ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

Лабораторная работа По дисциплине

«Системное программное обеспечение»

«Командная оболочка BASH»

Выполнил: студент группы ИП-813

Бурдуковский И.А.

Работу принял: Профессор кафедры ВС Мамойленко С.Н.

Новосибирск 2022

Постановка задачи

1. Определить тип используемой командной оболочки.
2. Вывести на экран значения всех переменных среды окружения. Проанализировать полученные результаты и объяснить значения известных вам переменных окружения.
3. Используя процедуру экспортирования изменить приглашение командной строки так, чтобы в основном приглашении имя машины выводилось красным цветом, а в приглашении для второй строки выводился номер команды и символ ''>''.
4. Используя системную переменную HOME и список команд выполнить следующие действия: одной командой -- перейти в домашний каталог, в случае удачного перехода выдать содержимое файла /etc/passwd.
5. Используя команды printf и read, выведете приглашение пользователю ввести команду. Если пользователь нажал без ввода команды, сообщить ему об ошибке, в противном случае выполнить то, что он ввел.
6. Оформите предыдущий пункт как скрипт и выполните его.
7. Вывести значения всех переменных среды окружения в файл с именем

\~/envs.

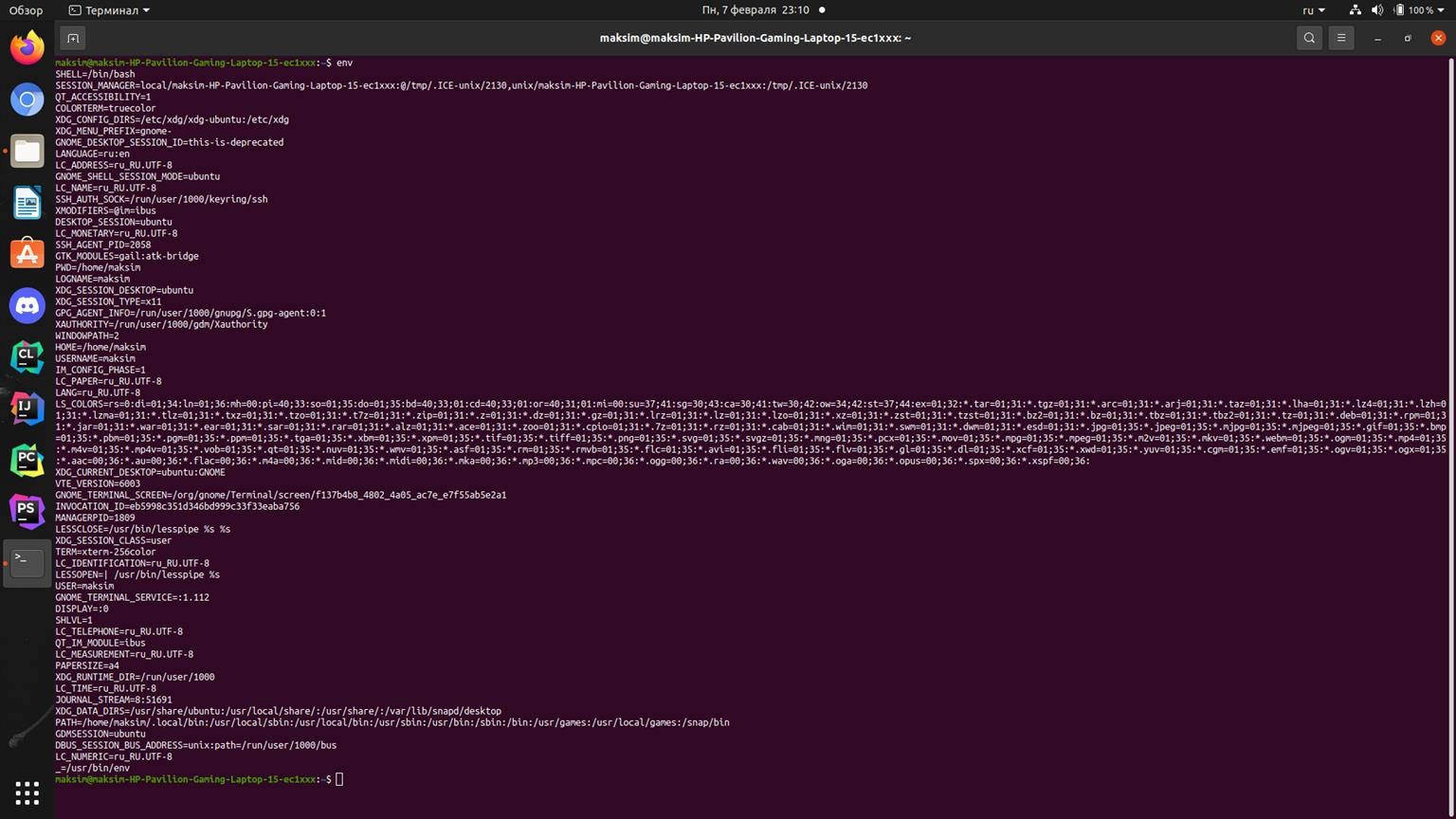
1. Используя системную переменную HOME, список, каналы и перенаправление вывода, выполнить следующие действия одной командой

