

Отчет по лабораторной работе №8:

Текстовый редактор vi

Федорова Наталия Артемовна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Контрольные вопросы	12
6	Выводы	15

Список иллюстраций

4.1	Создание каталога и файла	8
4.2	Ввод текста	8
4.3	Сохранение текста и завершение работы	9
4.4	Исполняемый файл	9
4.5	Режим редактирования файла	10
4.6	Изменения в файле	10
4.7	Удаление строки	11
4.8	Отмена действия	11

Список таблиц

1 Цель работы

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

2 Задание

1. Создать новый файл с использованием vi.
2. Редактировать существующий файл.

3 Теоретическое введение

В большинстве дистрибутивов Linux в качестве текстового редактора по умолчанию устанавливается интерактивный экранный редактор *vi* (*Visual display editor*).

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.

Для вызова редактора *vi* необходимо указать команду *vi* и имя редактируемого файла:

```
vi <имя_файла>
```

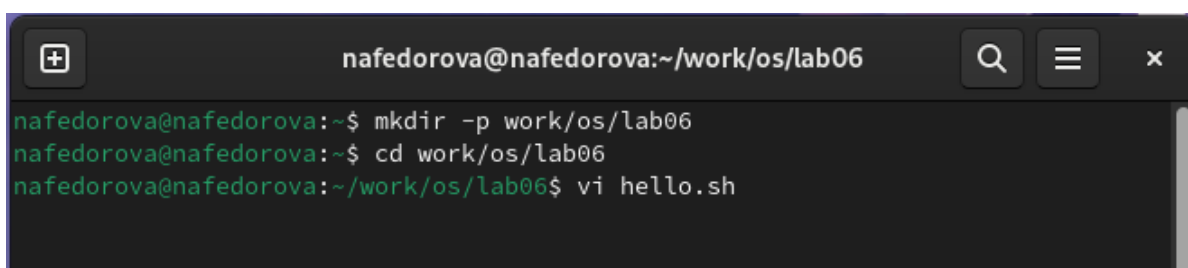
При этом в случае отсутствия файла с указанным именем будет создан такой файл. Переход в командный режим осуществляется нажатием клавиши *Esc*. Для выхода из редактора *vi* необходимо перейти в режим последней строки: находясь в командном режиме, нажать *Shift-;* (по сути символ *:* — двоеточие), затем:

- набрать символы *wq*, если перед выходом из редактора требуется записать изменения в файл;
- набрать символ *q* (или *q!*), если требуется выйти из редактора без сохранения.

Замечание. Следует помнить, что *vi* различает прописные и строчные буквы при наборе (восприятии) команд.

4 Выполнение лабораторной работы

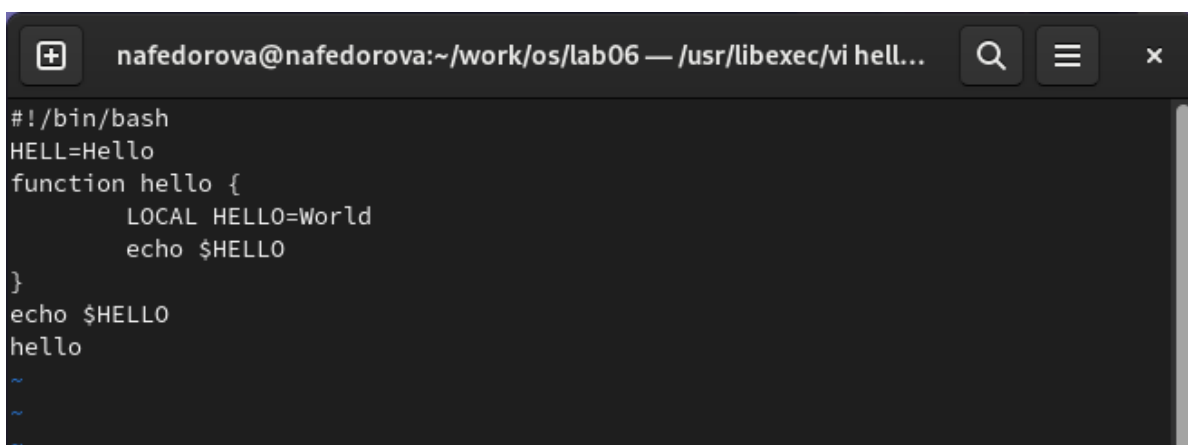
1. Создаю новый каталог с именем `~/work/os/lab06` командой `mkdir -p /work/os/lab06`, а после перехожу в него с помощью `cd`. Вызываю `vi` и создаю файл `hello.sh` командой `vi hello.sh` (рис. 4.1).



```
nafedorova@nafedorova:~/work/os/lab06
nafedorova@nafedorova:~$ mkdir -p work/os/lab06
nafedorova@nafedorova:~$ cd work/os/lab06
nafedorova@nafedorova:~/work/os/lab06$ vi hello.sh
```

