

Отчет по лабораторной работе №8

Текстовый редактор vi

Кочкарев “sakochkarev” Станислав

Содержание

| | | |
|----------|---------------------------------------|-----------|
| 1 | Цель работы | 3 |
| 2 | Задание | 4 |
| 3 | Выполнение лабораторной работы | 5 |
| 4 | Выводы | 14 |
| 5 | Ответы на контрольные вопросы | 15 |

1 Цель работы

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

2 Задание

- Ознакомление с теоретическим материалом
- Ознакомление с редактором vi
- Выполнение упражнений с редактором vi
 - Создание нового файла с использованием vi
 - Редактирование существующего файла с использованием vi

3 Выполнение лабораторной работы

Первым делом мы ознакомились с приведенным теоретическим материалом. Также перед лабораторной работой я ознакомился с самим редактором `vi`.

Стоит упомянуть, что при выполнении данной лабораторной работы использовался редактор `vim`, отличия которого в рамках данной лабораторной работы несущественны.

После ознакомления со всем материалом я приступил к выполнению основных заданий лабораторной работы.

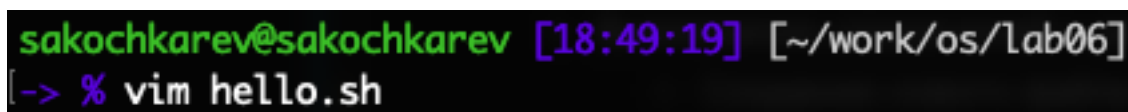
Первым заданием было создание нового файла с использованием редактора. Был создан каталог `~/work/os/lab06` (рис. 3.1).



```
sakochkarev@sakochkarev [18:49:06] [~]  
[-> % mkdir -p ~/work/os/lab06
```

Рис. 3.1: Создание каталога `~/work/os/lab06`

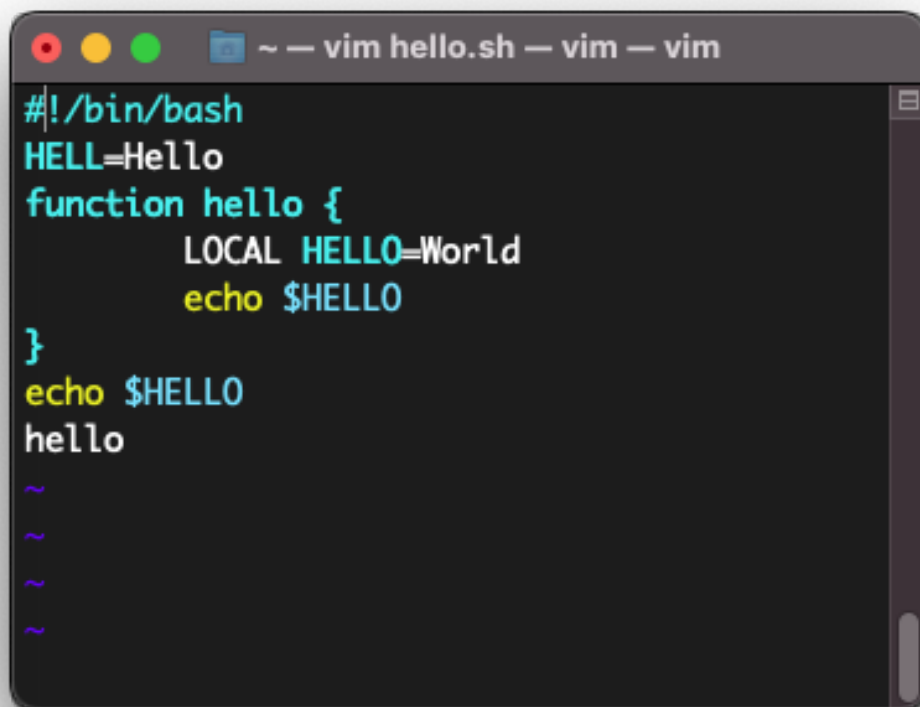
Далее внутри новосозданного каталога была выполнена команда `vim hello.sh` (рис. 3.2), которая создала и открыла в редакторе текстовый файл `hello.sh`.



```
sakochkarev@sakochkarev [18:49:19] [~/work/os/lab06]  
[-> % vim hello.sh
```

