

Липецкий государственный технический университет

Факультет автоматизации и информатики
Кафедра автоматизированных систем управления

Отчет по лабораторной работе № 5
по дисциплине «OS Linux»
на тему «Программирование на SHELL. Использование командных файлов»

Студент

Группа АС-18-1

Руководитель

К.Н.

учёная степень, учёное звание

подпись, дата

подпись, дата

Сухоруков К.О.

фамилия, инициалы

Кургасов В.В.

фамилия, инициалы

Липецк 2020 г.

СОДЕРЖАНИЕ

Цель работы	2
1 Ход работы	3
Вывод	11
Контрольные вопросы	12

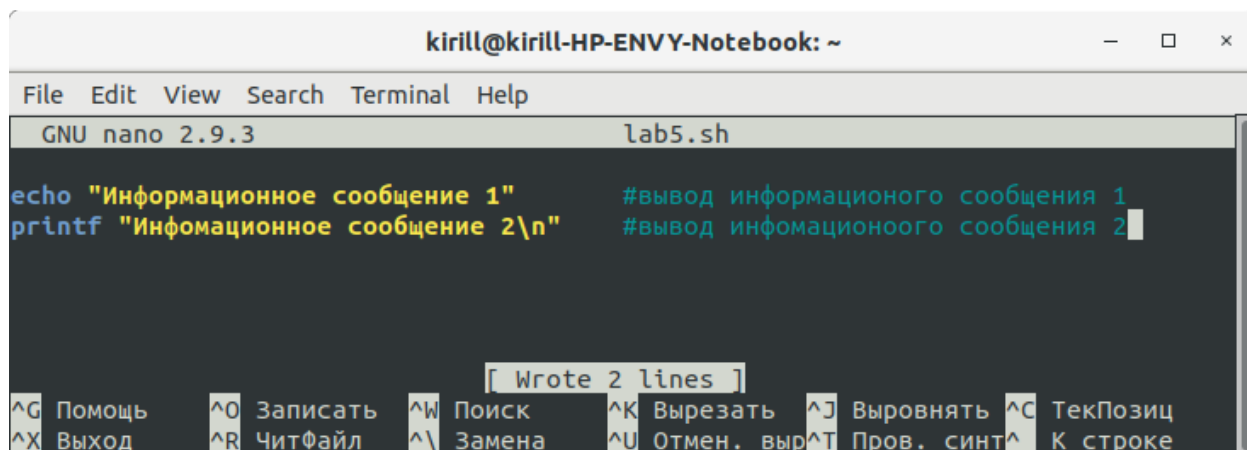
Цель работы

Изучение основных возможностей языка программирования Shell с целью автоматизации процесса администрирования системы за счет написания и использования командных файлов.

1 Ход работы

1. Используя команды ECHO, PRINTF вывести информационные сообщения на экран

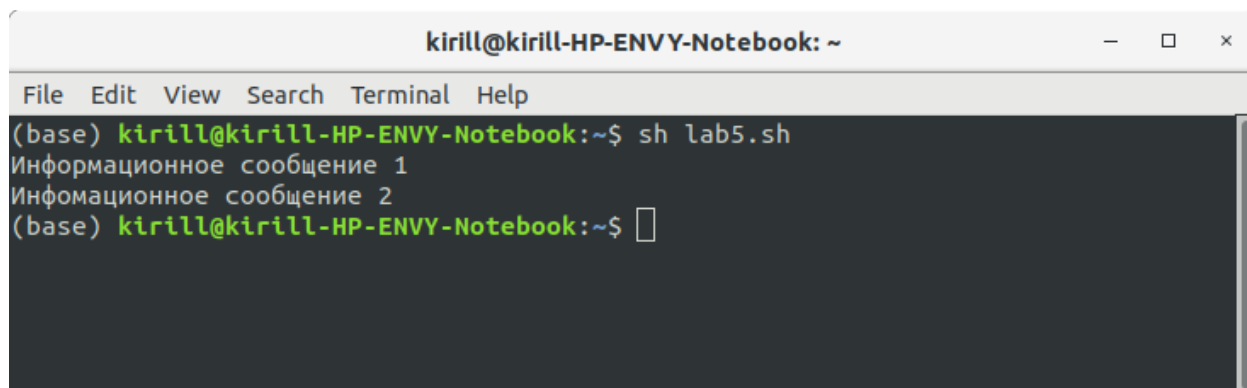
Текст скрипта с комментариями последовательности действий представлен на рисунке 1.1.



```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
GNU nano 2.9.3 lab5.sh  
echo "Информационное сообщение 1"      #вывод информационного сообщения 1  
printf "Информационное сообщение 2\\n"  #вывод инфомационного сообщения 2  
[ Wrote 2 lines ]  
^G Помощь  ^O Записать  ^W Поиск    ^K Вырезать  ^J Выводить  ^C ТекПозиц  
^X Выход    ^R ЧитФайл  ^\ Замена   ^U Отмен. выр ^T Пров. синт ^_ К строке
```

Рисунок 1.1 – Текст скрипта для задания 1

В результате запуска скрипта с помощью команды sh, получим результат, представленный на рисунке 1.2.