* перейти в домашний каталог, выдать содержимое файла /etc/passwd, отсортированное по имени пользователя в файл passwd.orig. Подсказка - команда сортировки – sort.

1. Используя перенаправление ввода с разделителем и перенаправление вывода, добавить в файл passwd.orig информацию о себе согласно формату записи файла /etc/passwd (все поля должны быть обязательно заполнены).
2. Описать содержимое файла ~/.bash\_profile и всех файлов, которые он использует.
3. Написать скрипт, выполняющий следующие действия: выводит меню, содержащее все файлы с расширением .c текущего каталога. После выбора пользователем файла, компилирует его.
4. Написать скрипт, анализирующий параметры командной строки с использованием специальных переменных. Все параметры должны быть выданы на экран.
5. Написать скрипт, анализирующий параметры командной строки. Параметры должны быть следующие - -d каталог, -f файл, -c, -r. При анализе опций должны быть установлены переменные: DIR, FILE, COMPIL, RUN. Поле анализа опций выполнить следующие действия: если определена переменная DIR и такой каталог существует, то выдать его содержимое. Если определена переменная FILE и такой файл существует, то выдать его содержимое на экран. Если переменная не определена, то в качестве имени файла использовать .bashrc. Если определена переменная COMPIL и определена переменная FILE, то откомпилировать указанный файл. Если результат компиляции положительный, то, если определена переменная RU, исполнить откомпилированный файл.
6. Используя цикл for объединить все файлы с расширением txt в текущем каталоге в файл \~/textx.txt. Для объединения использовать перенаправление потоков ввода-вывода.

Выполнение работы

1. Определить тип используемой командной оболочки.
2. Вывести на экран значения всех переменных среды окружения. Проанализировать полученные результаты и объяснить значения известных вам переменных окружения.



USER: текущее имя пользователя, использующего систему HOME: домашний каталог текущего пользователя

PATH: список каталогов, разделенных двоеточиями, в которых система ищет команды

PWD: текущий рабочий каталог

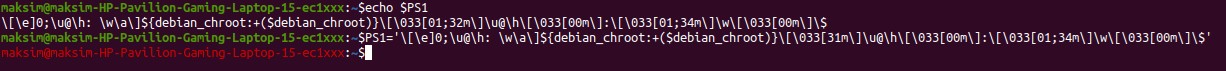
\_: самая последняя команда, выполненная в системе пользователем SHELL: оболочка, используемая для интерпретации команд в системе, она может быть много разных (например, bash, sh, zsh или другие)

LANG: кодировка языка, используемая в системе DESKTOP\_SESSION: текущий рабочий стол, используемый на вашем хосте (GNOME, KDE)

