

Лабораторная работа №9

Отчет

Бондарь Алексей Олегович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Задание 1:	6
2.2	Задание 2:	15
3	Контрольные вопросы	29

List of Tables

List of Figures

[illegible]

1 Цель работы

Познакомиться с операционной системой Linux. Получить практические навыки работы с редактором vi, установленным по умолчанию практически во всех дистрибутивах.

2 Выполнение лабораторной работы

2.1 Задание 1:

1. Создаем в домашней папке каталог `work/os/lab06`, используя команду «`mkdir -p work/os/lab06`», где опция `-p` позволяет создать все каталоги, которые указаны внутри пути.
2. Командой «`cd work/os/lab06`» переходим в созданный каталог.
3. Вызываем `vi` и создаем файл `hello.sh` помощью команды «`vi hello.sh`». (рис. 2.1) (рис. 2.2)

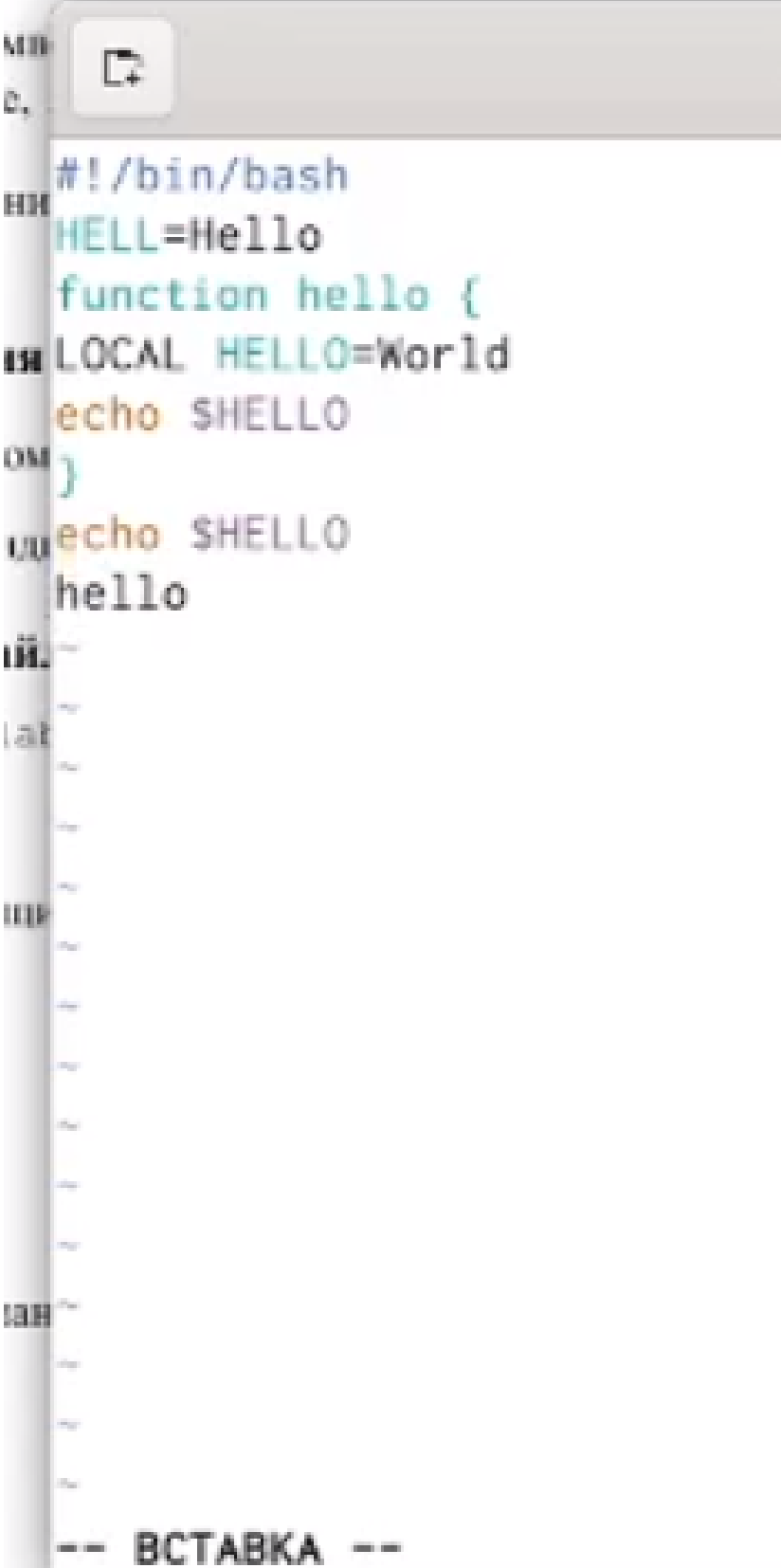
```
aabondarj@dk6n64 ~ $ cd
aabondarj@dk6n64 ~ $ mkdir -p work/os/lab09
aabondarj@dk6n64 ~ $ cd work/os/lab09
aabondarj@dk6n64 ~/work/os/lab09 $ vi hello.sh
```

