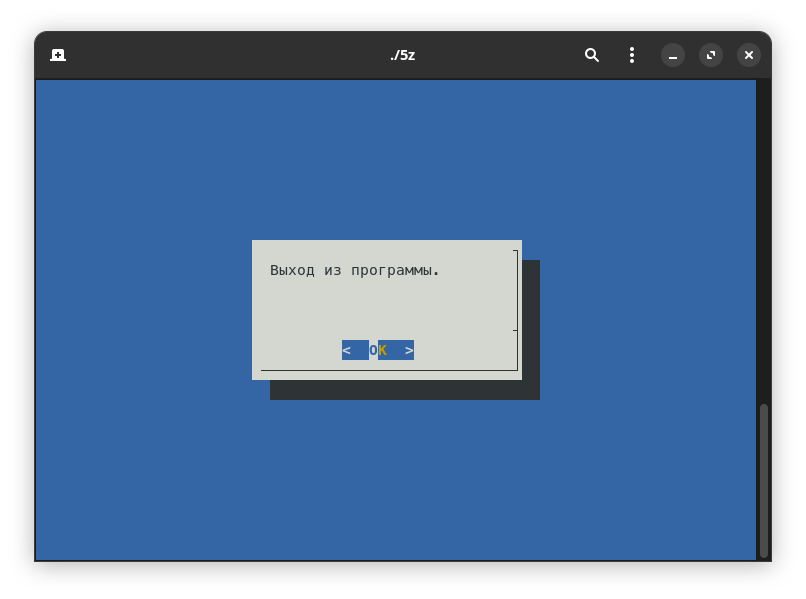
Для выхода из цикла нужно вместо названия валюты вводить exit.

Четвертый скрипт

Работа 4 скрипта

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

Скрипт 5 задания



Работа скрипта 5

**Задание2**

код программы

#!/bin/bash

# sys\_info\_page: программа вывода страницы с информацией о системе

PROGNAME=$(basename $0)

TITLE="System Information Report For $HOSTNAME"

CURRENT\_TIME=$(date +"%x %r %Z")

TIME\_STAMP="Generated $CURRENT\_TIME, by $USER"

report\_uptime () {

cat <<- \_EOF\_

<H2>System Uptime</H2>

<PRE>$(uptime)</PRE>

\_EOF\_

return

}

report\_disk\_space () {

cat <<- \_EOF\_

<H2>Disk Space Utilization</H2>

<PRE>$(df -h)</PRE>

\_EOF\_

return

}

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

}

write\_html\_page () {

cat <<- \_EOF\_

<HTML>

<HEAD>

<TITLE>$TITLE</TITLE>

</HEAD>

<BODY>

<H1>$TITLE</H1>

<P>$TIME\_STAMP</P>

$(report\_uptime)

$(report\_disk\_space)

$(report\_home\_space)

</BODY>

</HTML>

\_EOF\_

return

}

# Интерактивное меню с использованием Zenity для выбора действия

show\_menu() {

ACTION=$(zenity --list --title="System Information Menu" --column="Option" --text="Select an option:" \

"Display System Uptime" \

"Display Disk Space" \

"Display Home Space Utilization" \

"Generate Full HTML Report" \

"Exit")

case "$ACTION" in

"Display System Uptime")

uptime;;

"Display Disk Space")

df -h;;

"Display Home Space Utilization")

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

du -sh /home/\*

else

du -sh $HOME

fi;;

"Generate Full HTML Report")

filename="sys\_info\_report.html"

write\_html\_page > "$filename"

xdg-open "$filename" # открываем отчет в браузере

;;

"Exit")

exit 0;;

\*)