Рис. 3.2: Выполнение команды `vim hello.sh`

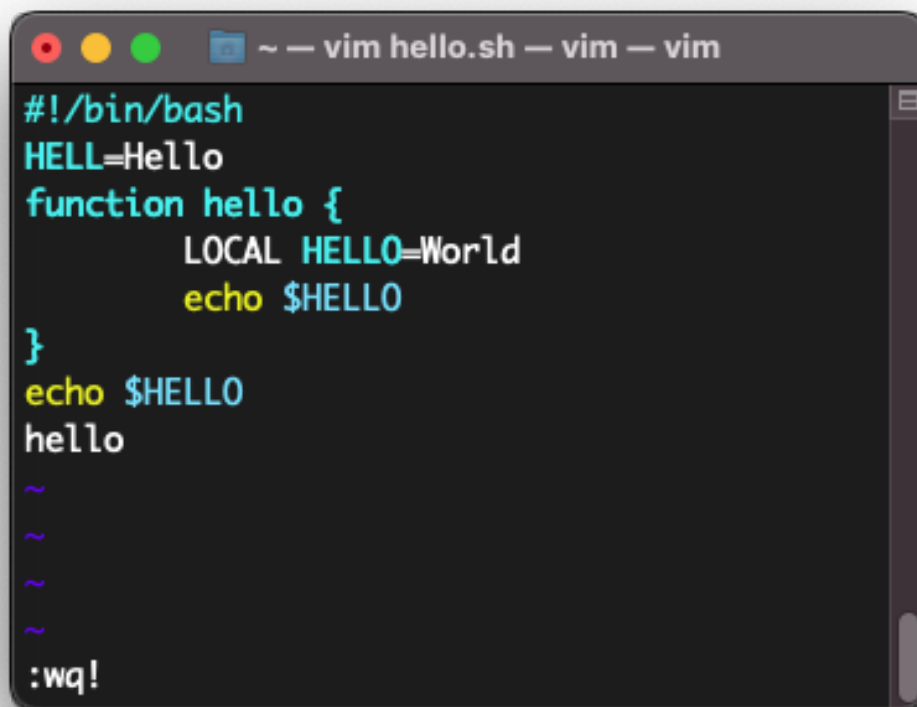
После открытия файла мы перешли в режим вставки и ввели приведенный текст (рис. 3.3).

A screenshot of a vim editor window. The title bar at the top shows a folder icon, a tilde (~), and the text "vim hello.sh — vim — vim". The editor area has a dark background with syntax-highlighted text. The text is a shell script: a shebang line, an assignment, a function definition, and a function call. At the bottom, there are four tilde (~) characters on separate lines.

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

Рис. 3.3: Введенный текст

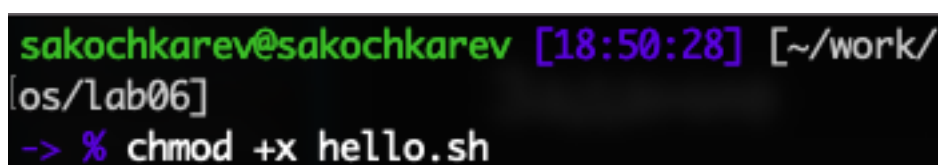
После того, как мы ввели текст, редактор был переведен обратно в командный режим и была введена команда `:wq!` для выхода из файла с сохранением (рис. 3.4).



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

Рис. 3.4: Выход из редактора с сохранением

Последним шагом было изменение разрешений файла, а конкретно выдача прав на выполнение файла командой `chmod +x hello.sh` (рис. 3.5).



```
sakochkarev@sakochkarev [18:50:28] [~/work/
os/lab06]
-> % chmod +x hello.sh
```