Figure 2.1: 1



Figure 2.2: 2

4. Нажимаем клавишу «i» и вводим текст, указанный в лабораторной работе.(рис. 2.3)

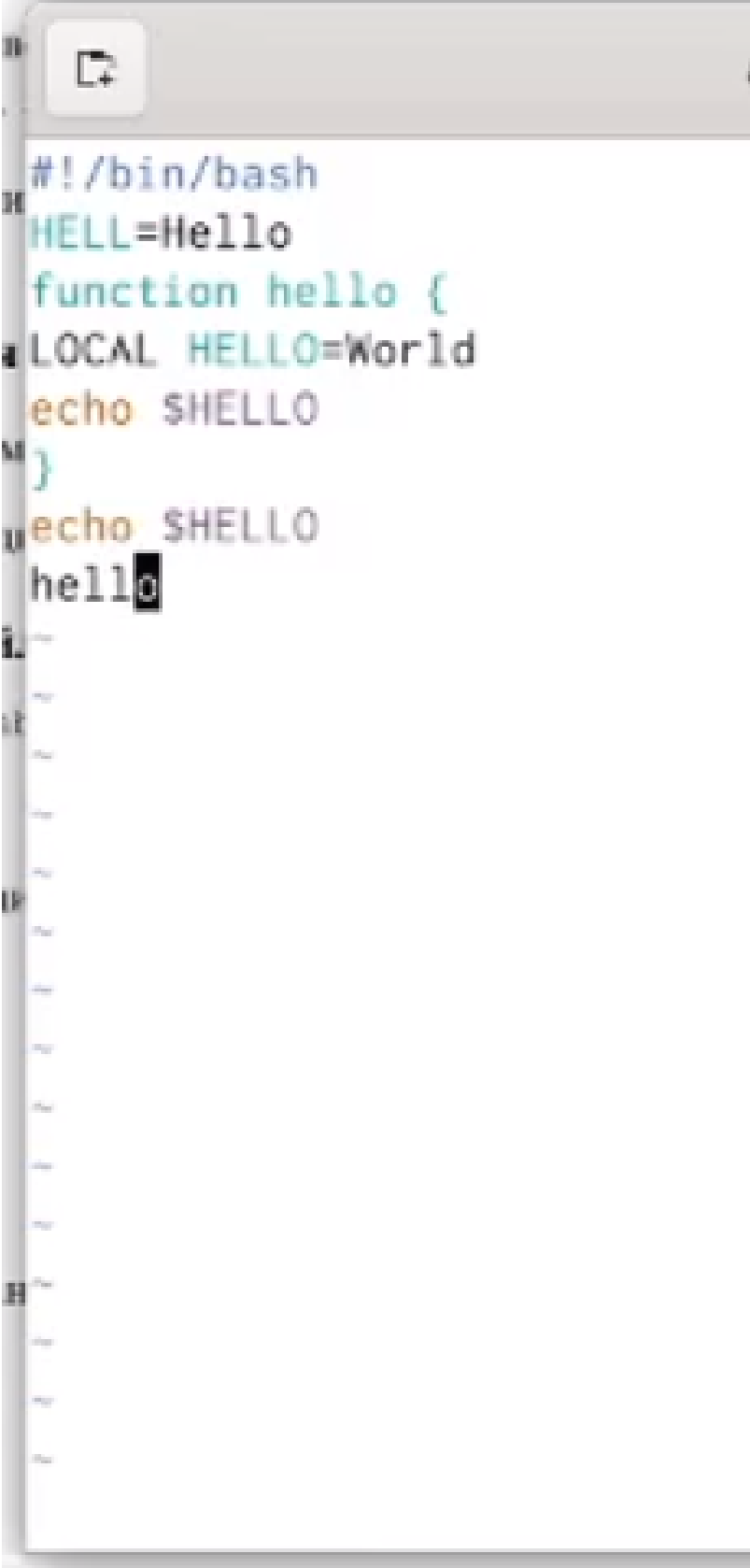


```
#!/bin/bash
HELL=Hello
function hello {
LOCAL HELLO=World
echo $HELLO
}
echo $HELLO
hello

-- BCTABKA --
```