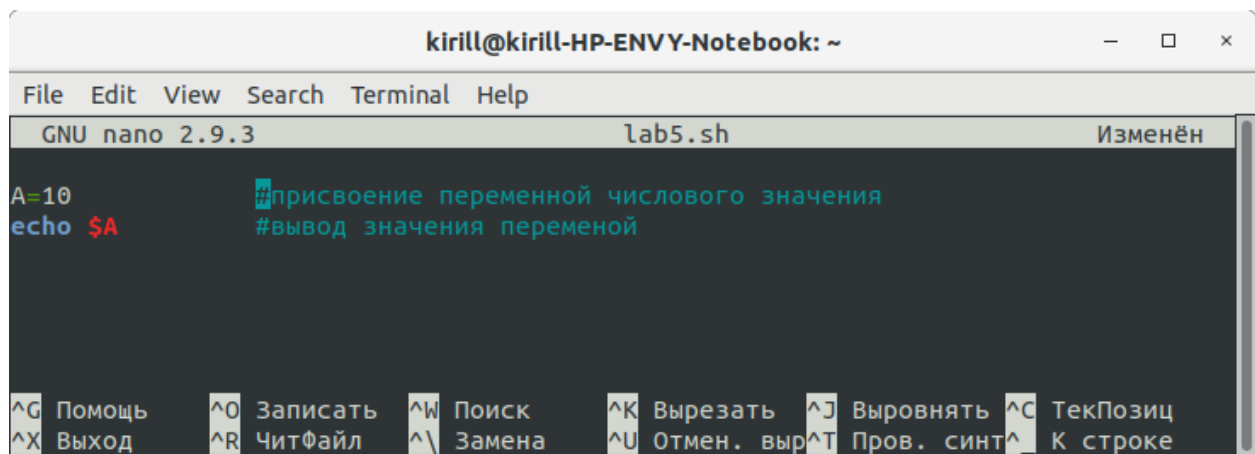


```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ sh lab5.sh  
Информационное сообщение 1  
Информационное сообщение 2  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.2 – Результат выполнения скрипта для задания 1

2. Присвоить переменной А целочисленное значение. Просмотреть значение переменной А

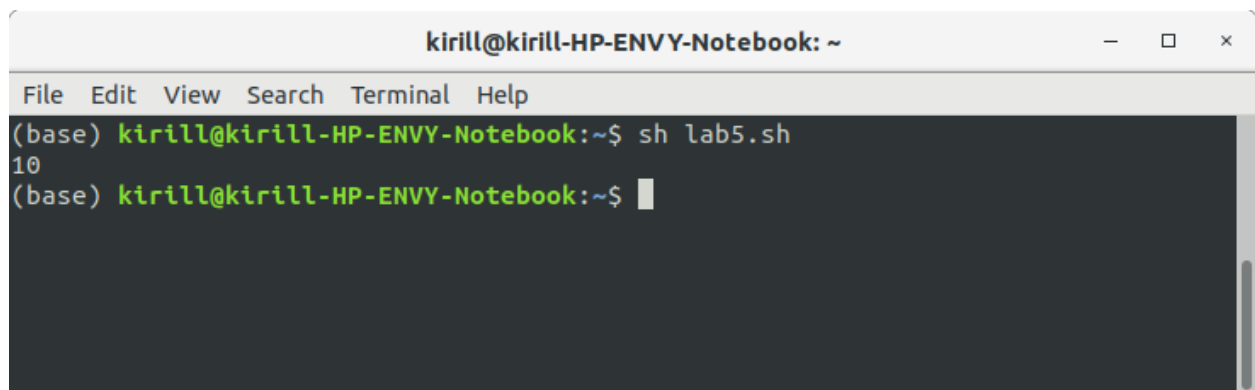
Текст скрипта с комментариями последовательности действий представлен на рисунке 1.3.



The screenshot shows a terminal window titled "kirill@kirill-HP-ENVY-Notebook: ~". Inside, the GNU nano 2.9.3 editor is open with a file named "lab5.sh". The script contains two lines: "A=10" with a comment "#присвоение переменной числового значения" and "echo \$A" with a comment "#вывод значения переменных". The bottom status bar shows various keyboard shortcuts for nano editor functions like "Помощь", "Записать", "Поиск", etc.

Рисунок 1.3 – Текст скрипта для задания 2

В результате запуска скрипта с помощью команды sh, получим результат, представленный на рисунке 1.4.

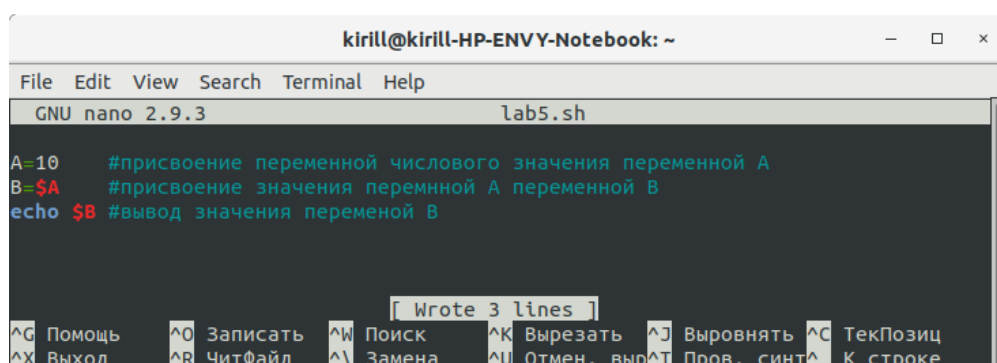


The screenshot shows a terminal window titled "kirill@kirill-HP-ENVY-Notebook: ~". The user has entered the command "sh lab5.sh" at the prompt. The output of the script is "10", which is displayed on the line immediately following the command. The prompt then returns to the shell.

Рисунок 1.4 – Результат выполнения скрипта для задания 2

3. Присвоить переменной В значение переменной А. Просмотреть значение переменной В

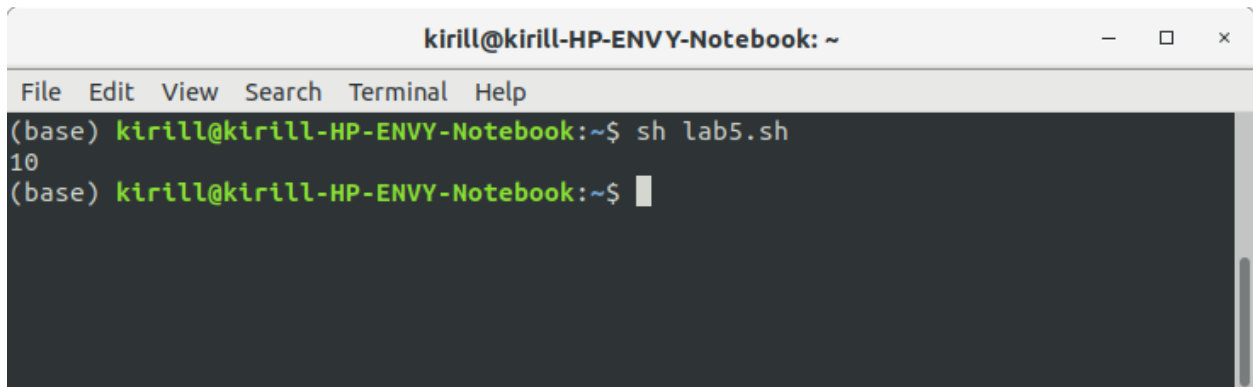
Текст скрипта с комментариями последовательности действий представлен на рисунке 1.5.



The screenshot shows a terminal window titled "kirill@kirill-HP-ENVY-Notebook: ~". Inside, the GNU nano 2.9.3 editor is open with a file named "lab5.sh". The script contains three lines: "A=10" with comment "#присвоение переменной числового значения переменной А", "B=\$A" with comment "#присвоение значения переменной А переменной В", and "echo \$B" with comment "#вывод значения переменной В". A status bar at the bottom indicates "Wrote 3 lines".

Рисунок 1.5 – Текст скрипта для задания 3

В результате запуска скрипта с помощью команды `sh`, получим результат, представленный на рисунке 1.6.

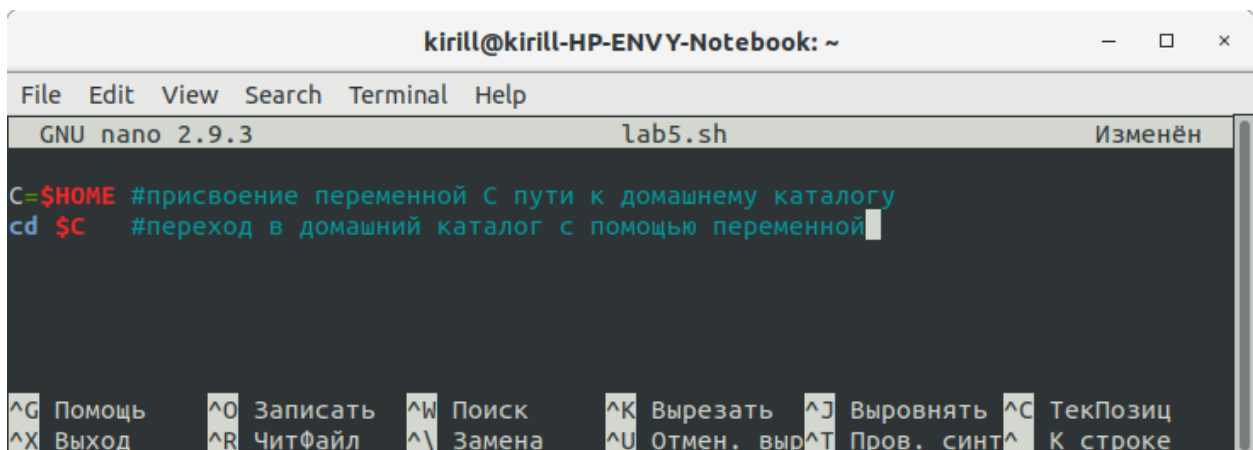