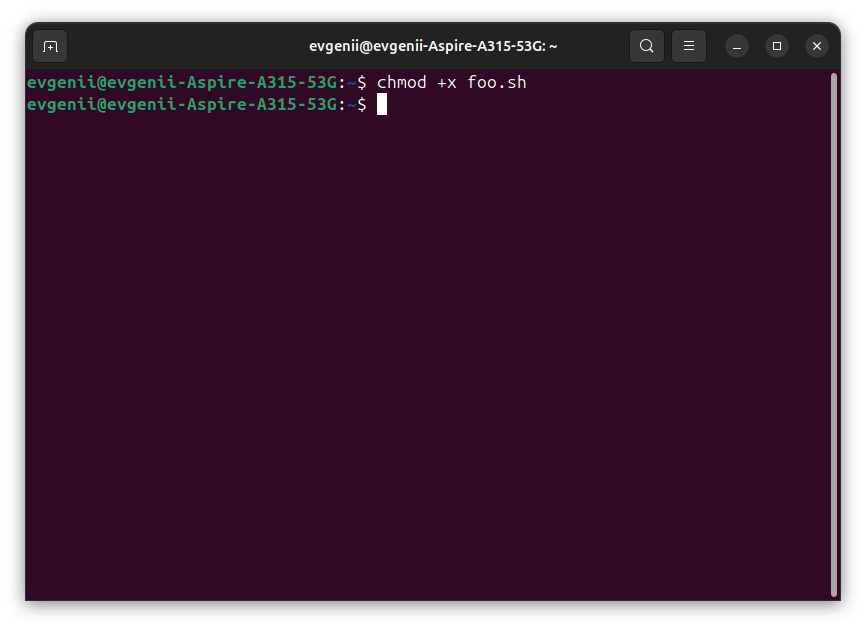
zenity --error --text="Invalid selection.";;

