

Отчёт по лабораторной работе №8

Текстовый редактор vi

Тагиев Павел Фаикович

Содержание

1	Цель работы	4
2	Задание	5
2.1	Задание 1. Создание нового файла с использованием vi	5
2.2	Задание 2. Редактирование существующего файла	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Задание 1. Создание нового файла с использованием vi	9
4.2	Задание 2. Редактирование существующего файла	10
5	Ответы на контрольные вопросы	16
6	Выводы	21
	Список литературы	22

Список иллюстраций

4.1	Создание каталога и задание прав	9
4.2	Набор текста	10
4.3	Изменение слова HELL	11
4.4	Удаление слова LOCAL	11
4.5	Набор слова local	12
4.6	Копирование текста	12
4.7	Вставка текста	13
4.8	Удаление строки	14
4.9	Отмена действия	14
4.10	Сохранение и выход	15
5.1	Граф режимов работы vi	20

1 Цель работы

Познакомиться с операционной системой Linux. Получить практические навыки работы с редактором `vi`, установленным по умолчанию практически во всех дистрибутивах [1].

2 Задание

Задания сформированы в соответствии с [1].

2.1 Задание 1. Создание нового файла с использованием

vi

1. Создайте каталог с именем `~/work/os/lab08`.
2. Перейдите во вновь созданный каталог.
3. Вызовите `vi` и создайте файл `hello.sh`:

```
vi hello.sh
```

4. Нажмите клавишу **i** и введите следующий текст:

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

5. Нажмите клавишу **ESC** для перехода в *командный режим* после завершения ввода текста.

6. Нажмите **:** для перехода в *режим последней строки* и внизу вашего экрана появится приглашение в виде двоеточия.
7. Нажмите **w** (записать) и **q** (выйти), а затем нажмите клавишу **ENTER** для сохранения вашего текста и завершения работы.
8. Сделайте файл исполняемым:

```
chmod +x hello.sh
```

2.2 Задание 2. Редактирование существующего файла

1. Вызовите **vi** на редактирование файла:

```
vi ~/work/os/lab08/hello.sh
```

2. Установите курсор в конец слова **HELL** второй строки.
3. Перейдите в *режим вставки* и замените на **HELLO**. Нажмите **ESC** для возврата в *командный режим*.
4. Установите курсор на четвертую строку и сотрите слово **LOCAL**.
5. Перейдите в *режим вставки* и наберите следующий текст: **local**, нажмите **ESC** для возврата в *командный режим*.
6. Установите курсор на последней строке файла. Вставьте после неё строку, содержащую следующий текст: **echo \$HELLO**.
7. Нажмите **ESC** для перехода в *командный режим*.
8. Удалите последнюю строку.
9. Введите команду отмены изменений **u** для отмены последней команды.
10. Введите символ **:** для перехода в *режим последней строки*. Запишите произведённые изменения и выйдите из **vi**.

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 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.

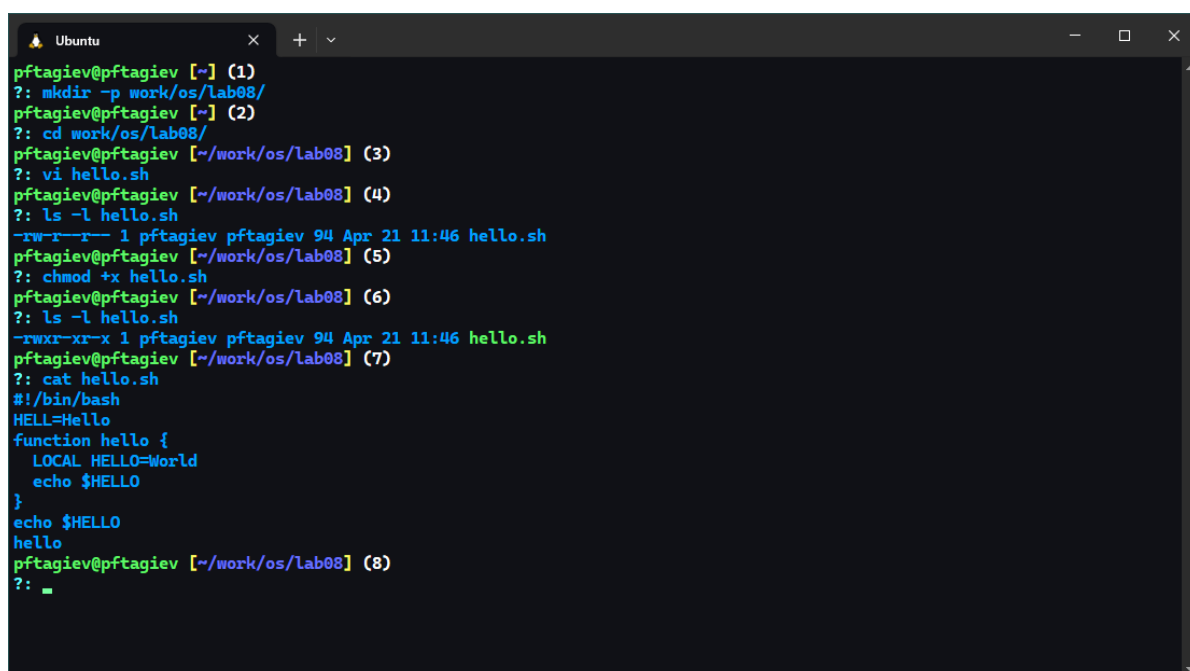
Переход в *командный режим* осуществляется нажатием клавиши **ESC**. Для выхода из редактора `vi` необходимо перейти в *режим последней строки*: находясь в командном режиме, нажать **SHIFT+;** (по сути символ **:** — двоеточие), затем:

- 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 [1].