```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ sh lab5.sh  
10  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.6 – Результат выполнения скрипта для задания 3

4. Присвоить переменной `C` значение "путь до своего каталога". Перейти в этот каталог с использованием переменной.

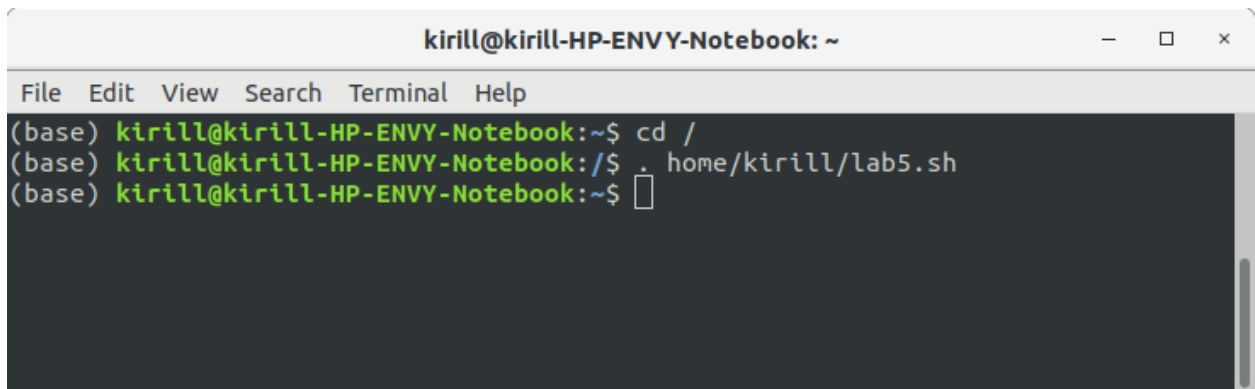
Текст скрипта с комментариями последовательности действий представлен на рисунке 1.7.



```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
GNU nano 2.9.3 lab5.sh Изменён  
C=$HOME #присвоение переменной C пути к домашнему каталогу  
cd $C #переход в домашний каталог с помощью переменной  
  
^G Помощь ^O Записать ^W Поиск ^K Вырезать ^J Выводить ^C ТекПозиц  
^X Выход ^R ЧитФайл ^\ Замена ^U Отмен. выр ^T Пров. синт ^_ К строке
```

Рисунок 1.7 – Текст скрипта для задания 4

Перейдем в корневой каталог с помощью команды `cd /`, затем запустим скрипт с помощью команды `.`, чтобы процесс скрипта запустился в текущем экземпляре `bash`. В результате запуска скрипта с помощью команды `.`, получим результат, представленный на рисунке 1.8.