Рис. 3.5: Выдача права на выполнение файла

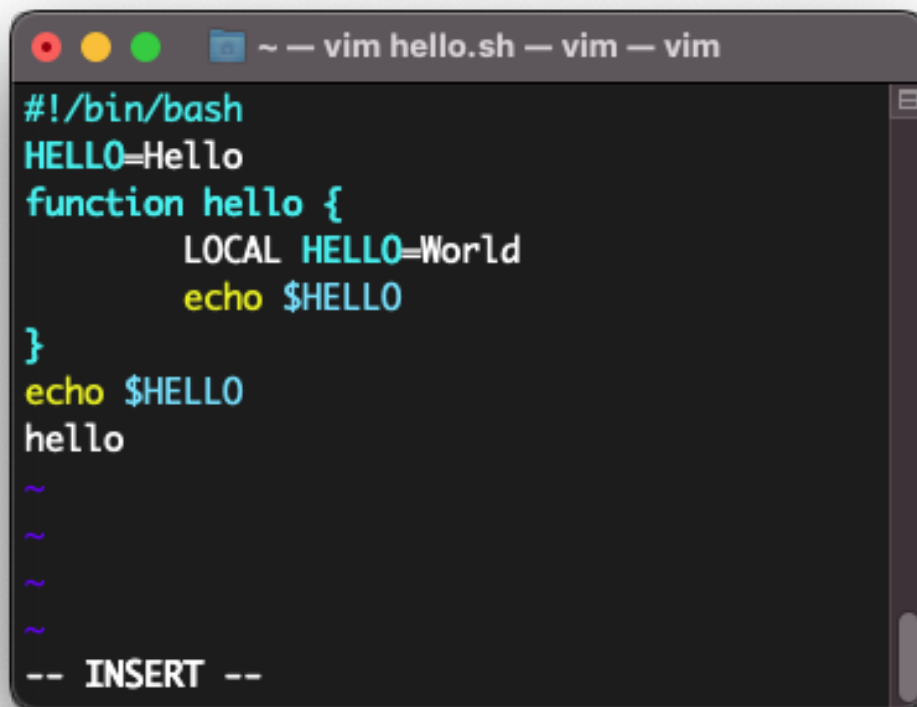
Далее шли задания по редактированию существующего файла с помощью редактора `vi`.

Первым делом был открыт ранее созданный файл в редакторе командой `vi`

~/work/os/lab06/hello.sh.

После периодичным нажатием клавиши `w` курсор был установлен в конце слова `HELL` второй строки.

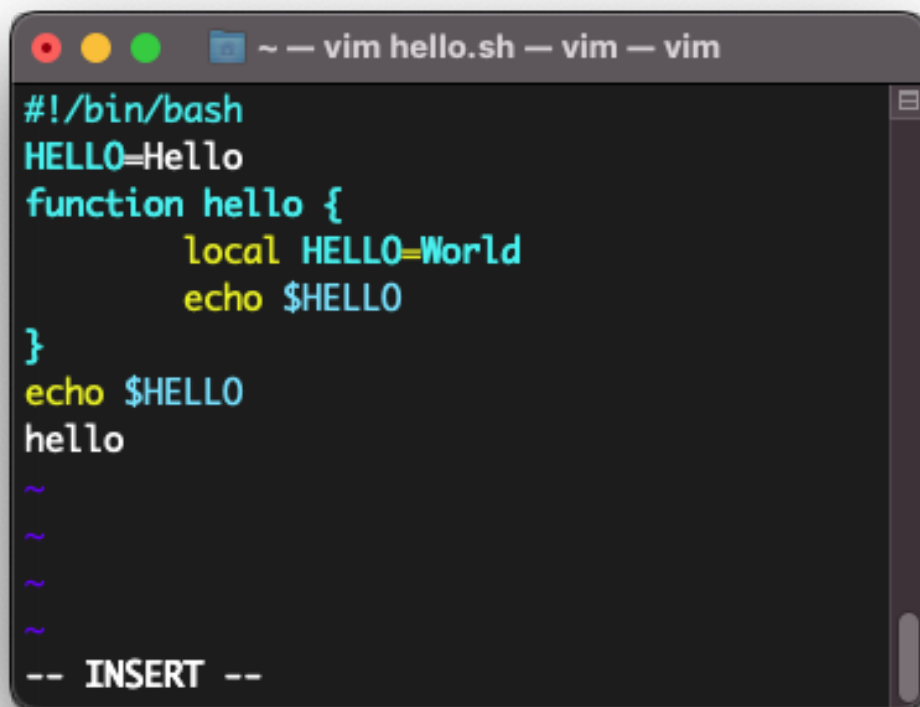
Далее редактор был переведен в режим вставки и `HELL` было заменено на `HELLO` (рис. 3.6).