esac

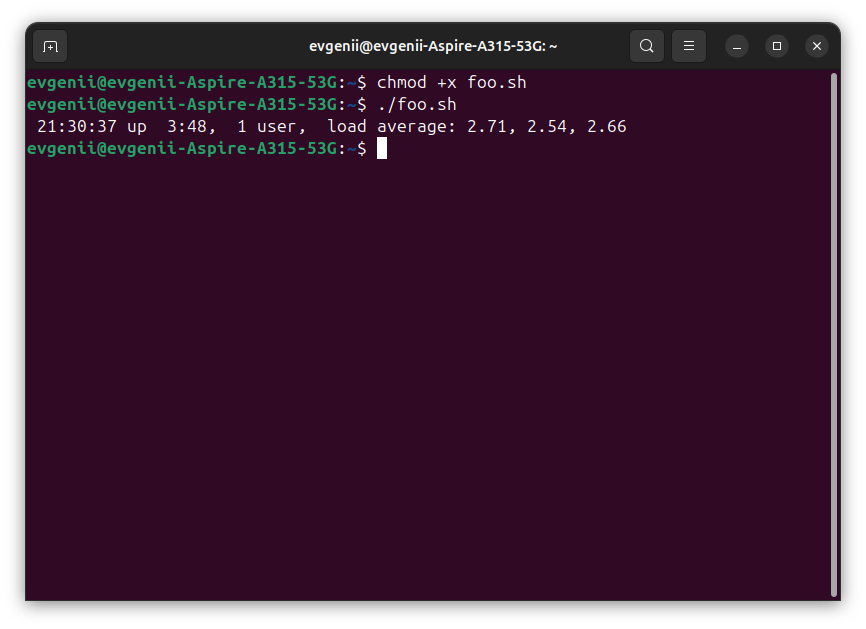
}

# Запускаем меню

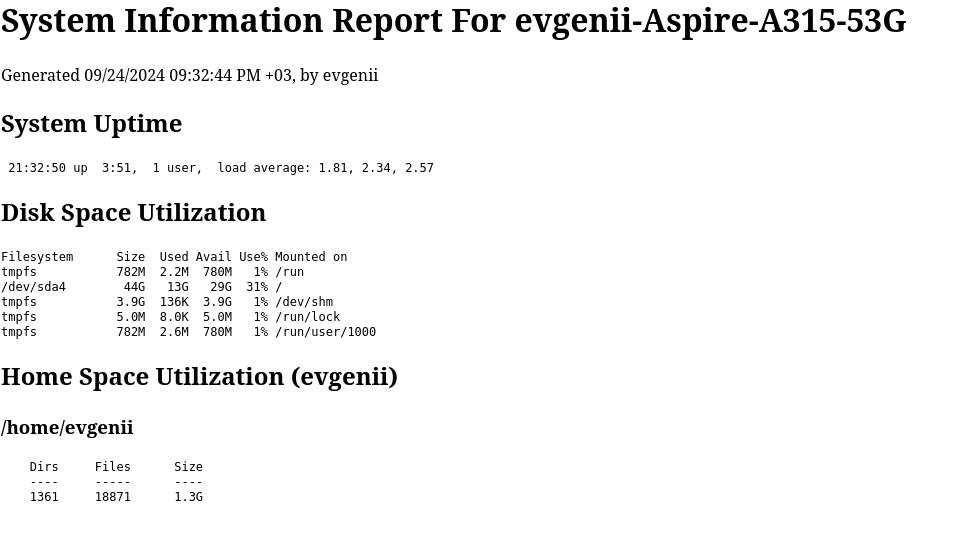
show\_menu



Делаем файл исполняемым

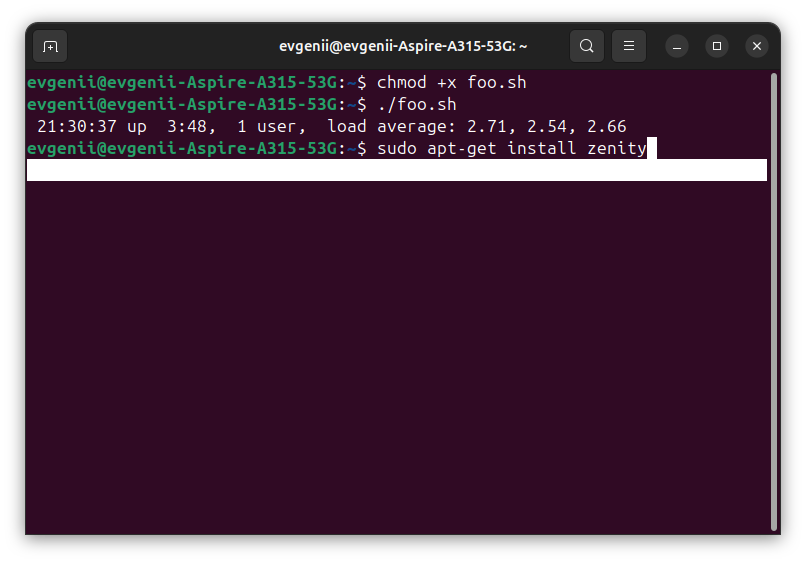


получили результат в консоли

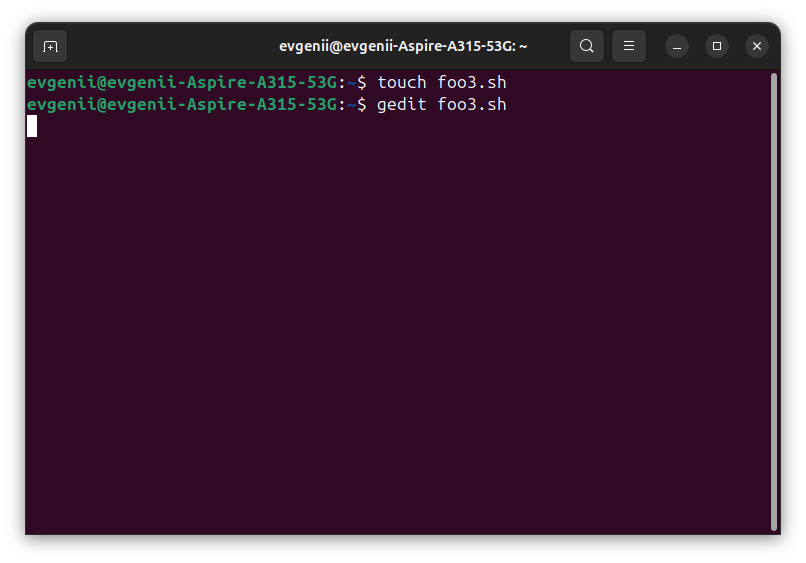


получаем результат в виде html страницы

**Задание3**



устанавливаем zenity для отображения графического интерфейса



создаем и редактируем файл foo3.sh для 3го задания

#!/bin/bash

# sys\_info\_page: программа для отображения информации о системе с использованием графического меню Zenity

PROGNAME=$(basename $0)

TITLE="System Information Report For $HOSTNAME"

CURRENT\_TIME=$(date +"%x %r %Z")

TIME\_STAMP="Generated $CURRENT\_TIME, by $USER"

FILENAME="sys\_info\_report.html"