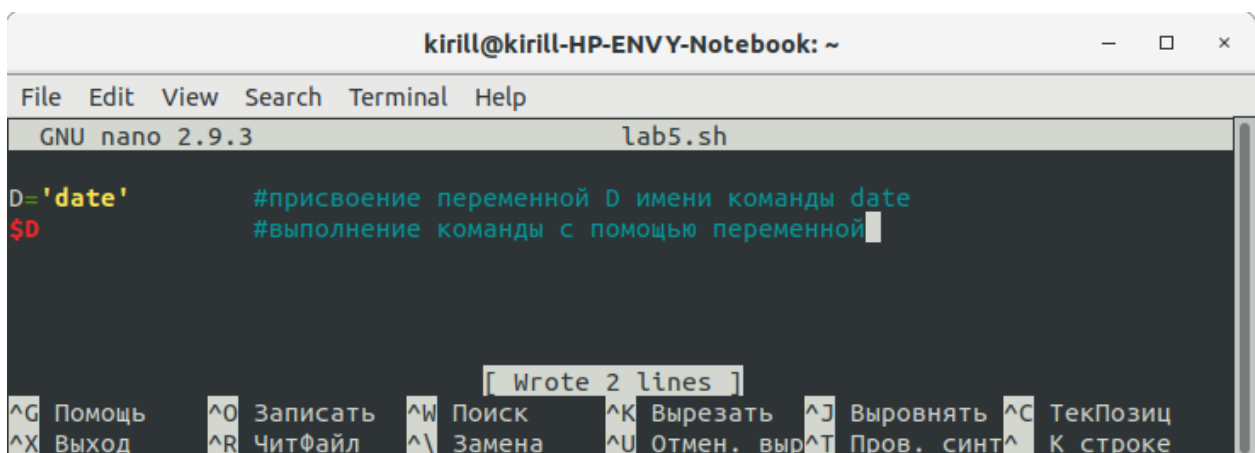


```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ cd /  
(base) kirill@kirill-HP-ENVY-Notebook:/$ . home/kirill/lab5.sh  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.8 – Результат выполнения скрипта для задания 4

5. Присвоить переменной D значение “имя команды”, а именно, команды DATE. Выполнить эту команду, используя значение переменной

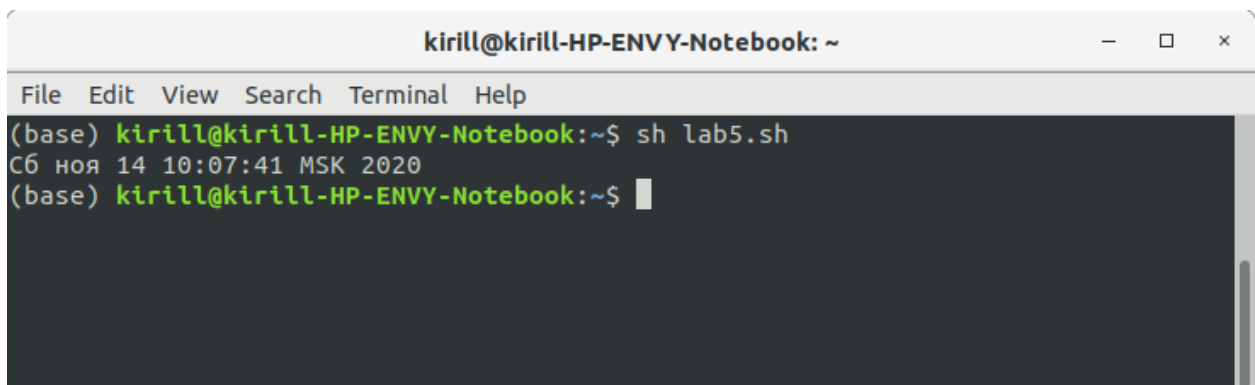
Текст скрипта с комментариями последовательности действий представлен на рисунке 1.9.



```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
GNU nano 2.9.3 lab5.sh  
D='date'           #присвоение переменной D имени команды date  
$D                #выполнение команды с помощью переменной  
[ Wrote 2 lines ]  
^G Помощь  ^O Записать  ^W Поиск    ^K Вырезать  ^J Выводить  ^C ТекПозиц  
^X Выход    ^R ЧитФайл  ^\ Замена  ^U Отмен. выр ^T Пров. синт ^_ К строке
```

Рисунок 1.9 – Текст скрипта для задания 3

В результате запуска скрипта с помощью команды sh, получим результат, представленный на рисунке 1.10.

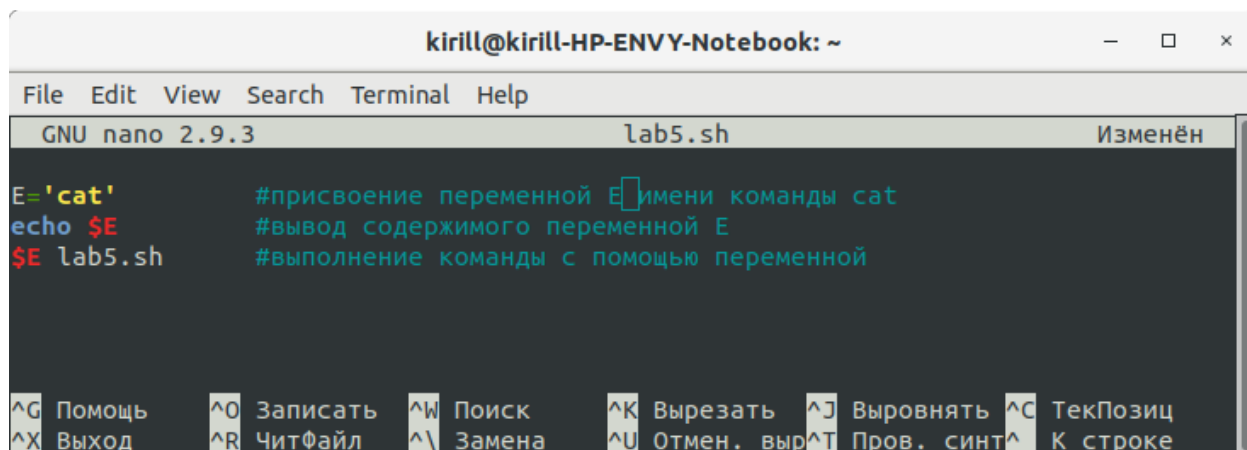


```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ sh lab5.sh  
Сб ноя 14 10:07:41 MSK 2020  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.10 – Результат выполнения скрипта для задания 5

6. Присвоить переменной E значение “имя команды”, а именно, команды просмотра содержимого файла, просмотреть содержимое переменной. Выполнить эту команду, используя значение переменной

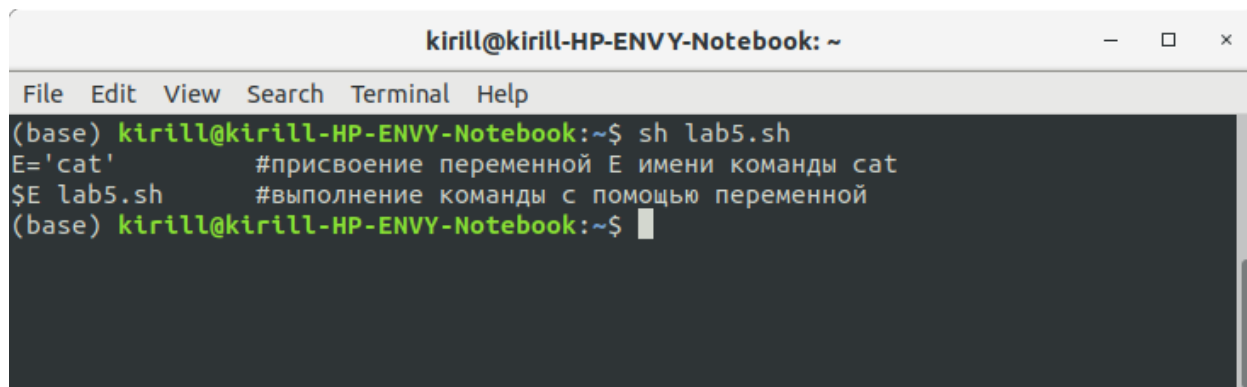
Текст скрипта с комментариями последовательности действий представлен на рисунке 1.11.



```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 lab5.sh Изменён
E='cat'          #присвоение переменной E имени команды cat
echo $E          #вывод содержимого переменной E
$E lab5.sh       #выполнение команды с помощью переменной
^G Помощь      ^O Записать   ^W Поиск      ^K Вырезать   ^J Выводить   ^C ТекПозиц
^X Выход       ^R ЧитФайл   ^\ Замена     ^U Отмен. выр ^T Пров. синт ^_ К строке
```