TERM: указывает тип терминала LANGUAGE=ru:en: языки используемые системой

COLORTERM=truecolor: цветовая палитра терминала LC\_ADDRESS=ru\_RU.UTF-8: локальный адрес(регион, часовой пояс) DESKTOP\_SESSION=ubuntu: тип операционной системы LOGNAME=maksim: имя авторизованного пользователя

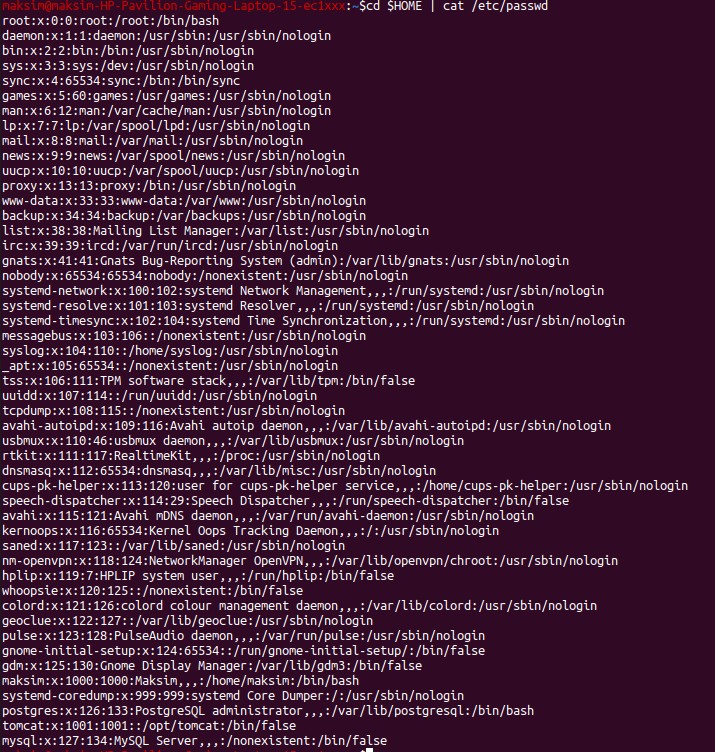
1. Используя процедуру экспортирования изменить приглашение командной строки так, чтобы в основном приглашении имя машины выводилось красным цветом, а в приглашении для второй строки выводился номер команды и символ ''>''.

Изменение PS1:

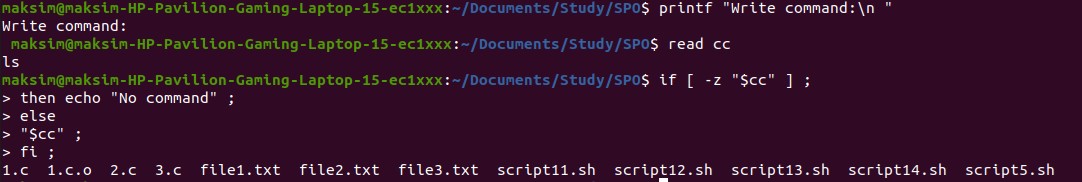
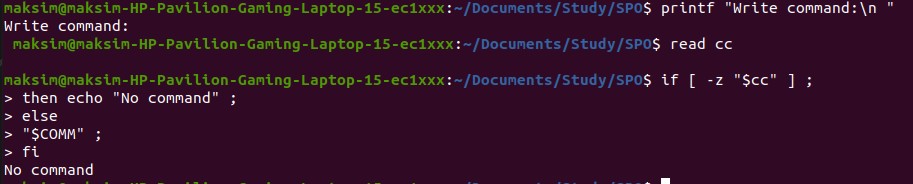
Изменение PS2:



1. Используя системную переменную HOME и список команд выполнить следующие действия: одной командой -- перейти в домашний каталог, в случае удачного перехода выдать содержимое файла /etc/passwd.



1. Используя команды printf и read, выведете приглашение пользователю ввести команду. Если пользователь нажал без ввода команды, сообщить ему об ошибке, в противном случае выполнить то, что он ввел.



1. Оформите предыдущий пункт как скрипт и выполните его.