Figure 2.3: 3

5. Нажимаем клавишу «esc», чтобы перейти в командный режим после завершения ввода текста.(рис. 2.4)

A terminal window with a light gray title bar and a standard macOS-style window control button (a square with a right-pointing arrow) in the top-left corner. The terminal displays a bash script with the following content:

```
#!/bin/bash
HELL=Hello
function hello {
LOCAL HELLO=World
echo $HELLO
}
echo $HELLO
hello
```

 The text is color-coded: the shebang is blue, variable assignments and function definitions are green, and the echo commands and the function call are orange. The cursor is positioned at the end of the 'hello' command on the sixth line. The background of the terminal is white, and the text is in a monospaced font.

Figure 2.4: 4

6. Нажимаем «:» для перехода в режим последней строки и внизу экрана видим, что появилось приглашение в виде двоеточия.(рис. 2.5)



```
#!/bin/bash
HELL=Hello
function hello {
LOCAL HELLO=World
echo $HELLO
}
echo $HELLO
hello
```

Figure 2.5: 5

7. Нажимаем «w» (записать) и «q» (выйти), далее нажимаем клавишу «enter» для сохранения текста и завершения работы.(рис. 2.6)

A terminal window with a light gray title bar and a vertical scrollbar on the left. The terminal text is as follows:

```
#!/bin/bash
HELL=Hello
function hello {
LOCAL HELLO=World
echo SHELL
}
echo SHELL
hello
```

The text is color-coded: the shebang is blue, variable assignments are green, function definitions are teal, and the 'echo' command is orange. The prompt at the bottom is ':wq' followed by a black cursor block.

```
:wq
```

Figure 2.6: 6

8. Чтобы сделать файл исполняемым, используем команду «chmod+xhello.sh».(рис. 2.7)

```
aabondarj@dk6n64 ~/work/os/lab09 $ chmod +x hello.sh  
aabondarj@dk6n64 ~/work/os/lab09 $
```

Figure 2.7: 7

2.2 Задание 2:

1. Вызовемviдля редактированияфайлас помощью команды «vi~/work/os/lab06/hello.sh».(рис. 2.8) и (рис. 2.9)

```
aabondarj@dk6n64 ~/work/os/lab09 $ vi ~/work/os/lab09/hello.sh
```

Figure 2.8: 8

2. Далее с помощью стрелок установила курсор в конец слова HELL второй строки.(рис. 2.10)

A terminal window with a light gray title bar and a dark gray border. The terminal content is as follows:

```
#!/bin/bash
HELL=Hello
function hello {
LOCAL HELLO=World
echo $HELLO
}
echo $HELLO
hello
```

The cursor is positioned at the end of the first empty line. At the bottom of the terminal, a status bar displays the text: `"hello.sh" 8L, 90C`.