Рисунок 1.11 – Текст скрипта для задания 6

В результате запуска скрипта с помощью команды sh, получим результат, представленный на рисунке 1.12.

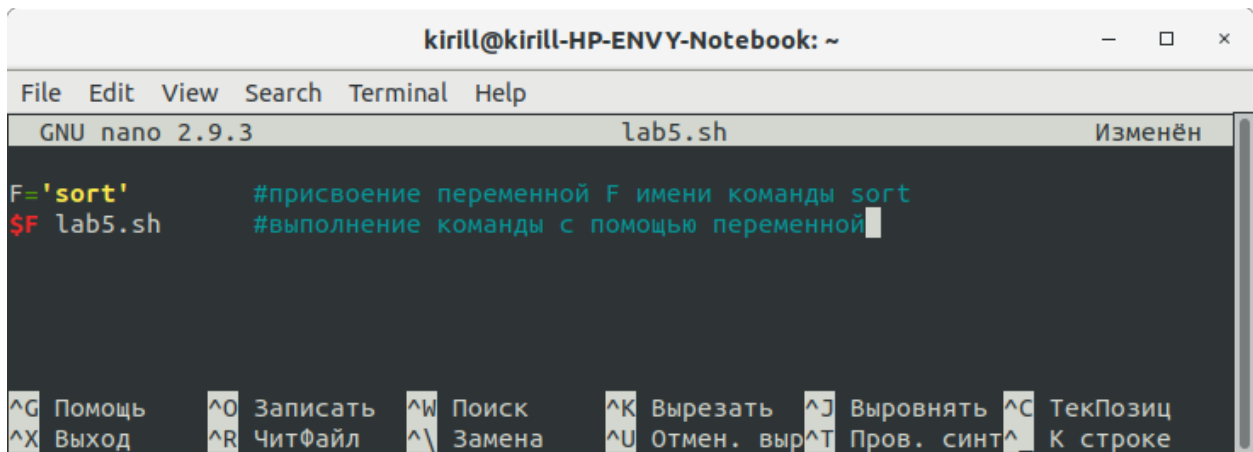


```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
(base) kirill@kirill-HP-ENVY-Notebook:~$ sh lab5.sh
E='cat'          #присвоение переменной E имени команды cat
$E lab5.sh       #выполнение команды с помощью переменной
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.12 – Результат выполнения скрипта для задания 6

7. Присвоить переменной F значение “имя команды”, а именно сортировки содержимого текстового файла. Выполнить эту команду, используя значение переменной

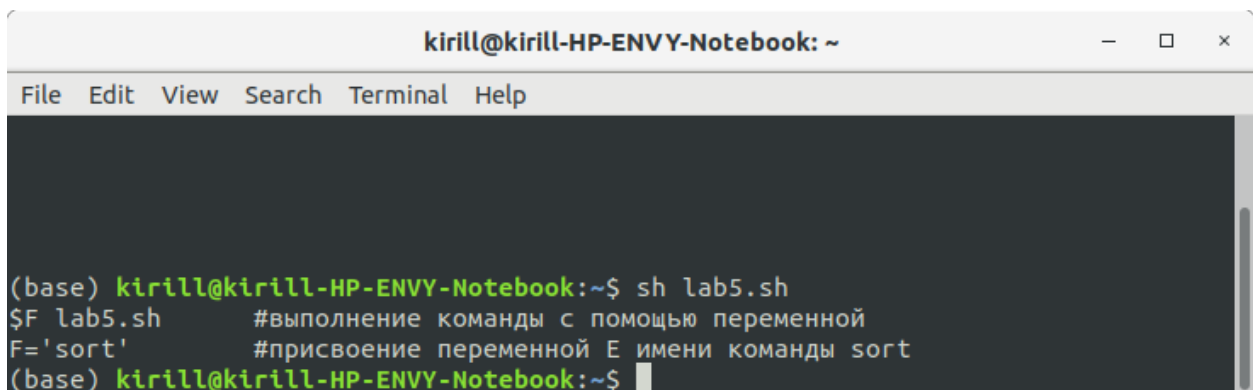
Текст скрипта с комментариями последовательности действий представлен на рисунке 1.13.



```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 lab5.sh Изменён
F='sort'      #присвоение переменной F имени команды sort
$F lab5.sh    #выполнение команды с помощью переменной
^G Помощь    ^O Записать  ^W Поиск     ^K Вырезать  ^J Выводить  ^C ТекПозиц
^X Выход     ^R ЧитФайл  ^\ Замена    ^U Отмен. выр ^T Пров. синт ^_ К строке
```

Рисунок 1.13 – Текст скрипта для задания 7

В результате запуска скрипта с помощью команды sh, получим результат, представленный на рисунке 1.14.

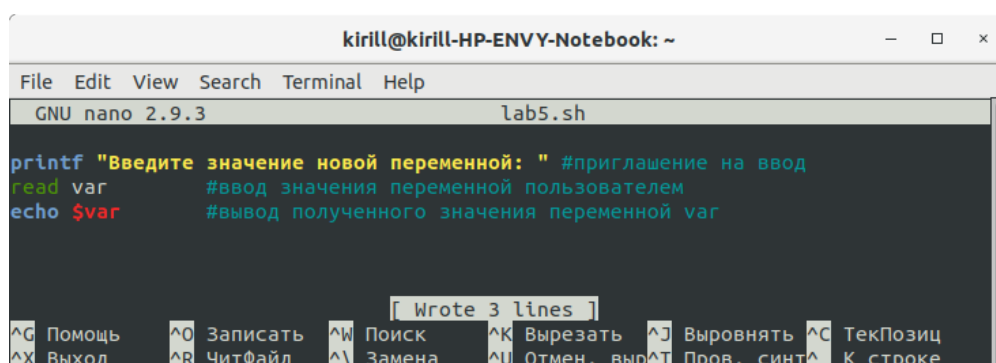