# script5

#!/bin/bash

printf "Write command: " read COMM

if [ -z "$COMM" ] ;

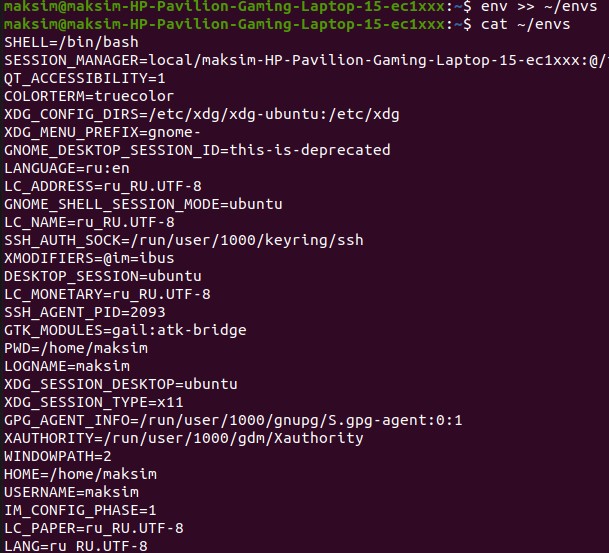
then echo "No command" ; else

$COMM ;

fi ;

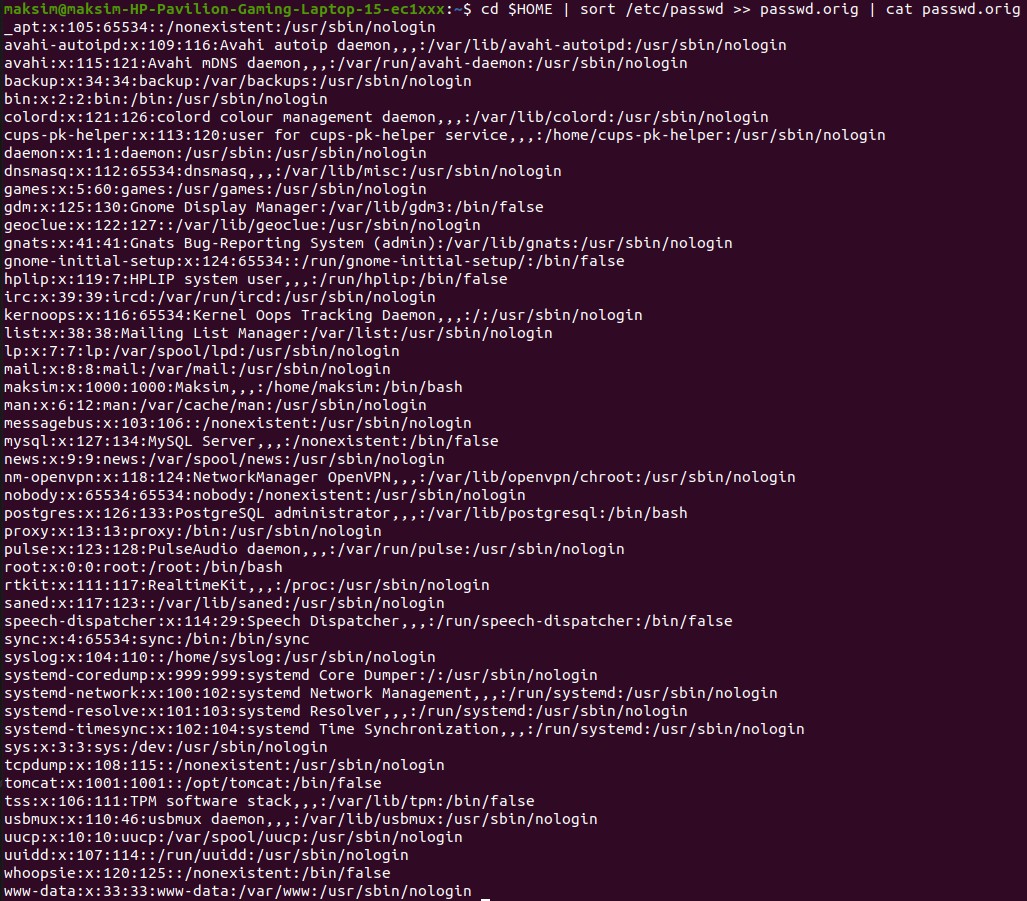
1. Вывести значения всех переменных среды окружения в файл с именем

\~/envs.