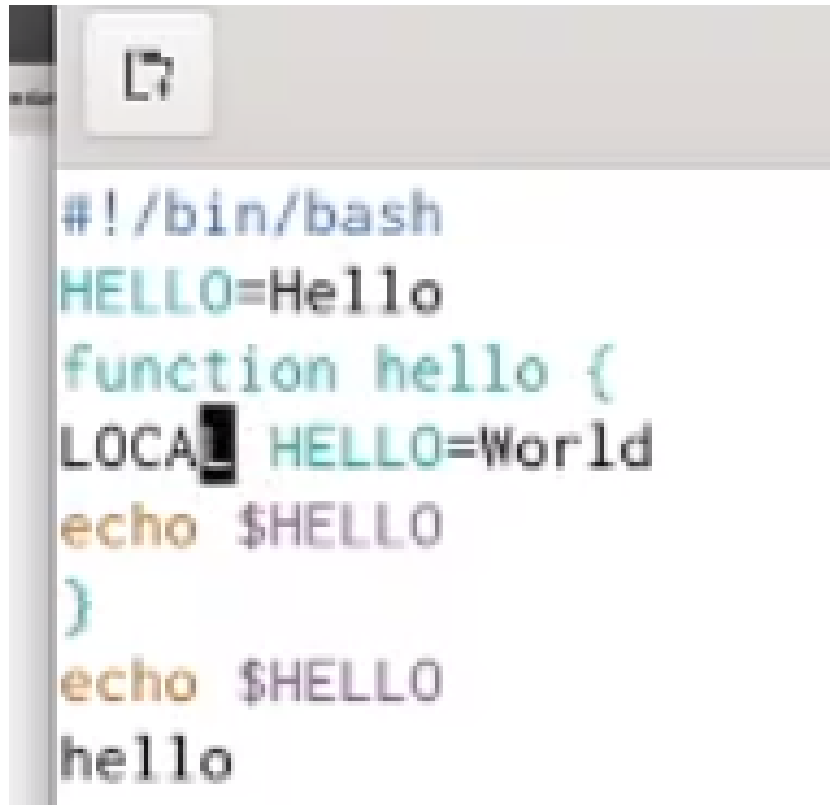
Figure 2.10: 10

3. Перейдем в режим вставки, нажав на клавишу «i», и заменим HELL на HELLO, дописав O. Нажмем «esc» для возврата в командный режим. (рис. 2.11)

```
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
```

Figure 2.11: 11

4. С помощью стрелок установим курсор на четвертую строку и сотрем слово LOCAL с помощью комбинации клавиш «d» (delete) и «w» (word).(рис. 2.12) и (рис. 2.13)



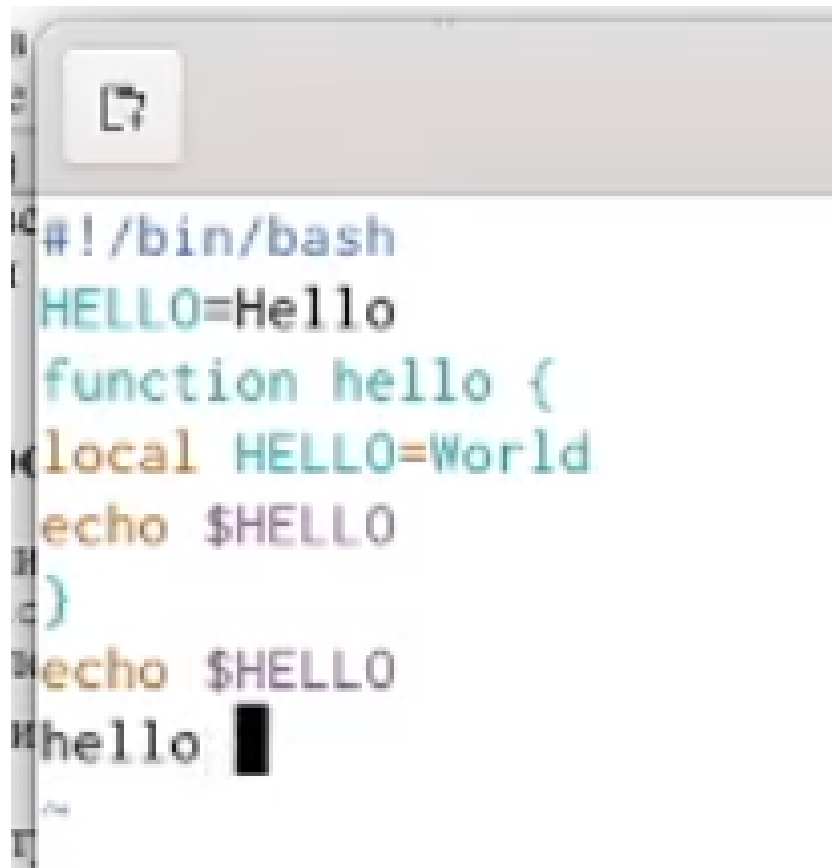
```
#!/bin/bash
HELLO=Hello
function hello {
LOCAL HELLO=World
echo $HELLO
}
echo $HELLO
hello
```