Рис. 4.1: Создание каталога и файла

Нажимаю клавишу `ii` и ввожу данный текст (рис. 4.2).



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

Рис. 4.2: Ввод текста

Нажимаю клавишу `Esc` для перехода в командный режим после завершения ввода

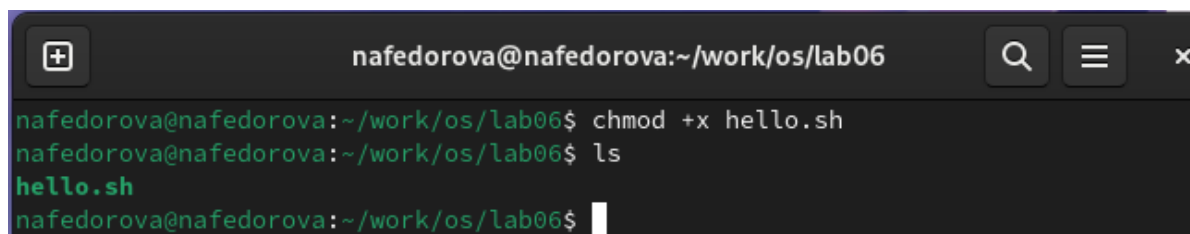
текста, затем нажимаю : для перехода в режим последней строки, нажимаю w (записать) и q (выйти), а затем нажимаю клавишу Enter для сохранения текста и завершения работы (рис. 4.3).



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

Рис. 4.3: Сохранение текста и завершение работы

После делаю файл исполняемым с помощью команды `chmod +x hello.sh` (рис. 4.4).



```
nafedorova@nafedorova:~/work/os/lab06
nafedorova@nafedorova:~/work/os/lab06$ chmod +x hello.sh
nafedorova@nafedorova:~/work/os/lab06$ ls
hello.sh
nafedorova@nafedorova:~/work/os/lab06$
```

Рис. 4.4: Исполняемый файл

2. Вызываю vi на редактирование файла. Для этого использую команду `vi hello.sh` (рис. 4.5).

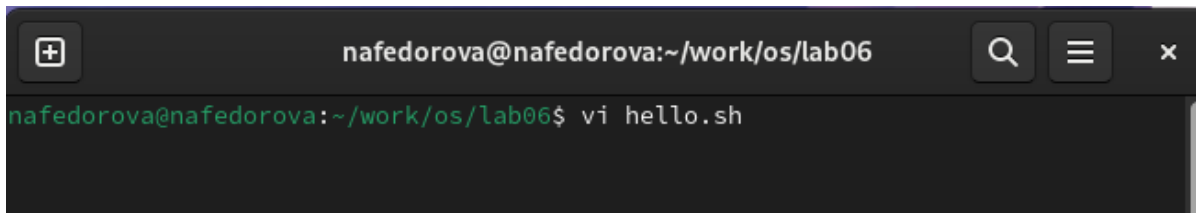


Рис. 4.5: Режим редактирования файла

Устанавливаю курсор в конец слова HELLO второй строки и, перейдя в режим вставки, заменяю на HELLO. После устанавливаю курсор на четвертую строку и стираю слово LOCAL, а вместо него в режиме вставки ввожу local. Также после последней строки файла вставляю echo \$HELLO (рис. 4.6).