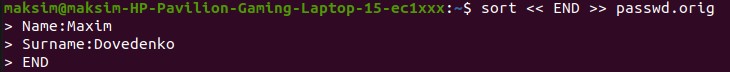


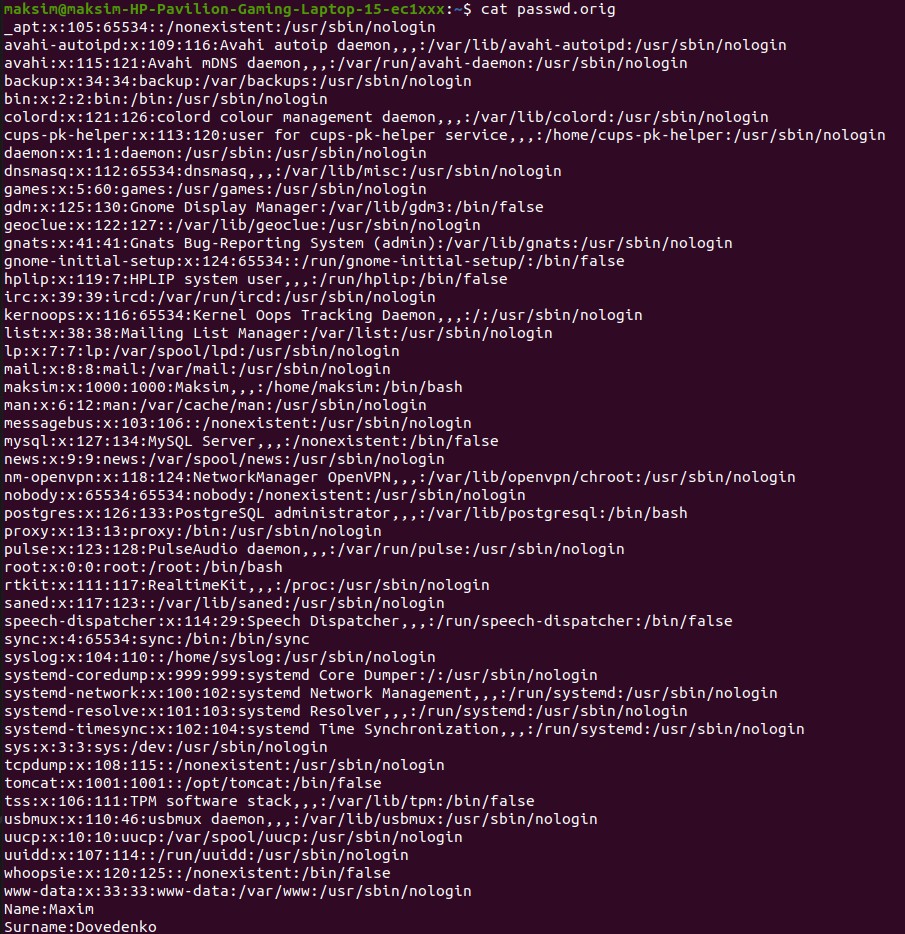
1. Используя системную переменную HOME, список, каналы и перенаправление вывода, выполнить следующие действия одной командой

* перейти в домашний каталог, выдать содержимое файла /etc/passwd, отсортированное по имени пользователя в файл passwd.orig. Подсказка - команда сортировки – sort.



1. Используя перенаправление ввода с разделителем и перенаправление вывода, добавить в файл passwd.orig информацию о себе согласно формату записи файла /etc/passwd (все поля должны быть обязательно заполнены).



