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: «Скрипты, диалоги, создание проекта»

Вариант 9

Выполнили**:**  студенты группы 10701222 Махнач М. В.

Дяденькина Т.М.

Шкробот А. А.

Приняла**:** ст. 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.

**Задание**

Задание 2

Создайте копию проекта, представленного 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.

**Выполнение**

Код приложения:

#!/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

}

usage () {

echo "$PROGNAME: usage: $PROGNAME [-f file | -i]"

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

}

# обработка параметров командной строки

interactive=

filename=

while [[ -n $1 ]]; do

case $1 in

-f | --file) shift

filename=$1

;;

-i | --interactive) interactive=1

;;

-h | --help) usage

exit

;;

\*) usage >&2

exit 1

;;

esac

shift

done

# интерактивный режим

if [[ -n $interactive ]]; then

while true; do

read -p "Enter name of output file: " filename

if [[ -e $filename ]]; then

read -p "'$filename' exists. Overwrite? [y/n/q] > "

case $REPLY in

Y|y) break

;;

Q|q) echo "Program terminated."

exit

;;

\*) continue

;;

esac

fi

done

fi

# вывод страницы html

if [[ -n $filename ]]; then

if touch $filename && [[ -f $filename ]]; then

write\_html\_page > $filename

else

echo "$PROGNAME: Cannot write file '$filename'" >&2

exit 1

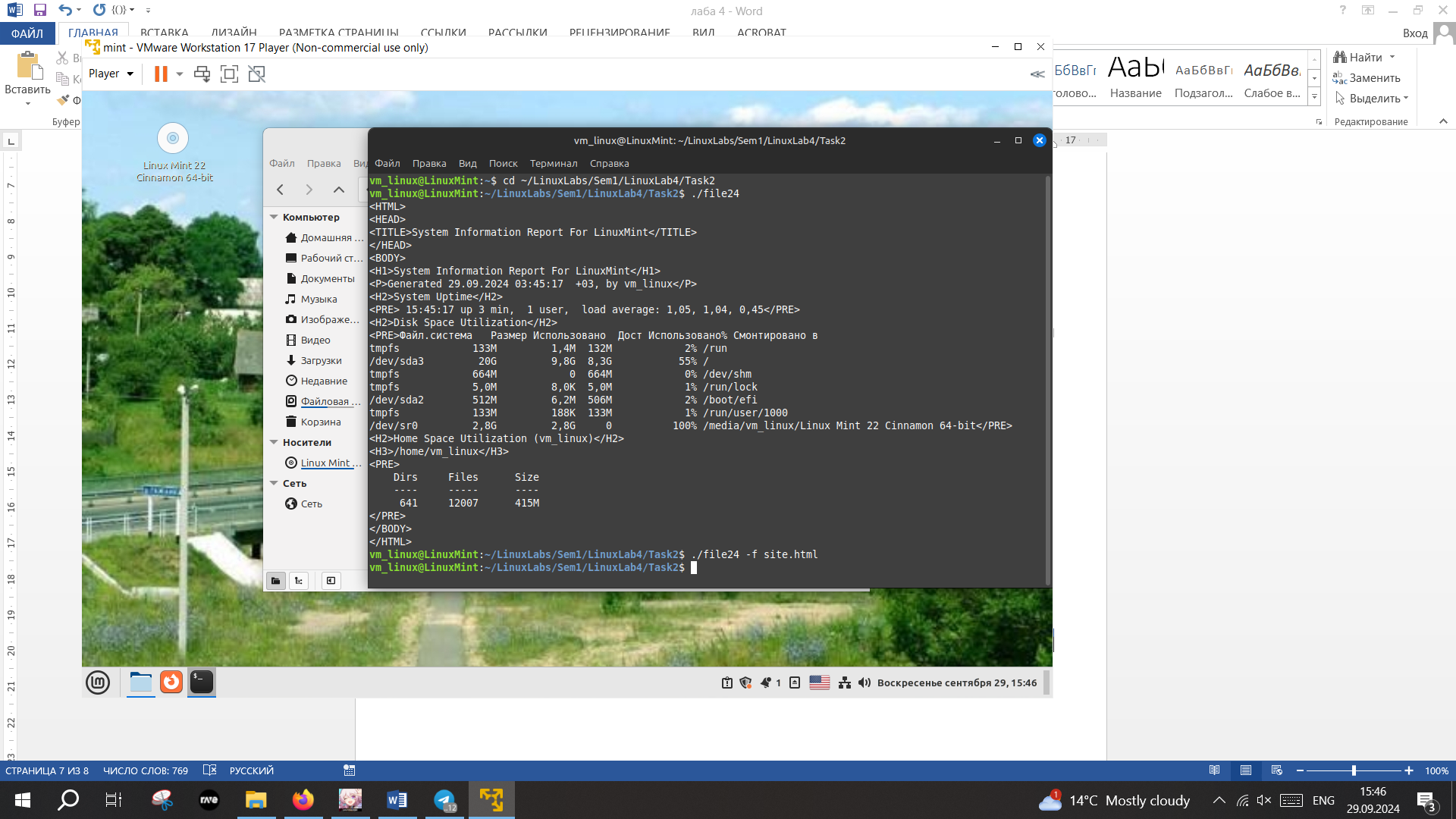
fi

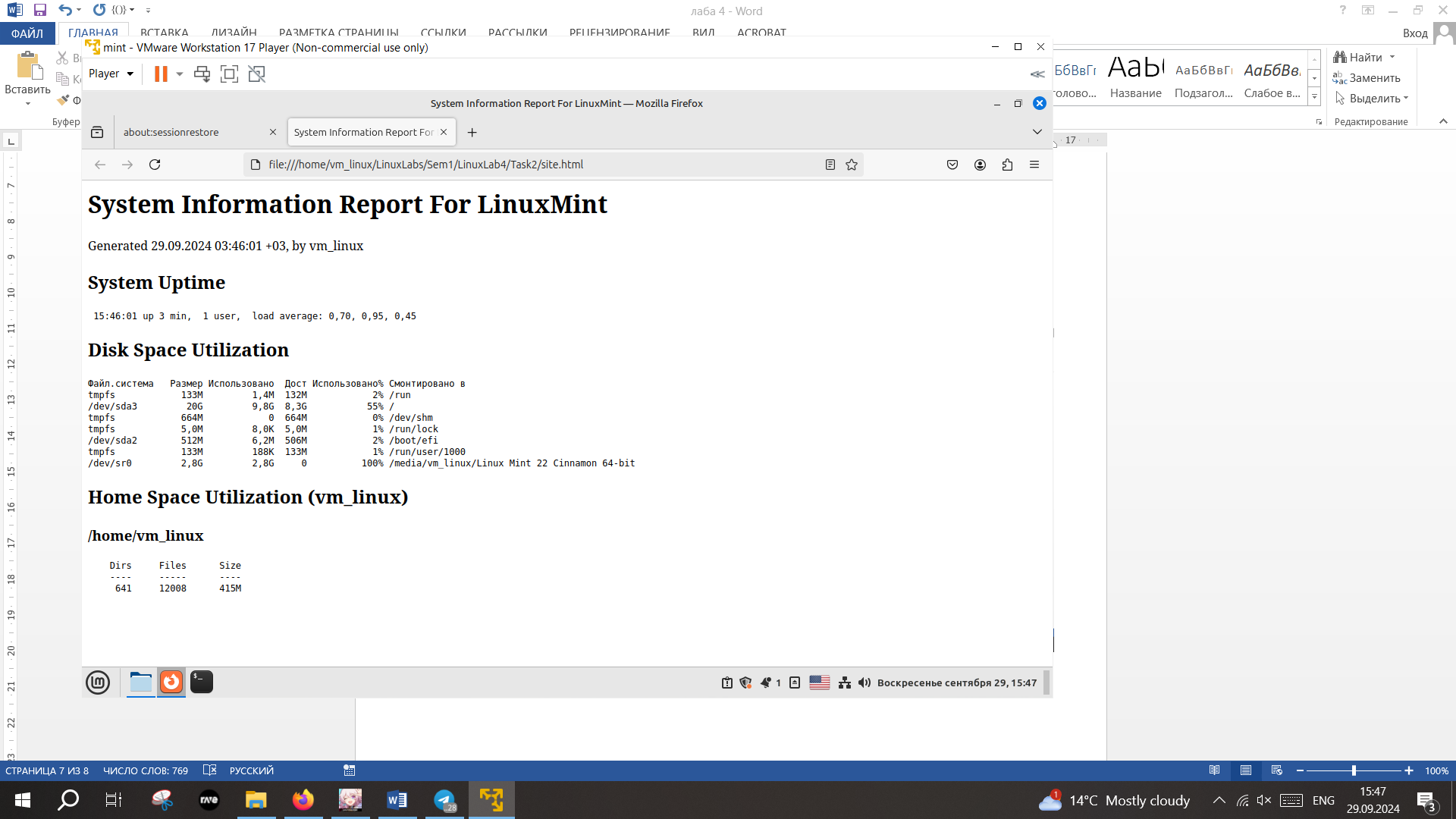
else

write\_html\_page

fi

Скриншоты работы:





В первом случае не используются никакие флаги, и сгенерированная html-страница выводится просто текстом, в консоль. Во втором случае указывается флаг –f и файл, куда надо сохранить данные. В случае, если файл не существует, он создаётся. На втором скиншоте представлено содержимое html-файла.

**Контрольные вопросы**

1. **Что такое «код завершения программы»?**

Код завершения программы (или код возврата) — это числовое значение, которое программа возвращает операционной системе при завершении своего выполнения. Этот код позволяет операционной системе (и другим программам) понять, успешно ли завершилась работа программы или произошла какая-либо ошибка. Код завершения обычно равен 0, если программа выполнена успешно, и ненулевое значение в случае ошибки. Например, в Unix-подобных системах (включая Linux) код завершения можно получить с помощью встроенной переменной `$?` после выполнения команды в командной строке.

**2) Какие существуют типы окон примитивного графического**

**интерфейса Linux-скриптов?**

Типы окон примитивного графического интерфейса в Linux-скриптах могут включать следующее:  
– Диалоговые окна: используются для отображения сообщений, запросов или получения информации от пользователя. Например, `zenity`, `yad`, `dialog` являются утилитами для создания диалоговых окон.  
– Окна ввода: позволяют пользователю вводить текст или данные. Как правило, это интегрируется в диалоговые окна.  
– Предупреждающие окна: показывают предупреждения или ошибки. Обычно они содержат кнопку "OK" или "Закрыть" для подтверждения.  
– Меню: позволяют пользователю выбирать из списка вариантов. Они могут быть реализованы в виде выпадающего меню или отдельного окна.  
– Окна с выбором файлов: предоставляют интерфейс для выбора файлов или директорий из файловой системы.