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

Рис. 3.6: Замена `HELL` на `HELLO`

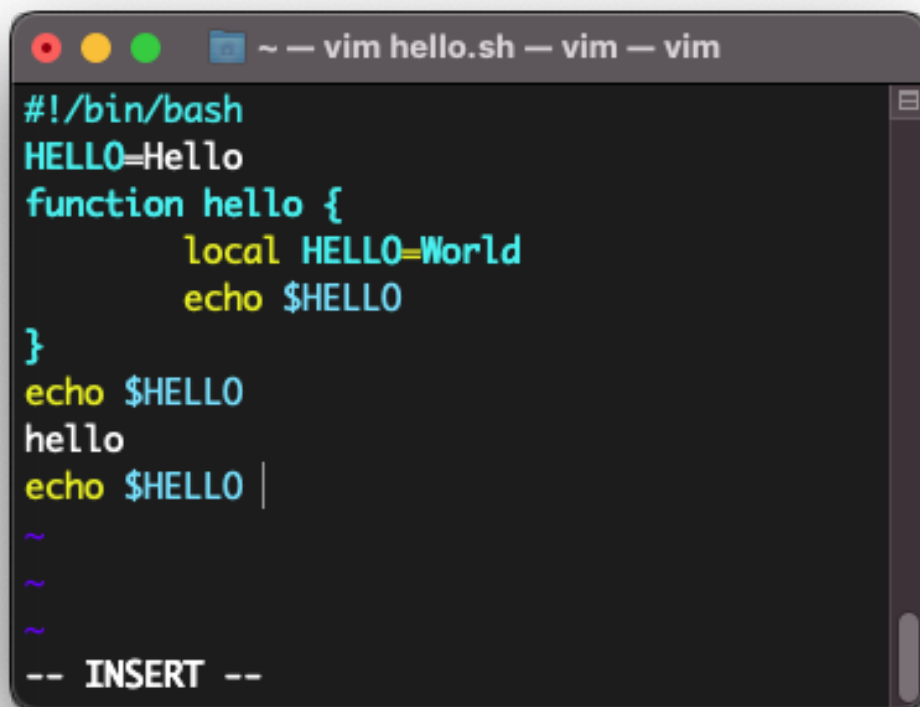
После этого курсор был установлен на четвертую строку и слово `LOCAL` было заменено на `local` (рис. 3.7).



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

Рис. 3.7: Замена LOCAL на local

Далее курсор был установлен на последнюю строку и после нее в режиме вставки была вставлена строка `echo $HELLO` (рис. 3.8).

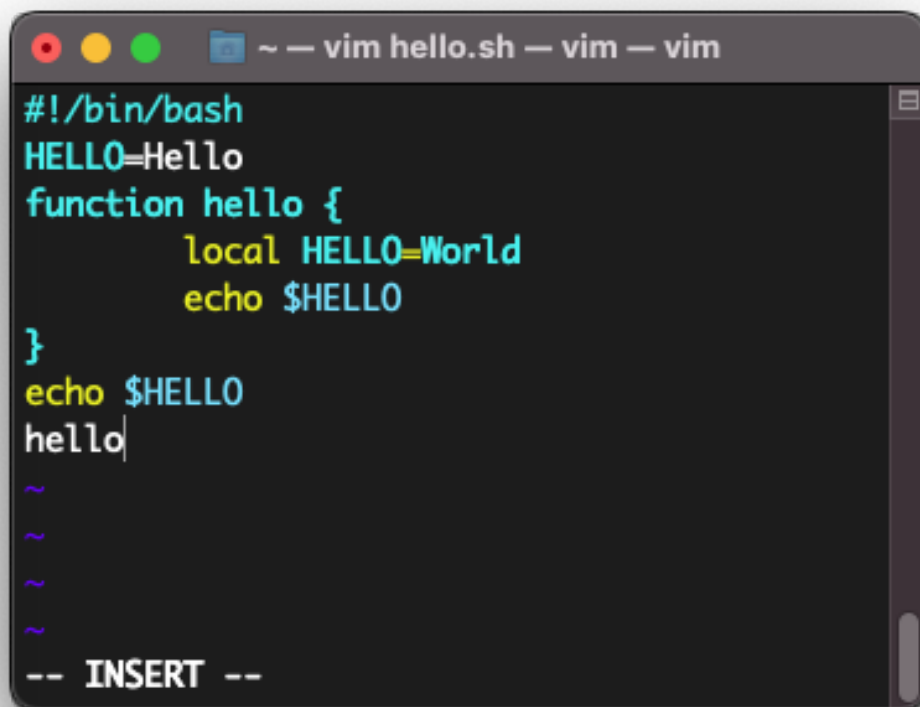


```
#!/bin/bash
HELLO=Hello
function hello {
    local HELLO=World
    echo $HELLO
}
echo $HELLO
hello
echo $HELLO |
```