4 Выполнение лабораторной работы

4.1 Задание 1. Создание нового файла с использованием vi



```
pftagiev@pftagiev [~] (1)
?: mkdir -p work/os/lab08/
pftagiev@pftagiev [~] (2)
?: cd work/os/lab08/
pftagiev@pftagiev [~/work/os/lab08] (3)
?: vi hello.sh
pftagiev@pftagiev [~/work/os/lab08] (4)
?: ls -l hello.sh
-rw-r--r-- 1 pftagiev pftagiev 94 Apr 21 11:46 hello.sh
pftagiev@pftagiev [~/work/os/lab08] (5)
?: chmod +x hello.sh
pftagiev@pftagiev [~/work/os/lab08] (6)
?: ls -l hello.sh
-rwxr-xr-x 1 pftagiev pftagiev 94 Apr 21 11:46 hello.sh
pftagiev@pftagiev [~/work/os/lab08] (7)
?: cat hello.sh
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
pftagiev@pftagiev [~/work/os/lab08] (8)
?: _
```

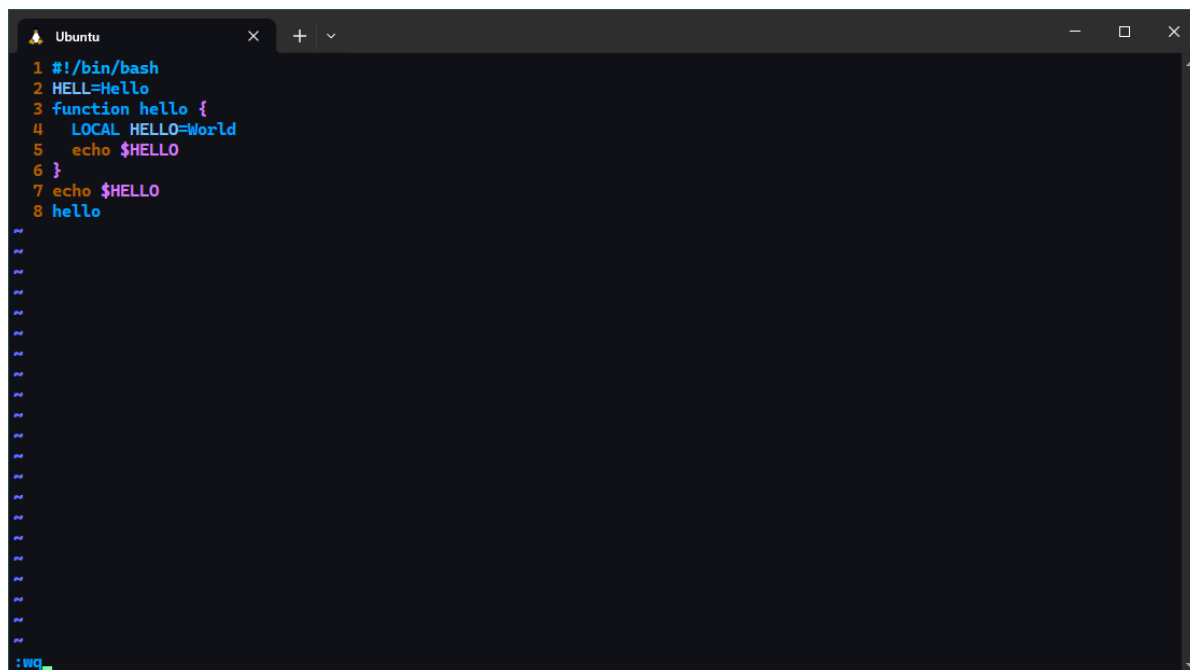
Рис. 4.1: Создание каталога и задание прав

Следуя заданию создадим каталог `~/work/os/lab08`, затем перейдем в него и создадим файл `hello.sh` через `vi` (этот шаг можно увидеть на рис. 4.1 на промтах (1)-(3)). Уже находясь в редакторе нажмем клавишу `i`, чтобы перейти в режим вставки и наберем требуемый текст (рис. 4.2).

После того как текст был набран перейдем в командный режим нажав клавишу

ESC, из него перейдем в *режим последней строки* нажав на клавиатуре символ **:**. Наберем **wq** (рис. 4.2) и нажмем клавишу **ENTER**, чтобы записать файл и выйти из него.

На рис. 4.1 в промтах (4)-(6) можно увидеть задание права на исполнение файла `hello.sh`. А в промте (7) вывод сохраненного файла командой `cat`.



```
1 #!/bin/bash
2 HELLO=Hello
3 function hello {
4     LOCAL HELLO=World
5     echo $HELLO
6 }
7 echo $HELLO
8 hello
```