```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
(base) kirill@kirill-HP-ENVY-Notebook:~$ sh lab5.sh
$F lab5.sh      #выполнение команды с помощью переменной
F='sort'        #присвоение переменной E имени команды sort
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.14 – Результат выполнения скрипта для задания 7

8. Программа запрашивает значение переменной, а затем выводит значение этой переменной

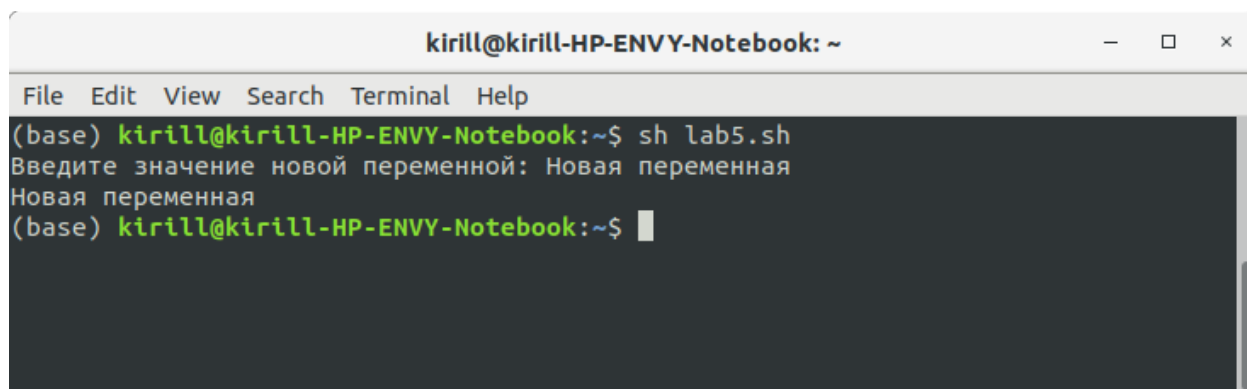
Текст скрипта с комментариями последовательности действий представлен на рисунке 1.15.



```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 lab5.sh
printf "Введите значение новой переменной: " #приглашение на ввод
read var                                     #ввод значения переменной пользователем
echo $var                                    #вывод полученного значения переменной var
[ Wrote 3 lines ]
^G Помощь    ^O Записать  ^W Поиск     ^K Вырезать  ^J Выводить  ^C ТекПозиц
^X Выход     ^R ЧитФайл  ^\ Замена    ^U Отмен. выр ^T Пров. синт ^_ К строке
```

Рисунок 1.15 – Текст скрипта для задания 8

В результате запуска скрипта с помощью команды `sh`, получим результат, представленный на рисунке 1.16.

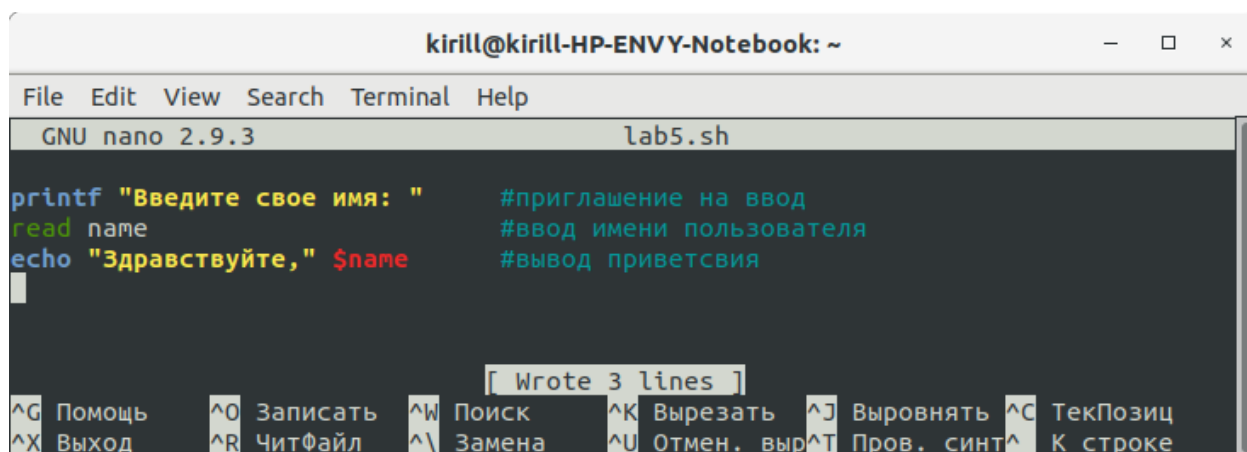


```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ sh lab5.sh  
Введите значение новой переменной: Новая переменная  
Новая переменная  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.16 – Результат выполнения скрипта для задания 8

9. Программа запрашивает имя пользователя, затем здоровается с ним, используя значение введенной переменной

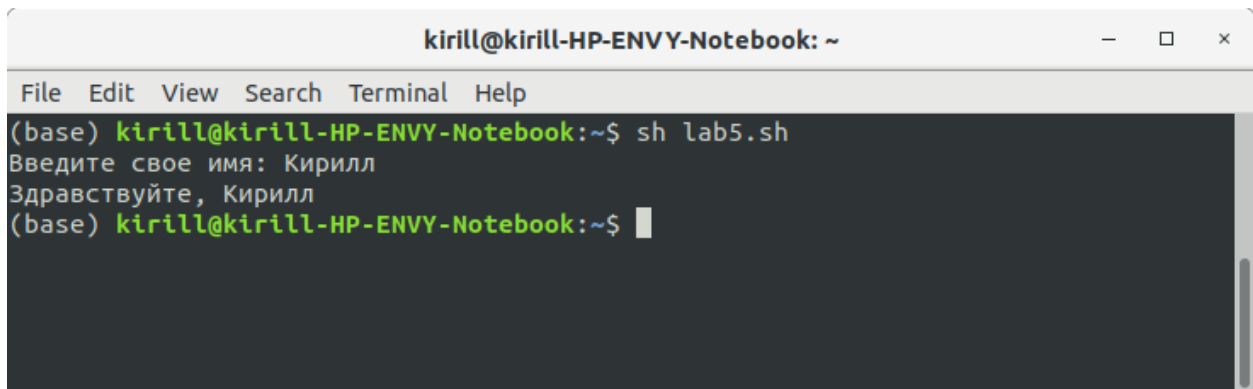
Текст скрипта с комментариями последовательности действий представлен на рисунке 1.17.