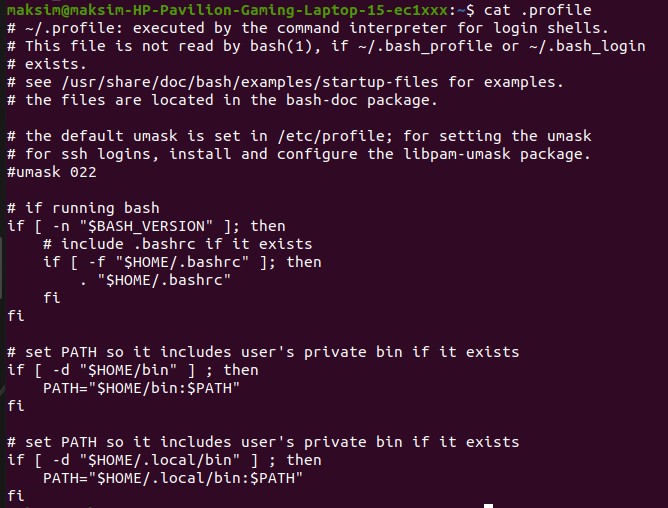


1. Описать содержимое файла ~/.bash\_profile и всех файлов, которые он использует.

В дистрибутивах Debian 5/6/7 данного файла по умолчанию не существует.

Если в домашней директории пользователя не существует ни сценария

~/.bash\_profile, ни сценария ~/bash\_login, командная оболочка bash будет проверять наличие сценария ~/.profile и исполнять его.



1. Написать скрипт, выполняющий следующие действия: выводит меню, содержащее все файлы с расширением .c текущего каталога. После выбора пользователем файла, компилирует его.

# script11.sh:

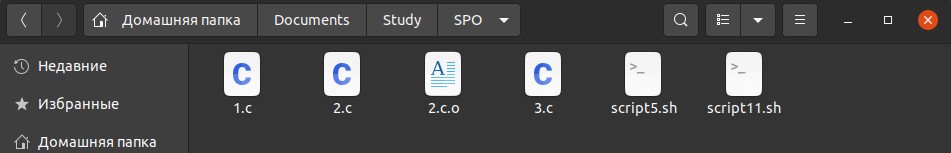
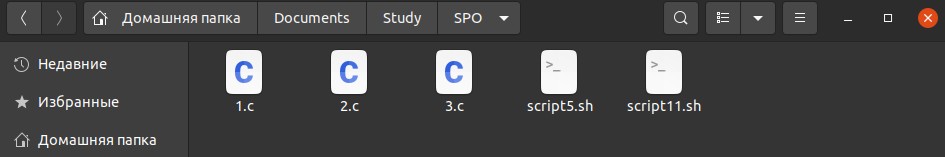
#!/bin/bash

options=$( find \* -iname '\*.c' )

select option in $options ; do

gcc -Wall -o $option.o $option ;

done



1. Написать скрипт, анализирующий параметры командной строки с использованием специальных переменных. Все параметры должны быть выданы на экран.

# script12.sh:

#!/bin/bash

echo There were $# parameters passed. echo " "

if [ -n "$3" ] then

echo The first parameter is $1. echo The second parameter is $2. echo The third parameter is $3. echo The last parameter was ${!#}.

else

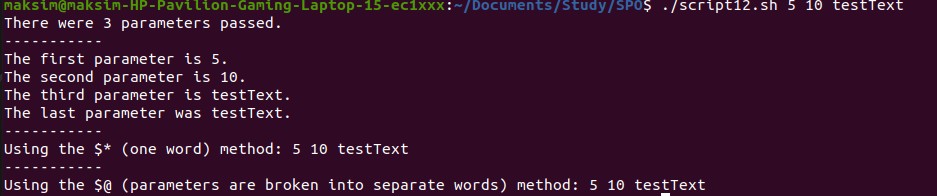
echo "Not enough parameters. " fi

echo " "

echo "Using the \$\* (one word) method: $\*" echo " "

echo "Using the \$@ (parameters are broken into separate words) method:

$@"



1. Написать скрипт, анализирующий параметры командной строки. Параметры должны быть следующие - -d каталог, -f файл, -c, -r. При анализе опций должны быть установлены переменные: DIR, FILE, COMPIL, RUN. Поле анализа опций выполнить следующие действия: если определена переменная DIR и такой каталог существует, то выдать его содержимое. Если определена переменная FILE и такой файл существует, то выдать его содержимое на экран. Если переменная не определена, то в качестве имени файла использовать .bashrc. Если определена переменная COMPIL и определена переменная FILE, то откомпилировать указанный файл. Если результат компиляции положительный, то, если определена переменная RUN, исполнить откомпилированный файл.