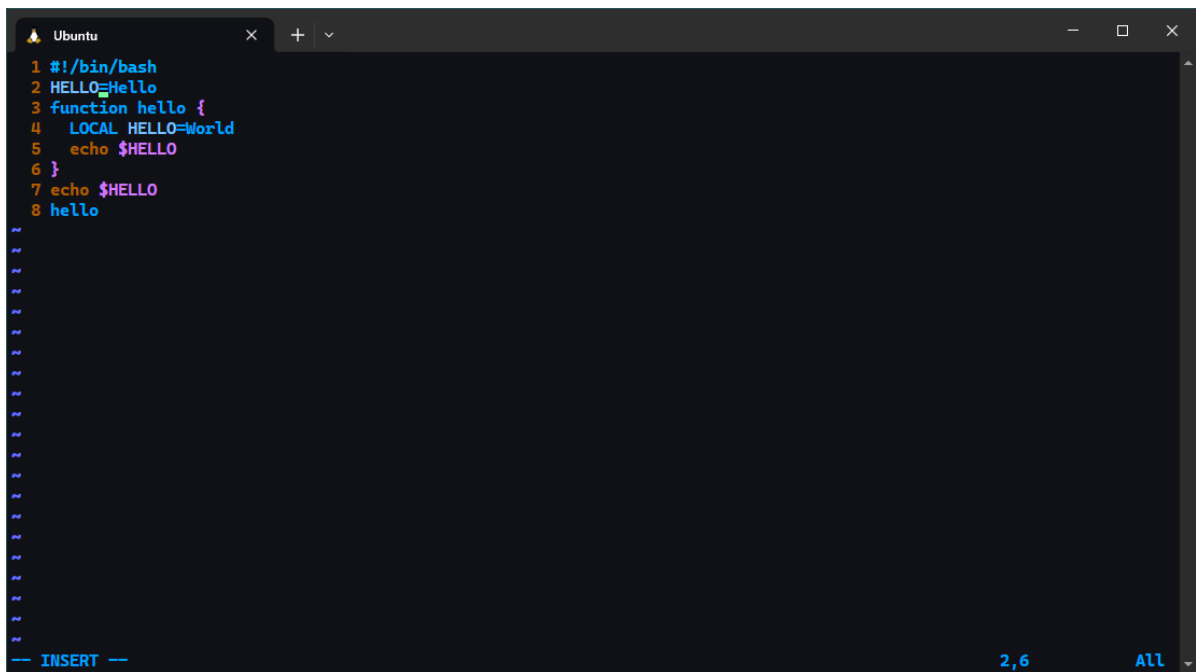
Рис. 4.2: Набор текста

4.2 Задание 2. Редактирование существующего файла

Снова откроем созданный ранее файл `hello.sh` командой:

```
vi ~/work/os/lab08/hello.sh
```

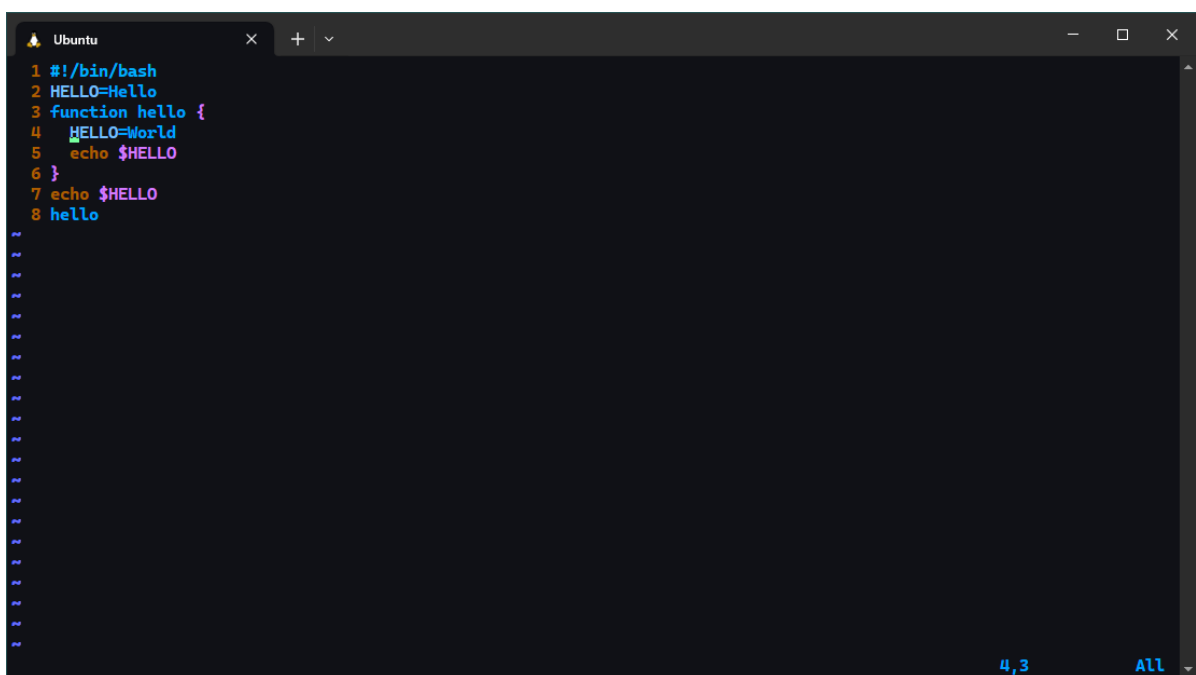
Переместим курсор на вторую строку, например клавишами **j** или **k**, потом нажмем **w** для переноса курсора в конец слова `HELL` и войдем в *режим вставки* клавишей **i**. Заменяем слово `HELL` на `HELLO`, результат можно увидеть на рис. 4.3. Нажмем **ESC** чтобы вернуться в *командный режим*.



```
1 #!/bin/bash
2 HELLO=Hello
3 function hello {
4     LOCAL HELLO=World
5     echo $HELLO
6 }
7 echo $HELLO
8 hello
```