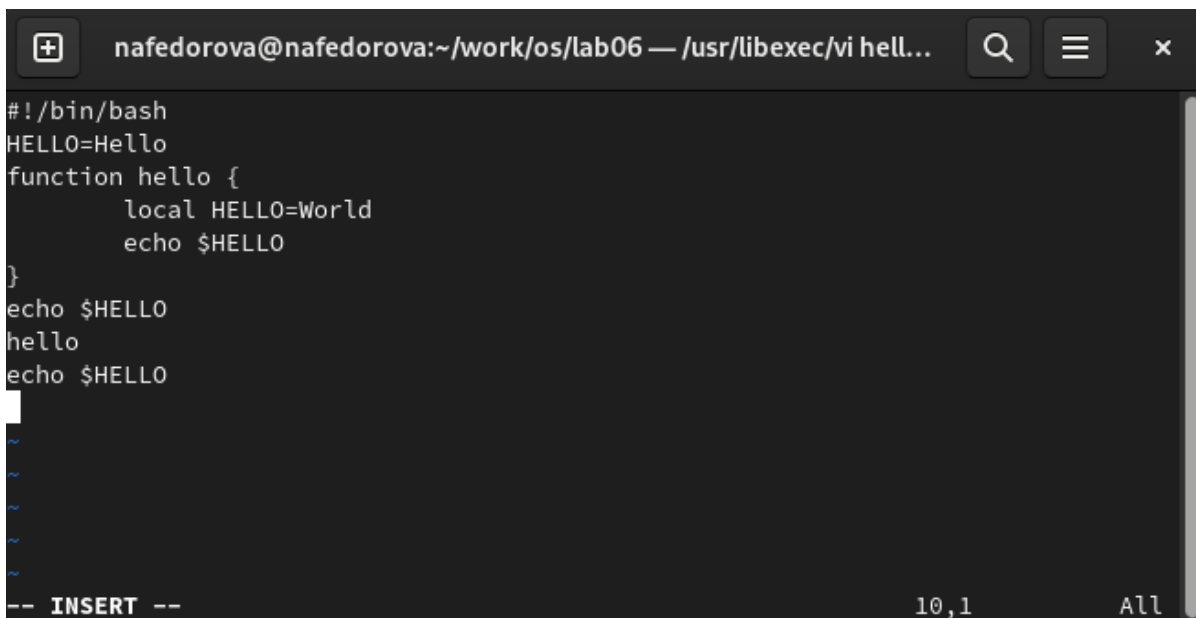
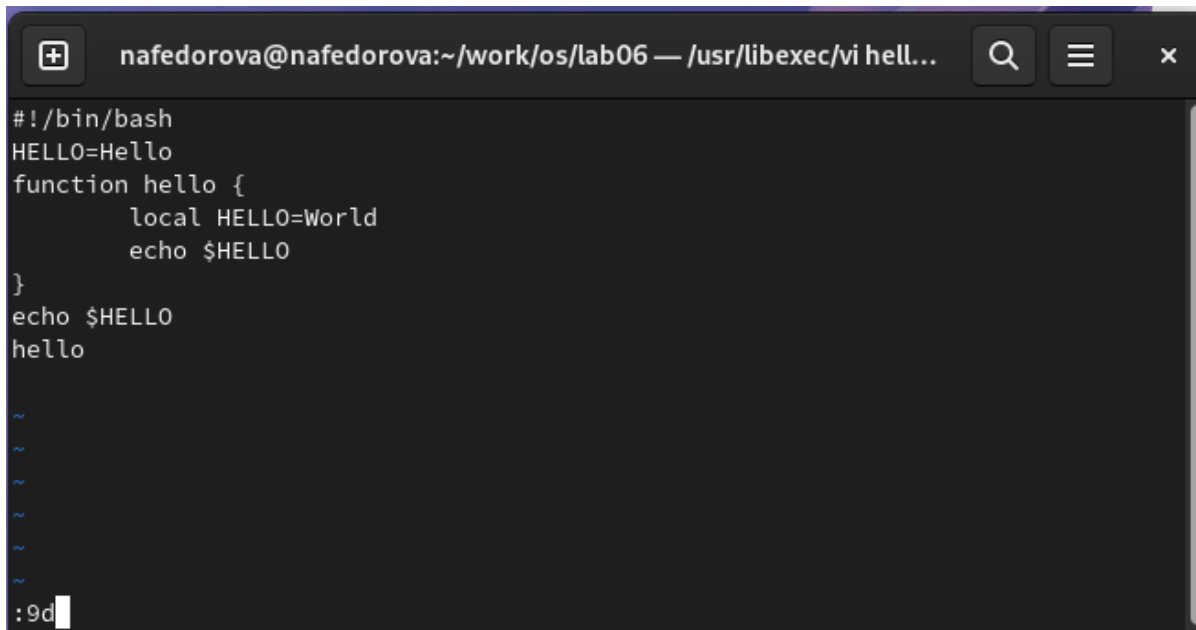


Рис. 4.6: Изменения в файле

Удаляю последнюю строчку в файле, нажав на : и введя 9d(рис. 4.7).