-- INSERT --

Рис. 3.8: Перенос курсора и вставка строки

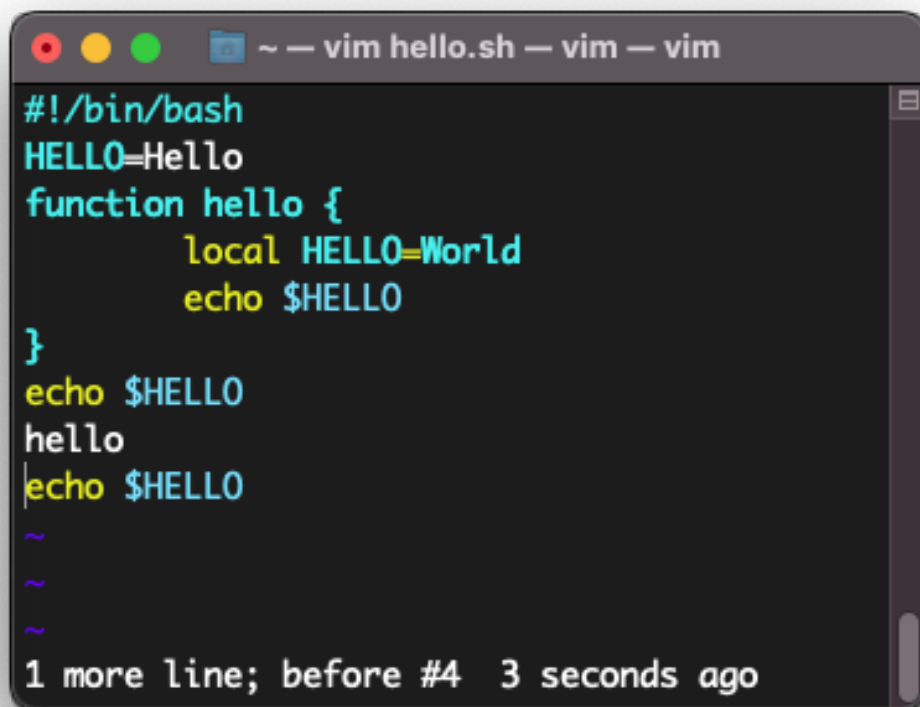
После в командном режим была удалена последняя строка (рис. 3.9).



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

Рис. 3.9: Удаление последней строки

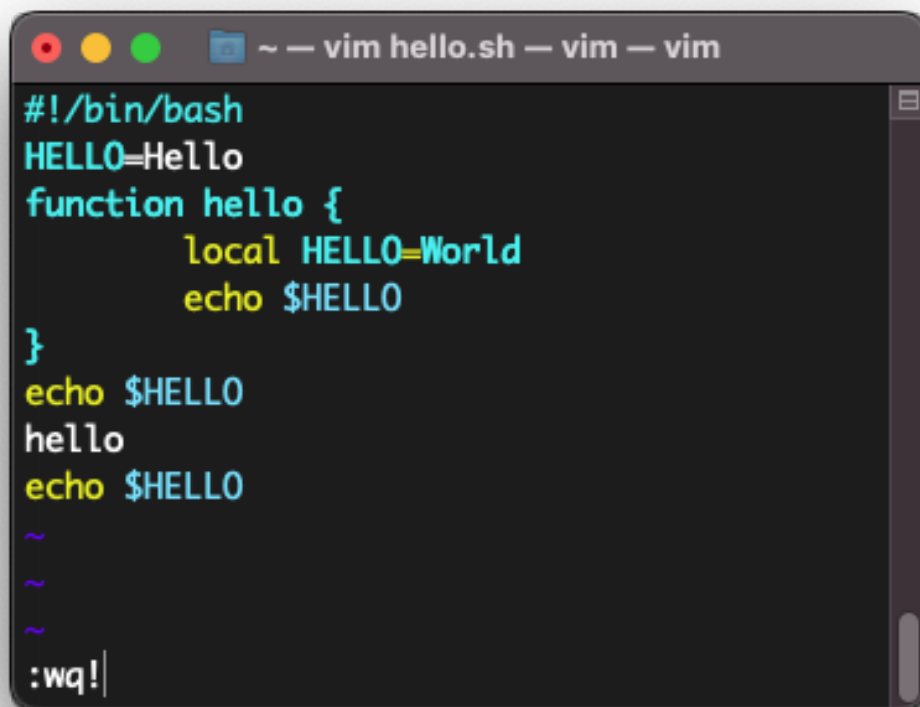
Следующим действием была отмена предыдущего действия (рис. 3.10).



```
#!/bin/bash
HELLO=Hello
function hello {
    local HELLO=World
    echo $HELLO
}
echo $HELLO
hello
echo $HELLO
~
~
~
1 more line; before #4 3 seconds ago
```

Рис. 3.10: Отмена последнего действия

И последним действием был выход из редактора с сохранением изменений (рис. 3.11).