— INSERT — 2,6 All

Рис. 4.3: Изменение слова HELLO



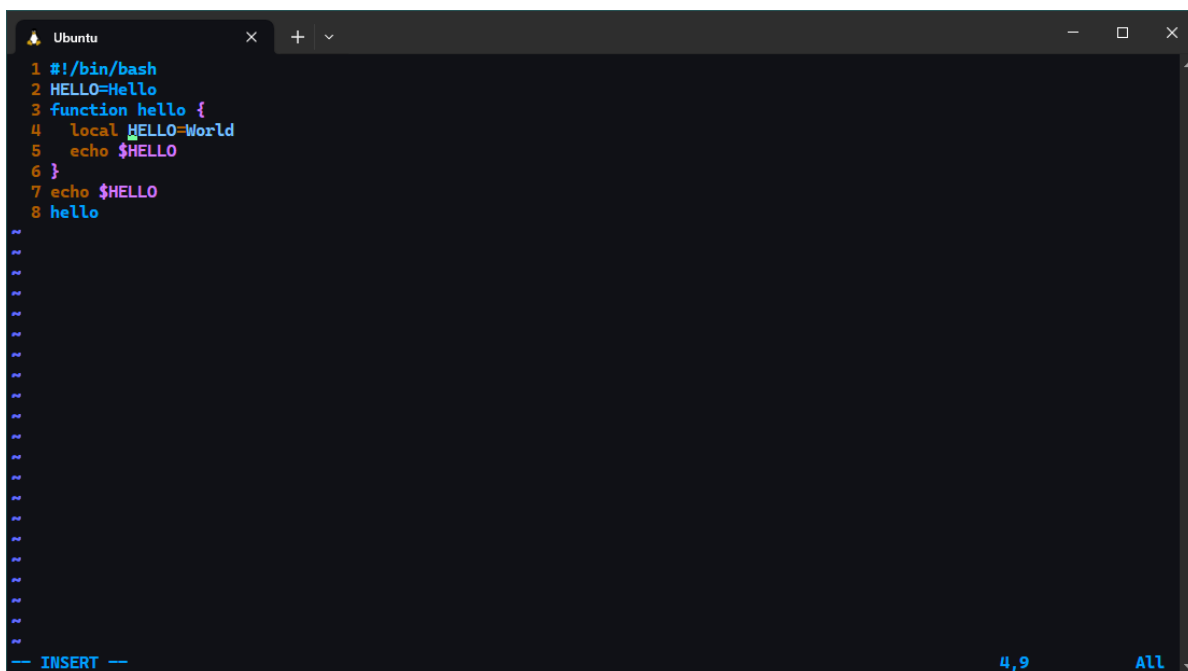
```
1 #!/bin/bash
2 HELLO=Hello
3 function hello {
4     HELLO=World
5     echo $HELLO
6 }
7 echo $HELLO
8 hello
```

4,3 All

Рис. 4.4: Удаление слова LOCAL

Находясь в *командном режиме* наберем **4G**, чтобы перейти к четвертой строке файла. Затем **wd**, чтобы удалить слово **LOCAL** в буфер (рис. 4.4). Перейдем

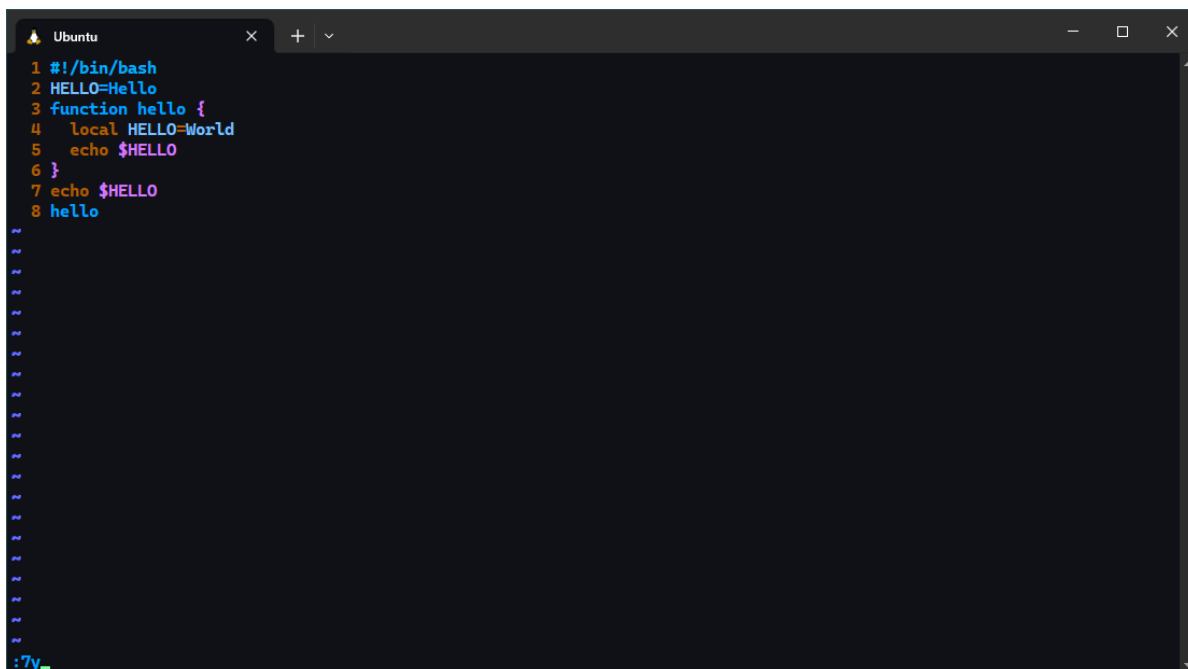
в режим вставки клавишей **i** и наберем текст: `local`, как на рис. 4.5. После вернемся в командный режим с помощью **ESC**.



```
1 #!/bin/bash
2 HELLO=Hello
3 function hello {
4   local HELLO=World
5   echo $HELLO
6 }
7 echo $HELLO
8 hello
```