```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
GNU nano 2.9.3 lab5.sh  
printf "Введите свое имя: "      #приглашение на ввод  
read name                       #ввод имени пользователя  
echo "Здравствуйте," $name      #вывод приветствия  
[ Wrote 3 lines ]  
^G Помощь  ^O Записать  ^W Поиск    ^K Вырезать  ^J Выводить  ^C ТекПозиц  
^X Выход    ^R ЧитФайл  ^\ Замена  ^U Отмен. выр ^T Пров. синт ^_ К строке
```

Рисунок 1.17 – Текст скрипта для задания 9

В результате запуска скрипта с помощью команды `sh`, получим результат, представленный на рисунке 1.18.

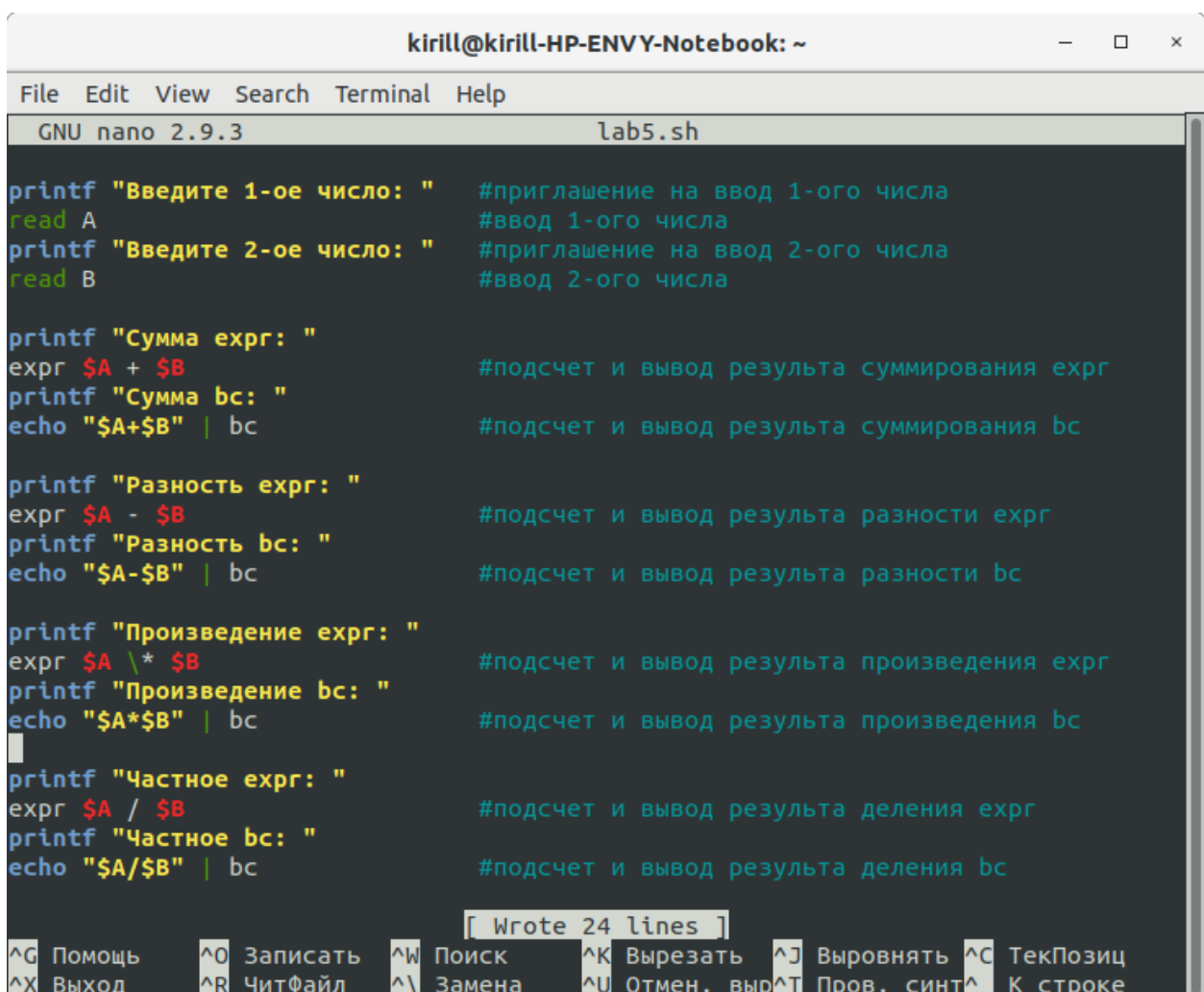


```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
(base) kirill@kirill-HP-ENVY-Notebook:~$ sh lab5.sh
Введите свое имя: Кирилл
Здравствуйте, Кирилл
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.18 – Результат выполнения скрипта для задания 9

10. Программа запрашивает значения двух переменных, вычисляет сумму(разность, произведение, деление) этих переменных. Результат выводится на экран (использовать команды а) EXPR; б) BC)