The image shows a terminal window with a vim editor. The title bar at the top reads '~ — vim hello.sh — vim — vim'. The editor contains a shell script with the following content:

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

Below the script, there are three tilde (~) characters on separate lines. At the bottom of the editor, the command `:wq!` is entered, indicating the user wants to save the file and quit the editor.

Рис. 3.11: Выход из редактора с сохранением изменений

4 Выводы

По выполнении лабораторной работы мы получили практические навыки работы с редактором vi, установленным по умолчанию практически во всех дистрибутивах, а также немного дополнительно познакомились с операционной системой Linux.

5 Ответы на 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. Введя команду :q! в командном режиме.

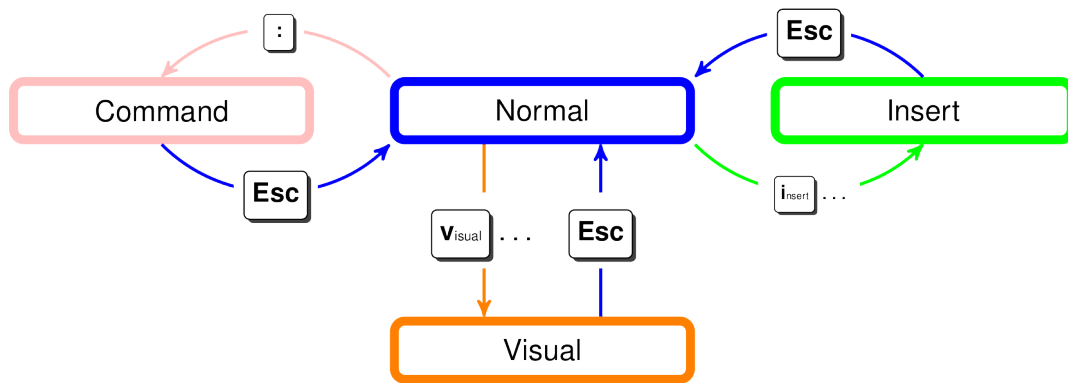
3.

- O – переход в начало строки
- \$ – переход в конец строки
- G – переход в конец файла
- nG – переход на строку с номером n (например 9G)

4. В зависимости от используемой команды редактор vi понимает слово по-разному. При использовании прописных W и B под разделителями понимаются только пробел, табуляция и возврат каретки. При использовании строчных w и b под разделителями понимаются также любые знаки пунктуации.

- 5.
- gg – переход в начало файла
 - G – переход в конец файла

6.
 - Вставка текста текста позволяет вставить текст относительно курсора или строки (n раз)
 - Вставка строки позволяет вставить строку относительно курсора
 - Удаление текста позволяет удалять текст посимвольно, относительно курсора, относительно строки и номерам строк
 - Отмена и повтор произведенных изменений позволяет отменить и повторить произведенные изменения
 - Копирование текста позволяет скопировать строку, n строк или слово в буфер
 - Вставка текста позволяет вставить текст из буфера относительно курсора
 - Замена текста позволяет заменить текст или слова
 - Поиск текста позволяет произвести поиск текста внутри файла
7. Как один из вариантов реализации задания – использовать команду 10A\$Esc, где 10 – кол-во символов на замену в строке.
8. Командой u.
9.
 - Команды копирования и перемещения текста позволяют удалять, перемещать и копировать текст (строки), а также записывать их в отдельный файл
 - Команды записи файла и выхода из редактора позволяют записать измененный текст как с выходом из редактора, так и без, а также выйти из редактора без записи изменений.
10. Можно использовать команду \$.
11. В редакторе Vi существует огромное множество опций. Их назначение можно узнать используя команду :help, например :help G.
12. Зачастую режим работы редактора указан в последней строке.



13.