**script13.sh:**

#!/bin/bash

DIR=""

FILE=""

FILENAME=""

COMPIL=""

RUN=""

while [ -n "$1" ] ; do

echo “Имеется параметр –“ $1

if [ "$1" == "-d" ] ; then

DIR=$2

shift

fi

if [ "$1" == "-f" ] ; then

FILE=$2

shift

fi

if [ "$1" == "-c" ] ; then

COMPIL="true"

shift

fi

if [ "$1" == "-r" ] ; then

RUN="true"

shift

else

shift

fi

done

if [ "$DIR" != "" ] ; then

ls $DIR

fi

if [ "$FILE" != "" ] ; then

cat $FILE

else

cat ~/.bashrc

fi

if [[ "$COMPIL" == "true" ]] ; then

if [[ "$FILE" == \*".c" ]] ; then

FILENAME=${FILE%".c"}

echo $FILENAME

gcc -Wall -o $FILENAME.o $FILE ;

if [[ "$RUN" == "true" ]] ; then

./$FILENAME.o;

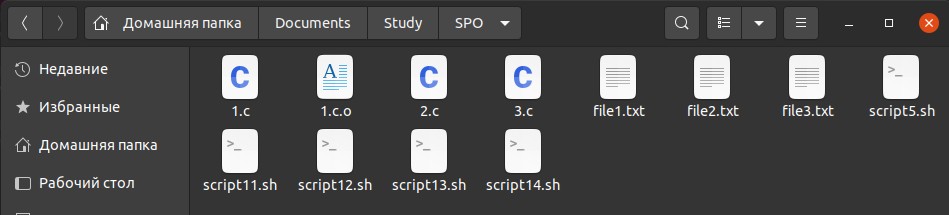
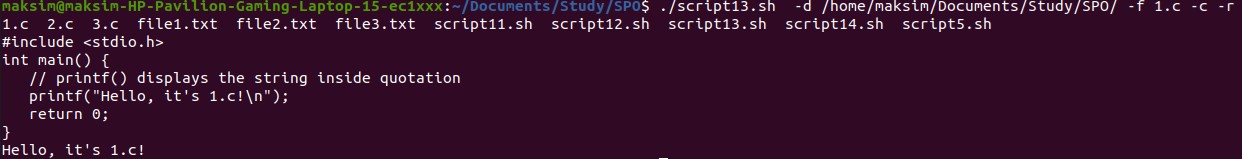
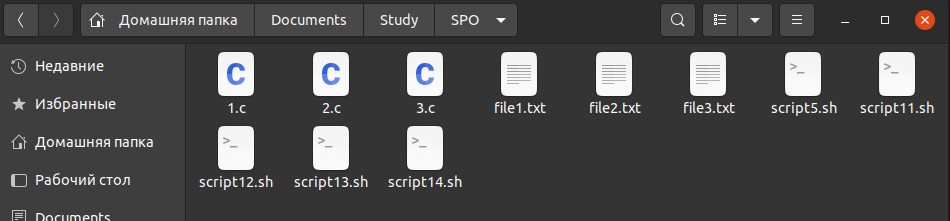
fi

else

echo "это не .с файл"

fi

fi



1. Используя цикл for объединить все файлы с расширением txt в текущем каталоге в файл \~/textx.txt. Для объединения использовать перенаправление потоков ввода-вывода.

# script14.sh:

#!/bin/bash

files=$( find \* -iname '\*.txt' )

for f in $files ; do

cat $f >> ~/textx.txt ;

done