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

~
~
~
~
~
~
~
:9d
```

Рис. 4.7: Удаление строки

В командном режиме нажимаю на клавишу u, чтобы отменить последнее действие (рис. 4.8).



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

~
~
~
~
~
~
~
1 more line; before #4 58 seconds ago 9,1 All
```

Рис. 4.8: Отмена действия

Введя :wq, записываю произведенные изменения и выхожу из vi.

5 Контрольные вопросы

1. Дайте краткую характеристику режимам работы редактора vi.

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. Как выйти из редактора, не сохраняя произведённые изменения?

Чтобы выйти из редактора vi, не сохраняя изменения, нужно находясь в режиме команд нажать :q! и затем клавишу Enter.

3. Назовите и дайте краткую характеристику командам позиционирования.

Команды позиционирования включают:

- h: Сдвигает курсор влево.
- j: Сдвигает курсор вниз.
- k: Сдвигает курсор вверх.
- l: Сдвигает курсор вправо.
- 0 (ноль): Перемещает курсор в начало строки.
- \$: Перемещает курсор в конец строки.

4. Что для редактора vi является словом?

Для редактора vi слово - это последовательность символов, разделенных пробелами или другими символами пробелов.

5. Каким образом из любого места редактируемого файла перейти в начало (конец) файла?

Для перемещения в начало файла используется комбинация клавиш gg, а для перемещения в конец файла - комбинация клавиш G.

6. Назовите и дайте краткую характеристику основным группам команд редактирования.

Основные группы команд редактирования в vi включают:

- **Команды удаления:** Позволяют удалять символы, слова, строки и т. д. Например, x удаляет символ под курсором, dw удаляет слово и так далее.
- **Команды вставки и изменения текста:** Позволяют вставлять новый текст или изменять существующий. Например, i входит в режим вставки перед текущим символом, a входит в режим вставки после текущего символа, c удаляет текст и входит в режим вставки.
- **Команды копирования и вставки:** Позволяют копировать и вставлять текст. Например, yy копирует текущую строку, p вставляет содержимое буфера обмена.
- **Команды поиска и замены:** Позволяют находить и заменять текст в файле. Например, / начинает поиск вперед, :%s/old/new/g заменяет все вхождения "old" на "new" в файле.

7. Необходимо заполнить строку символами \$. Каковы ваши действия?

Чтобы заполнить строку символами \$, можно войти в режим вставки (нажав клавишу i), набрать символ \$ нужное количество раз и затем выйти из режима вставки (нажав клавишу Esc).

8. Как отменить некорректное действие, связанное с процессом редактирования?

Для отмены некорректного действия в редакторе vi используется команда отмены. Для этого нужно нажать клавишу u в режиме команд.

9. Назовите и дайте характеристику основным группам команд режима последней строки.

Основные группы команд режима последней строки включают:

- Сохранение и выход: Например, :w сохраняет файл, :q выходит из редактора, :wq сохраняет и выходит.
- Поиск и замена: Например, :/pattern ищет вперед по шаблону, :%s/old/new/g заменяет все вхождения "old" на "new".
- Другие действия: Например, :e filename открывает другой файл для редактирования, :set позволяет устанавливать опции редактора.

10. Как определить, не перемещая курсора, позицию, в которой заканчивается строка?

Позицию, в которой заканчивается строка, можно определить, не перемещая курсора, используя команду \$. Например, :\$ переместит курсор на последний символ последней строки.

11. Выполните анализ опций редактора vi (сколько их, как узнать их назначение и т.д.).

Опций редактора vi может быть множество, включая опции форматирования, отображения, регистра и другие. Назначение опций можно узнать, посмотрев документацию по vi или выполнив команду :help options.

12. Как определить режим работы редактора vi?

Режим работы редактора vi можно определить по поведению курсора и реакции на ввод клавиш. Например, если курсор перемещается при вводе символов, это обычно означает режим команд, если же текст появляется в месте курсора, то это режим вставки.

6 Выводы

В данной лабораторной работе я познакомилась с операционной системой Linux, а также получила практические навыки работы с редактором vi, установленным по умолчанию практически во всех дистрибутивах.