# Функция для отчета о времени работы системы

report\_uptime () {

cat <<- \_EOF\_

<H2>System Uptime</H2>

<PRE>$(uptime)</PRE>

\_EOF\_

}

# Функция для отчета об использовании дискового пространства

report\_disk\_space () {

cat <<- \_EOF\_

<H2>Disk Space Utilization</H2>

<PRE>$(df -h)</PRE>

\_EOF\_

}

# Функция для отчета об использовании пространства домашней директории

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

}

# Функция для генерации HTML-отчета

write\_html\_page () {

cat <<- \_EOF\_ > $FILENAME

<HTML>

<HEAD>

<TITLE>$TITLE</TITLE>

</HEAD>

<BODY>

<H1>$TITLE</H1>

<P>$TIME\_STAMP</P>

$(report\_uptime)

$(report\_disk\_space)

$(report\_home\_space)

</BODY>

</HTML>

\_EOF\_

}

# Функция для выбора действия через Zenity

show\_menu() {

ACTION=$(zenity --list --title="System Information Menu" --column="Option" --text="Select an option:" \

"Display System Uptime" \

"Display Disk Space" \

"Display Home Space Utilization" \

"Generate Full HTML Report" \

"Exit")

case "$ACTION" in

"Display System Uptime")

zenity --info --text="$(uptime)" --title="System Uptime";;

"Display Disk Space")

zenity --info --text="$(df -h)" --title="Disk Space Utilization";;

"Display Home Space Utilization")

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

zenity --info --text="$(du -sh /home/\*)" --title="Home Space Utilization (All Users)"

else

zenity --info --text="$(du -sh $HOME)" --title="Home Space Utilization ($USER)"

fi;;

"Generate Full HTML Report")

write\_html\_page

zenity --info --text="Report generated at $FILENAME"

xdg-open "$FILENAME" ;;

"Exit")

exit 0;;

\*)

zenity --error --text="Invalid selection.";;