Figure 2.12: 12

```
#!/bin/bash
HELLO=Hello
function hello {
HELLO=World
echo $HELLO
}
echo $HELLO
hello
```

Figure 2.13: 13

5. Перейдем в режим вставки, нажав клавишу «i», и наберем следующий текст:
local. Нажмем «esc» для возврата в командный режим. (рис. 2.14)



```
#!/bin/bash
HELLO=Hello
function hello {
local HELLO=World
echo $HELLO
}
echo $HELLO
hello █
```

Figure 2.14: 14

6. Установим курсор на последней строке файла, используя стрелки. Вставим после неё строку, содержащую следующий текст:echo \$HELLO.(рис. 2.15) и (рис. 2.16)



```
#!/bin/bash
HELLO=Hello
function hello {
    local HELLO=World
    echo $HELLO
}
echo $HELLO
hello
```

Figure 2.15: 15



```
#!/bin/bash
HELLO=Hello
function hello {
    local HELLO=World
    echo $HELLO
}
echo $HELLO
hello
echo $HELLO
```

Figure 2.16: 16

7. Далее нажимаем «esc», чтобы перейти в командный режим.
8. Удаляем последнюю строку, используя комбинацию клавиш «d» и «d». (рис. 2.17)

A terminal window with a title bar containing a window control icon. The terminal displays a bash script being executed. The script defines a variable HELLO, creates a function hello that redefines HELLO and echoes it, and then calls the function. The output shows the variable's value changing from Hello to World.

```
#!/bin/bash
HELLO=Hello
function hello {
    local HELLO=World
    echo $HELLO
}
echo $HELLO
hello
```