The terminal window shows a function definition. The word `local` is being typed on line 4. The status bar at the bottom indicates 'INSERT' mode and cursor position '4,9'.

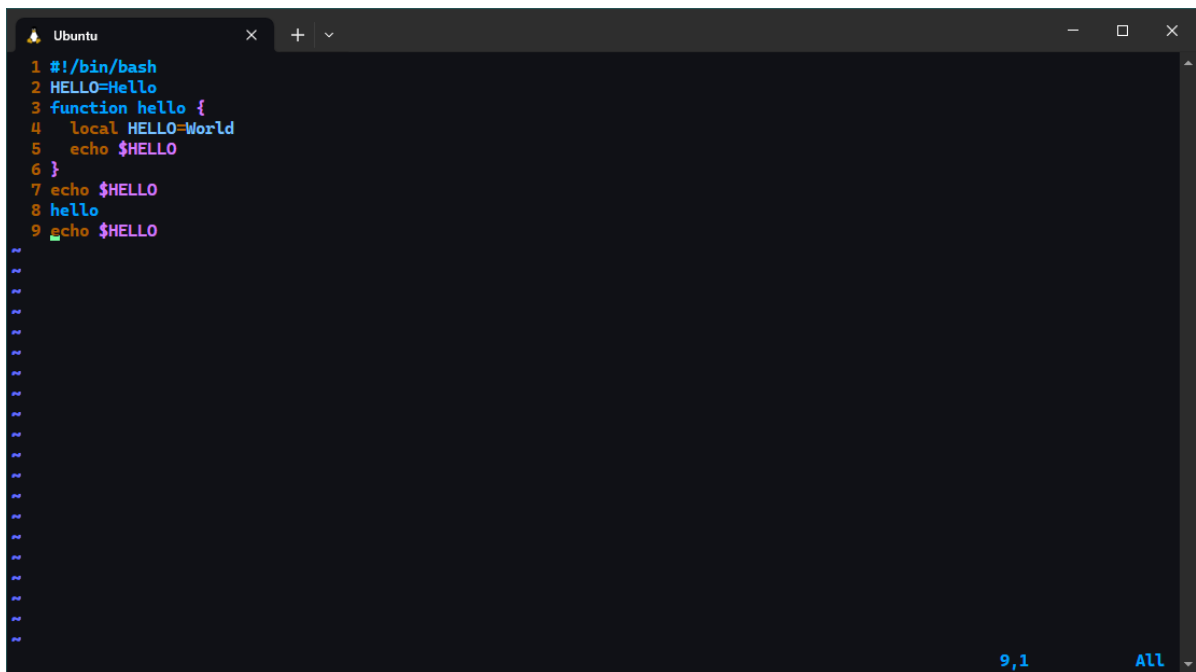
Рис. 4.5: Набор слова `local`



```
1 #!/bin/bash
2 HELLO=Hello
3 function hello {
4   local HELLO=World
5   echo $HELLO
6 }
7 echo $HELLO
8 hello
```

The terminal window shows the same function definition. The cursor is now at the end of line 8, ready for the next command.

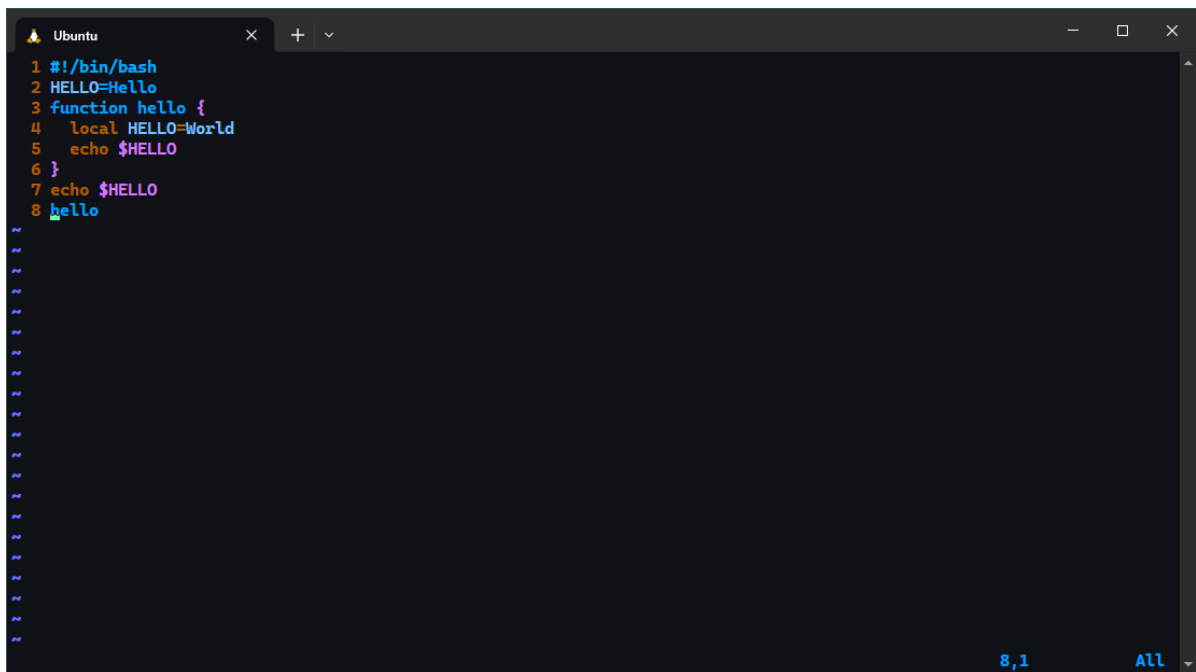
Рис. 4.6: Копирование текста



```
1 #!/bin/bash
2 HELLO=Hello
3 function hello {
4     local HELLO=World
5     echo $HELLO
6 }
7 echo $HELLO
8 hello
9 echo $HELLO
```