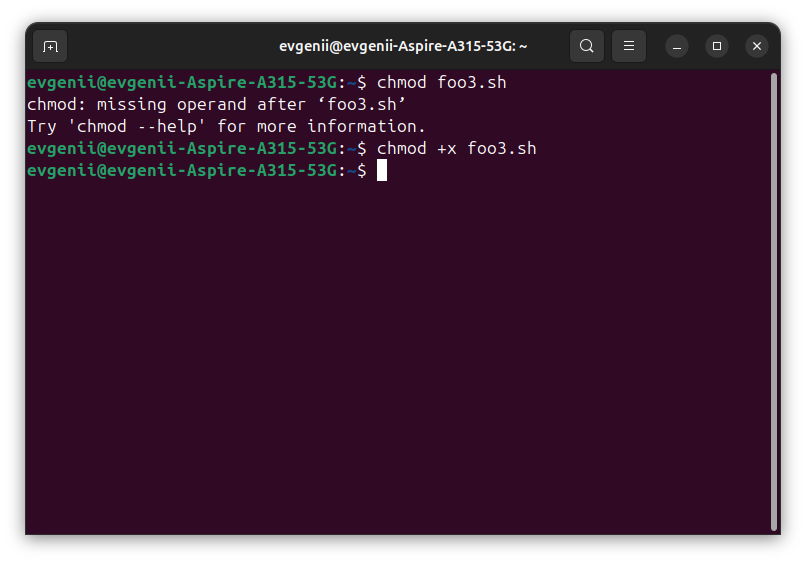
esac

}

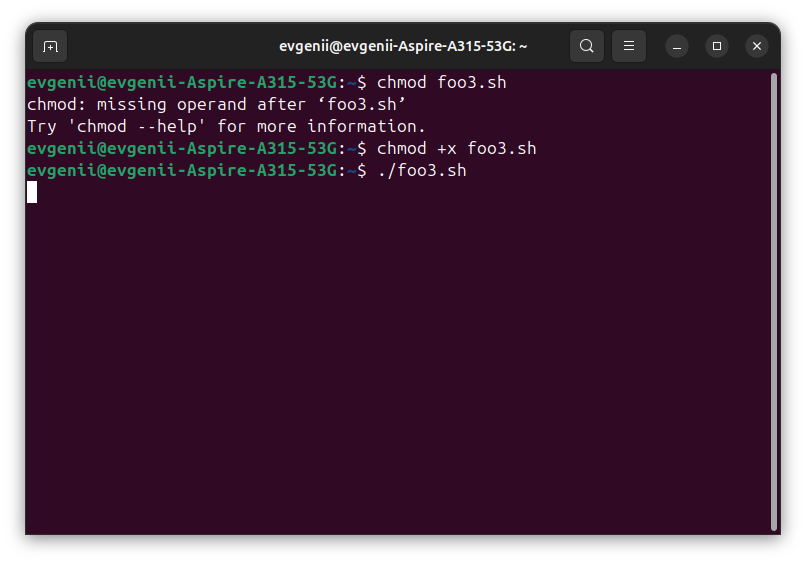
# Запускаем меню

show\_menu

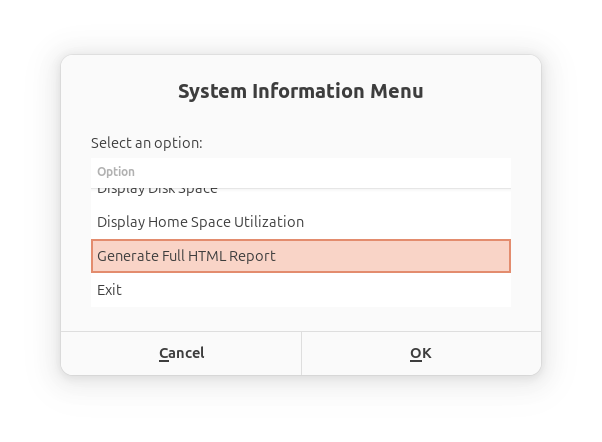
код скрипта



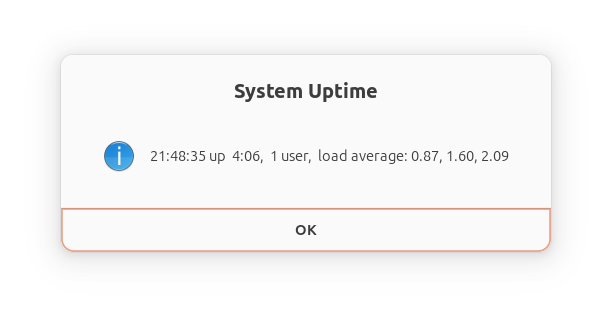
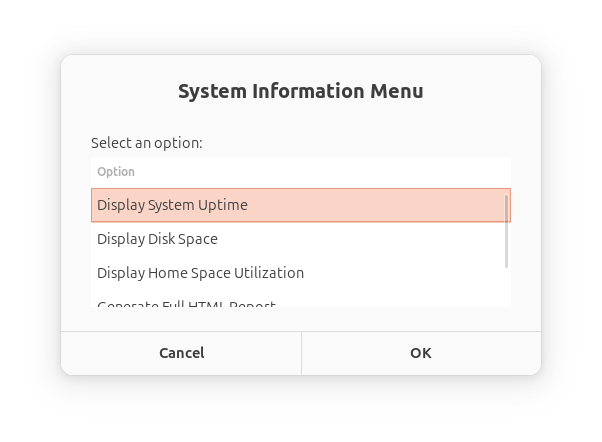
делаем файл испольняемым