Figure 2.17: 17

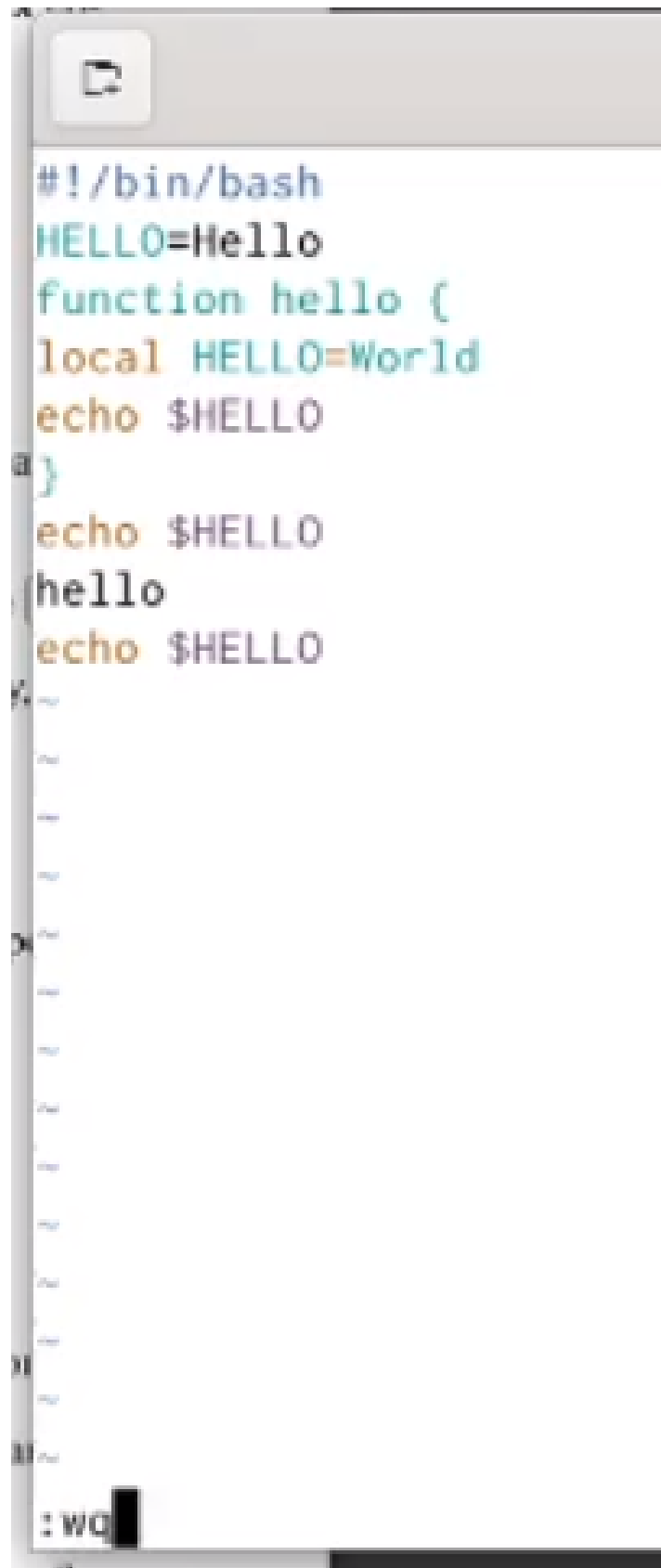
9. Введем команду отмены изменений «u» для отмены последней команды.(рис. 2.18)



```
#!/bin/bash
HELLO=Hello
function hello {
  local HELLO=World
  echo $HELLO
}
echo $HELLO
hello
echo $HELLO
```

Figure 2.18: 18

10. Введем символ «:» для перехода в режим последней строки. Запишем произведённые изменения, нажав «w» и «q» и выйдем из vi. (рис. 2.19)



```
#!/bin/bash
HELLO=Hello
function hello {
  local HELLO=World
  echo $HELLO
}
echo $HELLO
hello
echo $HELLO
```

The image shows a terminal window with a light gray title bar. The terminal content is as follows:

```
#!/bin/bash
HELLO=Hello
function hello {
  local HELLO=World
  echo $HELLO
}
echo $HELLO
hello
echo $HELLO
```

Below the script, there are several lines of output, each preceded by a prompt character (likely '\$'). The first output line is 'Hello', and the subsequent lines are 'World'. The terminal window has a dark gray status bar at the bottom showing ': WQ'.

Figure 2.19: 19

3 Контрольные вопросы

1. Редактор vi имеет три режима работы: командный режим – предназначен для ввода команд редактирования и навигации по редактируемому файлу; режим вставки – предназначен для ввода содержания редактируемого файла; режим последней (или командной) строки – используется для записи изменений в файл и выхода из редактора.
2. Чтобы выйти из редактора, не сохраняя произведённые изменения, нужно в режиме командной строки нажать клавиши «:» «q» «!»
3. Команды позиционирования: «0»(ноль) – переход в начало строки; «\$» – переход в конец строки; «G» – переход в конец файла; n«G» – переход на строку с номером n.
4. При использовании прописных W и B под разделителями понимаются только пробел, табуляция и возврат каретки. При использовании строчных w и b под раздел в ходе выполнения данной лабораторной работы я познакомилась с операционной системой Linux и получила практические навыки работы с редактором vi. ителями понимаются также любые знаки пунктуации.
5. Чтобы из любого места редактируемого файла перейти в начало (конец) файла, нужно в режиме командной строки нажать клавиши «1» «G» («G»).
6. Команды редактирования:
 - Вставка текста «a» – вставить текст после курсора; «A» – вставить текст в конец строки; «i» – вставить текст перед курсором; n«i» – вставить текст праз; «I» – вставить текст в начало строки. Вставка строки «O» – вставить строку под

курсором; «O»–вставить строку над курсором. Удаление текста «x»–удалить один символ в буфер; «d»«w»–удалить одно слово в буфер; «d»«\$»–удалить в буфер текст от курсора до конца строки; «d»«0»–удалить в буфер текст от начала строки до позиции курсора; «d»«d»–удалить в буфер одну строку; n«d»«d»–удалить в буфер nстрок. Отмена и повтор произведённых изменений «u»–отменить последнее изменение; «.»–повторить последнее изменение.