Рис. 4.7: Вставка текста

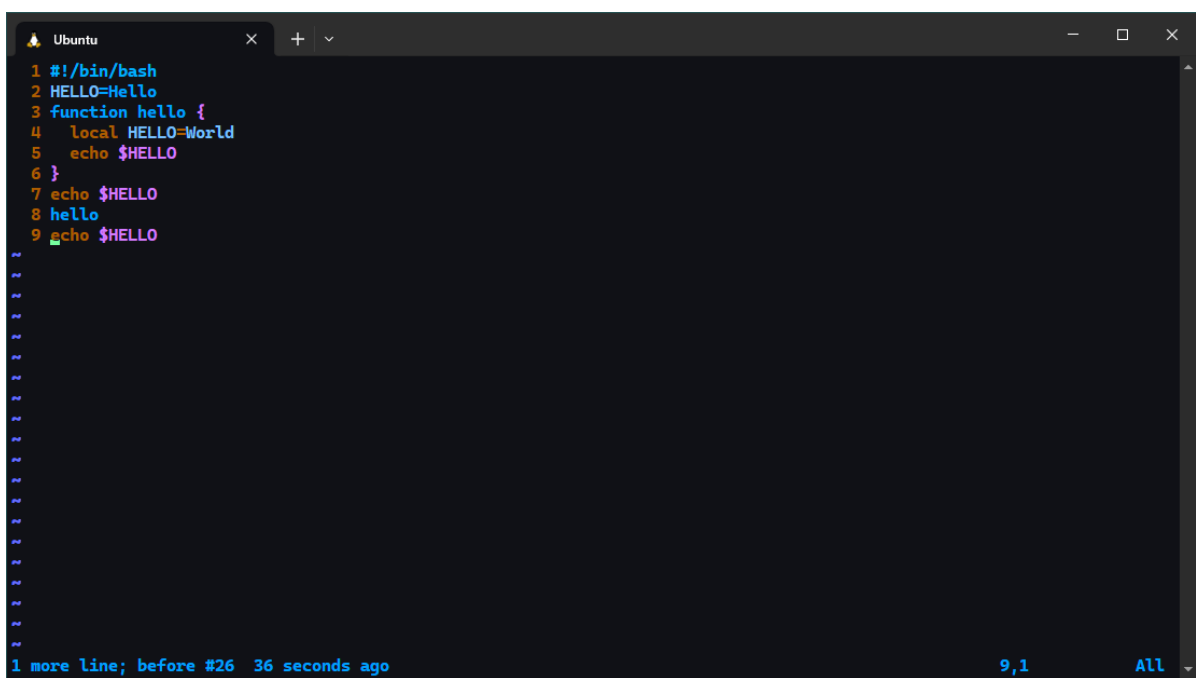
Теперь перейдем на последнюю строку нажав **G**, затем скопируем текст со строки 7 перейдя в *режим последней строки* и введя **7y** (рис. 4.6). Вставим скопированную строку нажав **p**, результат на рис. 4.7. Удалим последнюю строку введя **dd** (рис. 4.8). Отменим последнее действие введя **u** (рис. 4.9).



```
1 #!/bin/bash
2 HELLO=Hello
3 function hello {
4     local HELLO=World
5     echo $HELLO
6 }
7 echo $HELLO
8 hello
```

The screenshot shows a terminal window titled 'Ubuntu'. It contains a script with 8 lines. Line 8, which is 'hello', is highlighted with a blue background. The status bar at the bottom right shows '8,1' and 'All'.