Текст скрипта с комментариями последовательности действий представлен на рисунке 1.19.



```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 lab5.sh

printf "Введите 1-ое число: " #приглашение на ввод 1-ого числа
read A #ввод 1-ого числа
printf "Введите 2-ое число: " #приглашение на ввод 2-ого числа
read B #ввод 2-ого числа

printf "Сумма expr: "
expr $A + $B #подсчет и вывод результата суммирования expr
printf "Сумма bc: "
echo "$A+$B" | bc #подсчет и вывод результата суммирования bc

printf "Разность expr: "
expr $A - $B #подсчет и вывод результата разности expr
printf "Разность bc: "
echo "$A-$B" | bc #подсчет и вывод результата разности bc

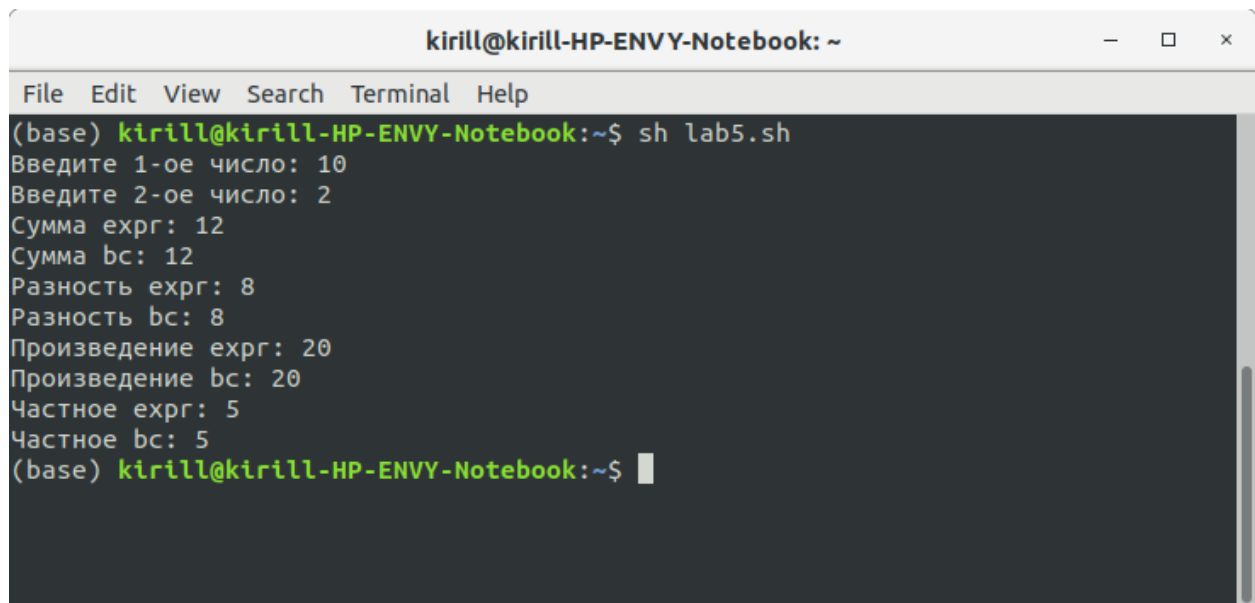
printf "Произведение expr: "
expr $A * $B #подсчет и вывод результата произведения expr
printf "Произведение bc: "
echo "$A*$B" | bc #подсчет и вывод результата произведения bc

printf "Частное expr: "
expr $A / $B #подсчет и вывод результата деления expr
printf "Частное bc: "
echo "$A/$B" | bc #подсчет и вывод результата деления bc

[ Wrote 24 lines ]
^G Помощь ^O Записать ^W Поиск ^K Вырезать ^J Выводить ^C ТекПозиц
^X Выход ^R ЧитФайл ^\ Замена ^U Отмен. выр ^T Пров. синт ^_ К строке
```

Рисунок 1.19 – Текст скрипта для задания 10

В результате запуска скрипта с помощью команды `sh`, получим результат, представленный на рисунке 1.20.

A screenshot of a terminal window titled "kirill@kirill-HP-ENVY-Notebook: ~". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal content shows a user running a script with the command `sh lab5.sh`. The script prompts for two numbers: "Введите 1-ое число: 10" and "Введите 2-ое число: 2". It then calculates and displays: "Сумма exrg: 12", "Сумма bc: 12", "Разность exrg: 8", "Разность bc: 8", "Произведение exrg: 20", "Произведение bc: 20", "Частное exrg: 5", and "Частное bc: 5". The prompt returns to `(base) kirill@kirill-HP-ENVY-Notebook:~$` with a cursor.

```
(base) kirill@kirill-HP-ENVY-Notebook:~$ sh lab5.sh
Введите 1-ое число: 10
Введите 2-ое число: 2
Сумма exrg: 12
Сумма bc: 12
Разность exrg: 8
Разность bc: 8
Произведение exrg: 20
Произведение bc: 20
Частное exrg: 5
Частное bc: 5
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.20 – Результат выполнения скрипта для задания 10

Вывод

В ходе выполнения лабораторной работы я ознакомился на практике с понятием процесса в операционной системе, приобрел опыт и навыки управления процессами в операционной системе Linux.

Контрольные вопросы

1. Перечислите состояния задачи в ОС Ubuntu

В ОС Ubuntu задачи могут находиться в следующих состояниях:

- Running (работа) - процесс работает (он является текущим процессом в системе) или готов к работе (ждет выделения ресурсов процессора);
- Sleeping (спячка) - в это состояние процесс переходит при блокировке;
- Stopped (остановка) - в этом состоянии процесс останавливает работу, обычно после получения соответствующего сигнала. Например, процесс может быть остановлен для отладки;
- Dead (смерть) - в этом состоянии процесс может быть удален из системы;
- Active (активный) и expired (неактивный) - используются при планировании выполнения процесса, и поэтому они не сохраняются в переменной state;
- Zombie (зомби) - процесс мертв, то есть он был остановлен, но в системе осталась выполняемая им задача.

2. Как создаются задачи в ОС Ubuntu?

Задачи создаются путем вызова системной функции `clone`. Любые обращения к `fork` или `vfork` преобразуются в системные вызовы `clone` во время компиляции. Функция `fork` создает дочернюю задачу, виртуальная память для которой выделяется по принципу копирования при записи (copy-on-write). Процедура `vfork` приостанавливает работу родительского процесса в том случае, когда дочерний процесс вызывает функции `execve` или `exit`, чтобы обеспечить загрузку дочерним процессом новых страниц до того, как родительский процесс начнет выполнять бесполезные операции копирования при записи.

3. Назовите классы потоков ОС Ubuntu

В операционной системе Linux алгоритмом диспетчеризации различаются три класса потоков:

- Потоки реального времени, обслуживаемые по алгоритму FIFO;
- Потоки реального времени, обслуживаемые в порядке циклической очереди;
- Потоки разделения времени.

4. Как используется приоритет планирования при запуске задачи

Планировщик различает 40 различных уровней приоритета: от -20 до 19. В соответствии с конвенцией UNIX наименьшее значение означает наибольший приоритет в алгоритме планирования (т.е. -20 - это самый высокий приоритет, который может иметь процесс). Чем меньше значение приоритета, тем больше ресурсов процессора ей выделяется.

5. Как можно изменить приоритет для выполняющейся задачи?

Чтобы изменить приоритет процесса можно воспользоваться командой `renice -n <новый приоритет> <PID процесса>`.