- Копирование текста в буфер «Y»–скопировать строку в буфер; n«Y»–скопировать nстрок в буфер; «y»«w»–скопировать слово в буфер. Вставка текста из буфера «r»–вставить текст из буфера после курсора; «R»–вставить текст из буфера перед курсором. Замена текста «c»«w»–заменить слово; n«c»«w»–заменить nслов; «c»«\$»–заменить текст от курсора до конца строки; «r»–заменить слово; «R»–заменить текст.
 - Поиск текста «/» текст–произвести поиск вперёд по тексту указанной строки символов текст; «?»текст–произвести поиск назад по тексту указанной строки
- В ходе выполнения данной лабораторной работы я познакомилась с операционной системой Linux и получила практические навыки работы с редактором vi.мволов текст.Копирование и перемещение текста

«:» n,m «d» –удалить строки с n по m;

«:» i,j «m» k –переместить строки с i по j, начиная со строки k;

«:» i,j «t» k –копировать строки с i по j в строку k;

«:» i,j «w» имя-файла –записать строки с i по j в файл с именем имя-файла.

7. Чтобы заполнить строку \$, необходимо для начала перейти на эту строку, нажав клавиши n “G”, где n-номер строки , далее нажать “0” для перехода в начало строки. Теперь необходимо нажать “c” \$, чтобы заменить текст от курсора конца строки, и вывести символы “\$”.

8. Чтобы отменить по одному предыдущему действию последовательно, необходимо нажать «u». Чтобы отменить все изменения, произведённые со времени последней записи, нужно нажать «:» «e» «!».

9. Команды редактирования в режиме командной строки

Копирование и перемещение текста «:»n,m «d»–удалить строки с n по m; «:»i,j «m»k –переместить строки с i по j, начиная со строки k; «:»i,j «t»k –копировать строки с i по j в строку k; «:»i,j «w»имя-файла–записать строки с i по j в файл с именем имя-файла. Запись в файл и выход из редактора «:»«w»–записать изменённый текст в файл, не выходя из vi; «:»«w»имя-файла–записать изменённый текст в новый файл с именем имя-файла; «:»«w»«!»имя-файла–записать изменённый текст в файл с именем имя-файла; «:»«w»«q»–записать изменения в файл и выйти из vi; «:»«q»–выйти из редактора vi; «:»«q»«!»–выйти из редактора без записи; «:»«e»«!»–вернуться в командный режим, отменив все изменения, произведённые со времени последней записи. Опции редактора vi позволяют настроить рабочую среду. Для задания опций используется команда set (в режиме последней строки): «:»set all –вывести полный список опций; «:»set nu –вывести номера строк; «:»set list –вывести невидимые символы; «:»set ic –не учитывать при поиске, является ли символ прописным или строчным. Если вы хотите отказаться от использования опции, то в команде set перед именем опции надо поставить no. 10. Чтобы определить, не перемещая курсора, позицию, в которой заканчивается строка, нужно в командном режиме находясь на нужной строке нажать «\$» и посмотреть на число после запятой в правом нижнем углу экран 11. Опции редактора vi позволяют настроить рабочую среду. Для задания опций используется команда set (в режиме командной строки). Если вы хотите отказаться от использования опции, то в команде set перед именем опции надо поставить no. Чтобы просмотреть опции редактора vi, необходимо нажать «:» setall. Нажав «:» help “название_опции”, можно узнать назначение конкретной опции. 12. В режиме командной строки внизу редактора присутствует «:», в режиме ввода – «–ВСТАВКА –», в командном режиме внизу ничего нет. 13. Граф взаимосвязи

режимов работы редактора vi # Вывод В ходе выполнения данной лабораторной работы я познакомился с операционной системой Linux и получил практические навыки работы с редактором vi.