Рис. 4.8: Удаление строки

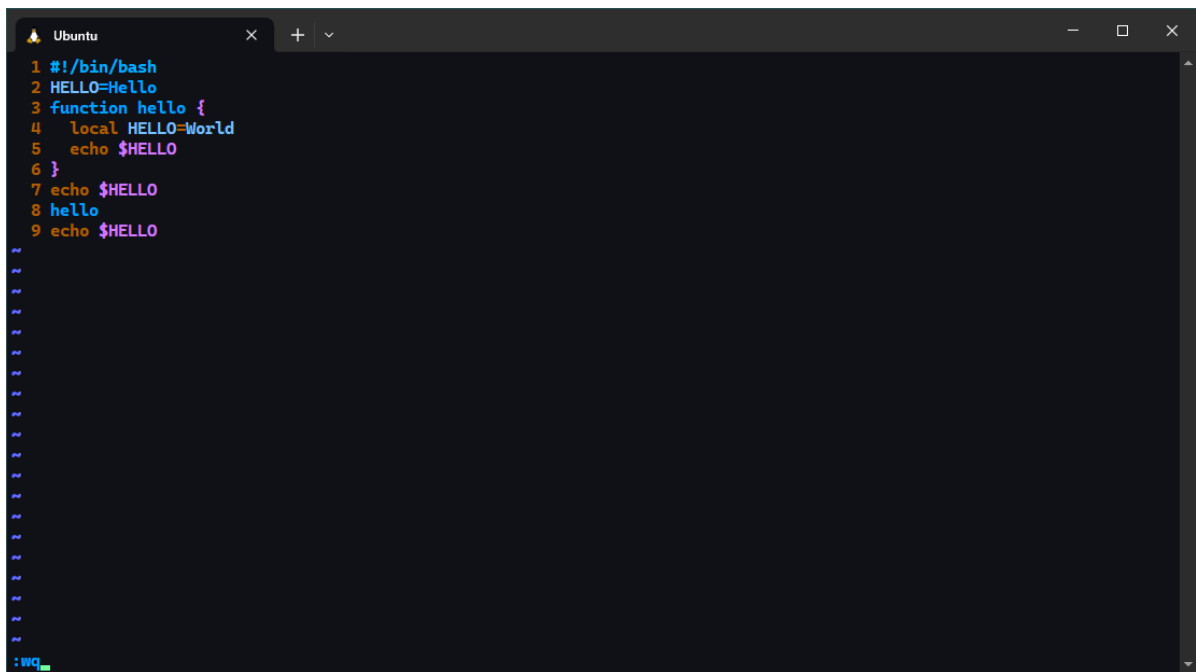


```
1 #!/bin/bash
2 HELLO=Hello
3 function hello {
4     local HELLO=World
5     echo $HELLO
6 }
7 echo $HELLO
8 hello
9 echo $HELLO
```

The screenshot shows the same terminal window as before, but now with 9 lines. Line 9, which is 'echo \$HELLO', is highlighted with a green background. The status bar at the bottom left shows '1 more line; before #26 36 seconds ago', and the bottom right shows '9,1' and 'All'.

Рис. 4.9: Отмена действия

Сохраним и выйдем из файла набрав в *режиме последней строки* **wq** (рис. 4.10) и нажав **ENTER** .

A terminal window titled 'Ubuntu' with a dark background and light-colored text. The window contains a script with 9 lines of code. Lines 1-6 define a function 'hello' that sets a local variable 'HELLO' to 'World' and echoes the value of '\$HELLO'. Lines 7-9 execute the script: line 7 echoes the global 'HELLO' (Hello), line 8 calls the 'hello' function, and line 9 echoes the local 'HELLO' (World). The prompt ': wq' is visible at the bottom left.

```
1 #!/bin/bash
2 HELLO=Hello
3 function hello {
4     local HELLO=World
5     echo $HELLO
6 }
7 echo $HELLO
8 hello
9 echo $HELLO

:wq
```

Рис. 4.10: Сохранение и выход

5 Ответы на контрольные вопросы

1. Дайте краткую характеристику режимам работы редактора `vi`.

- *командный режим* — предназначен для ввода команд редактирования и навигации по редактируемому файлу;
- *режим вставки* — предназначен для ввода содержания редактируемого файла;
- *режим последней строки* — используется для записи изменений в файл и выхода из редактора.

2. Как выйти из редактора, не сохраняя произведённые изменения?

Нужно использовать команду `q!`, в *режиме последней строки*.

3. Назовите и дайте краткую характеристику командам позиционирования.

- `k` — переход на одну строку вверх;
- `j` — переход на одну строку вниз;
- `h` — переход на один символ влево;
- `l` — переход на один символ вправо;
- `0` — переход в начало строки;
- `$` — переход в конец строки;
- `gg` — переход в начало файла;
- `G` — переход в конец файла;
- `nG` — переход на строку с номером `n`;
- `CTRL + d` — перейти на пол-экрана вперед;
- `CTRL + u` — перейти на пол-экрана назад;

- **CTRL + f** — перейти на страницу вперед;
- **CTRL + b** — перейти на страницу назад;
- **w** — перейти на слово вперед;
- **nw** — перейти на n слов вперед;
- **b** — перейти на слово назад;
- **nb** — перейти на n слов назад;

4. Что для редактора **vi** является словом?

В **vi**, словом считается последовательность символов, разделенных пробелами, табуляциями, переносами строки, или другими символами, которые считаются разделителями слов (например, точка, запятая, точка с запятой и т.д.).

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

Для перехода в начало файла можно использовать **gg**, для перехода в конец **G**.

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

- Вставка текста:
 - **a** — вставить текст после курсора;
 - **A** — вставить текст в конец строки;
 - **i** — вставить текст перед курсором;
 - **ni** — вставить текст n раз;
 - **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;

- **dw** — удалить одно слово в буфер;
- **d\$** — удалить в буфер текст от курсора до конца строки;
- **d0** — удалить в буфер текст от начала строки до позиции курсора;
- **dd** — удалить в буфер одну строку;
- **ndd** — удалить в буфер n строк.
- Отмена и повтор произведённых изменений:
 - **u** — отменить последнее изменение;
 - **.** — повторить последнее изменение.
- Копирование текста в буфер:
 - **Y** — скопировать строку в буфер;
 - **nY** — скопировать n строк в буфер;
 - **yw** — скопировать слово в буфер.
- Вставка текста из буфера:
 - **p** — вставить текст из буфера после курсора;
 - **P** — вставить текст из буфера перед курсором.
- Замена текста:
 - **cw** — заменить слово;
 - **ncw** — заменить n слов;
 - **c\$** — заменить текст от курсора до конца строки;
 - **r** — заменить слово;
 - **R** — заменить текст.