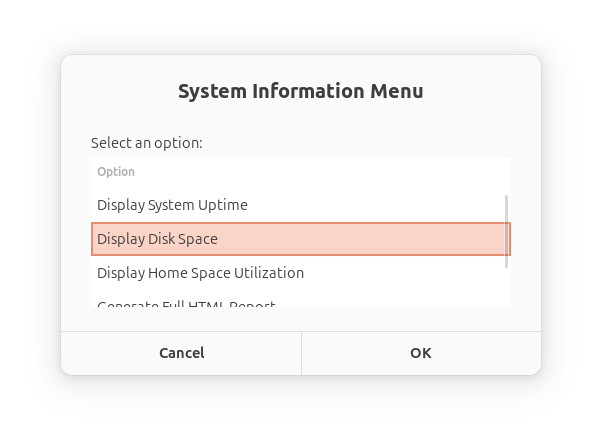


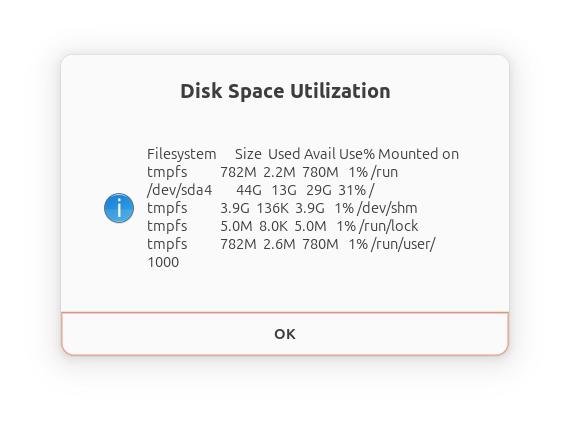
запускаем файл

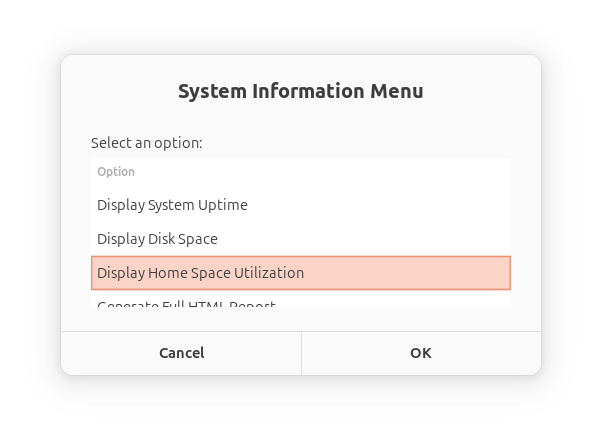
снова выбираем формат запуска

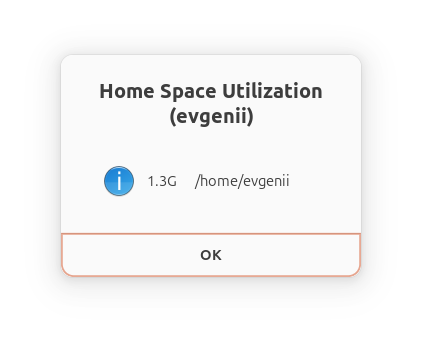
Дальше результат при каждом выборе











Ответы на вопросы:

Зухта Кирилл

1. Что такое конфликт имен переменных и как решается эта проблемма?

Ответ: это ситуация, когда две или более переменных в системе имеют одинаковое имя. Это может привести к непредсказуемому поведению программ, так как операционная система не всегда сможет правильно определить, к какой из них обращаться.

**Решения:**

• **Используйте уникальные имена**

**Используйте префиксы: Добавляйте префиксы к именам переменных, чтобы их легче отличать от стандартных переменных. Например, вместо** PATH**, вы можете использовать** MY\_PATH**.**

**Проверьте переменные среды: Перед созданием новых переменных, убедитесь, что нет уже существующих переменных с таким же именем.**

2. С помощью каких утилит реализуется средства примитивного

графического интерфейса в Linux-скриптах?

Ответ: **1.** dialog

**Описание:** Самая популярная утилита для создания диалоговых окон. Она предоставляет множество стандартных элементов, таких как текстовые поля, флажки, списки, парольные поля и т.д.

**2.** zenity

**Описание:** Другая популярная утилита, более современная, чем dialog. Предоставляет множество элементов, и хорошо интегрируется с окружением рабочего стола GNOME.

**3.** whiptail

**Описание:** Утилита, схожая с dialog, но с более простым синтаксисом.

Медведский Евгений

10.1 Какой командой и каким средством в скриптах обеспечиваетсяинтерактивность программы?

### ****Команда**** read****:****

Это одна из самых простых команд для взаимодействия с пользователем через терминал. Она позволяет запросить ввод от пользователя и сохранить его в переменной.

### ****Команда**** select****:****

select используется для создания интерактивного меню, позволяющего пользователю выбирать из списка опций.

### ****Графические диалоговые окна с использованием**** Zenity ****(или**** dialog****):****

Для создания графических окон в скриптах Bash можно использовать инструменты, такие как Zenity (в среде с графическим интерфейсом) или dialog (для текстового интерфейса в терминале).