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

Находясь в *командном режиме* переместим курсор в начало строки, затем воспользуемся командой **c\$**, чтобы очистить ее. После находясь в *режиме вставки* можно начать заполнять строку долларами. Еще **vi** поддерживает *макросы*, с помощью которых можно автоматизировать этот процесс, но их использование выходит за рамки этой работы.

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

Перейти в *командный режим* и воспользоваться командой **u**.

9. Назовите и дайте характеристику основным группам команд режима последней 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,jt~~k~~** — копировать строки с *i* по *j* в строку *k*;
 - **:i,jw** 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;
 - **:wq** — 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.

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

Можно включить опцию: `set list`, для отображения невидимых символов. Конец строки будет отмечен знаком доллара.

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

Вести список всех опций можно введя в *режиме последней строки* команду `:set all`. Открыть документацию ко всем опциям можно

командой `:help set`, чтобы найти информацию по конкретной команде нужно использовать поиск `/имя_команды`. Подробнее можно узнать в [2].

12. Как определить режим работы редактора `vi`? Чтобы определить текущий режим работы `vi`, достаточно просто взглянуть на последнюю строку редактора.

- Если вы видите надпись `-- INSERT --`, значит вы находитесь в *режиме вставки*.
- Если вы видите символ `:`, значит вы находитесь в *режиме последней строки*.
- Если ничего не отображается, значит это *командный режим*.

13. Постройте граф взаимосвязи режимов работы редактора `vi`.

Построенный граф можно увидеть на рис. 5.1.

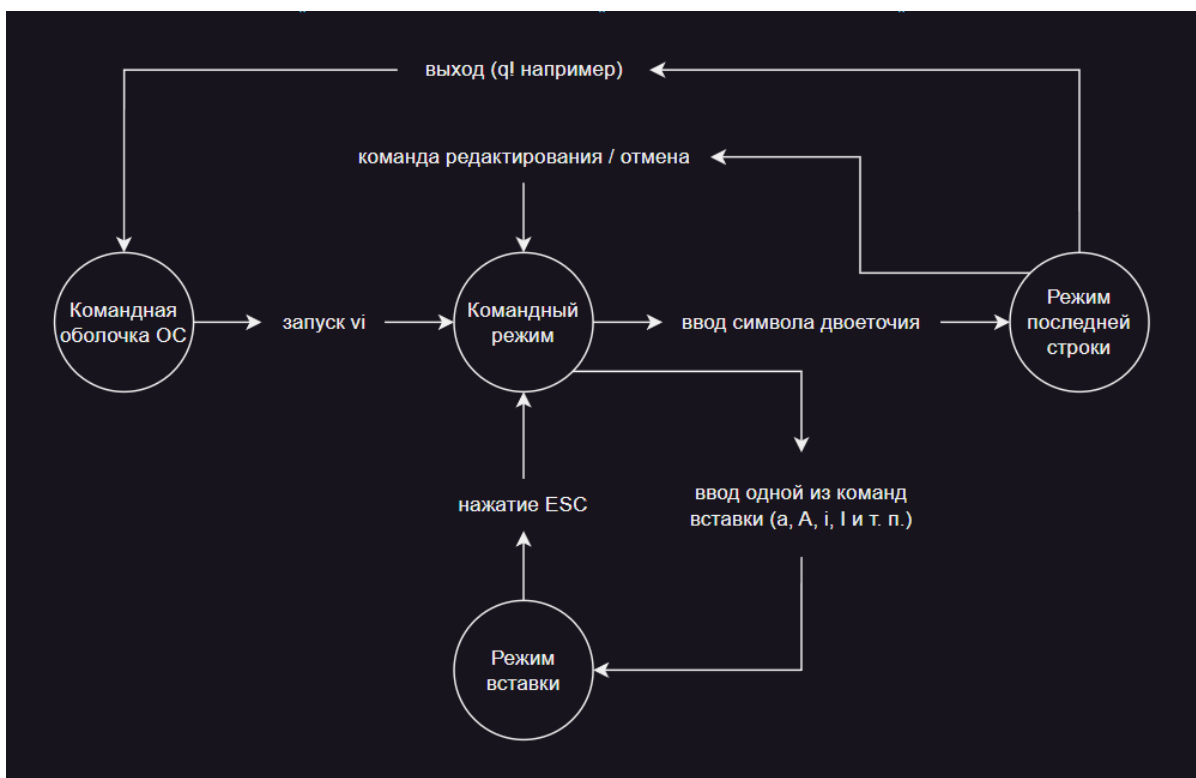


Рис. 5.1: Граф режимов работы `vi`

6 Выводы

В этой работе мы получили практические навыки по работе с `vi`, одним из самых популярных текстовых редакторов в Unix-подобных ОС.

Список литературы

1. Кулябов. Операционные системы. Москва: РУДН, 2016. 118 с.
2. Vim documentation: help [Электронный ресурс]. 2010. URL: <https://vimdoc.sourceforge.net/html/doc/help.html>.