

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

## Факультет физико-математических и естественных наук

### Кафедра прикладной информатики и теории вероятностей

## ОТЧЕТ

### ПО ЛАБОРАТОРНОЙ РАБОТЕ № 9

#### *дисциплина: Операционные системы*

Студент: Губина Ольга Вячеславовна Группа: НПИбд-01-20

Преподаватель: Велиева Татьяна Рефатовна

МОСКВА

2021 г.

#### Цель работы:

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

#### Теоретические сведения:

В данной лабораторной работе мы ознакомимся с редактором vi. Для этого сначала ознакомимся с теорией.

Интерактивный экранный редактор vi (Visual display editor) используется по умолчанию в большинстве дистрибутивов Linux в качестве текстового редактора.

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.

Основными группами команд для редактора vi являются:

- Команды управления курсором;
- Команды позиционирования;
- Команды перемещения по файлу;
- Команды перемещения по словам;
- Команды редактирования:
  - Вставка текста;
  - Вставка строки;
  - Удаление текста;
  - Отмена и повтор произведённых изменений;
  - Копирование текста в буфер;

- Вставка текста из буфера;
- Замена текста;
- Поиск текста;
- Команды редактирования в режиме командной строки:
  - Копирование и перемещение текста;
  - Запись в файл и выход из редактора;
- Опции.

## Задачи:

1. Ознакомиться с редактором vi;
2. Научиться создавать и редактировать файлы с помощью редактора vi;
3. Ознакомиться с командами vi, использовать их на практике.

---

## Выполнение работы:

1. Ознакомились с теоретическим материалом, предложенным в материалах к *лабораторной работе №9* [\[1\]](#).
2. Ознакомились с редактором vi, посредством прочтения материалов и изучения команд для редактора. Подробнее с ними можно ознакомиться в *справочнике по редактору* [\[2\]](#)
3. Теперь выполним предложенные упражнения, используя команды vi.

### Задание 1. Создание нового файла с использованием vi

1. Создадим каталог с именем `~/work/os/lab06` (рисунок 1). Каталог `work` уже был создан, поэтому перейдем в него через команду `cd`, в нем создадим каталог `os` через `mkdir`, перейдем в него, и там создадим каталог `lab09`.

Рисунок 1: создание каталога `~/work/os/lab06`:

```
[ovgubina@localhost ~]$ cd work
[ovgubina@localhost work]$ mkdir os
[ovgubina@localhost work]$ cd ~/work/os
[ovgubina@localhost os]$ mkdir lab09
[ovgubina@localhost os]$ cd ~/work/os/lab09
bash: /home/ovgubina/work/os/lab09: Это каталог
[ovgubina@localhost os]$ cd ~/work/os/lab09
[ovgubina@localhost lab09]$ █
```

2. Перейдем во вновь созданный каталог: `cd~/work/os/lab06` (рисунок 1).
3. Вызовем `vi` и создадим файл `hello.sh`: `vi hello.sh` (рисунок 2). Поскольку файл первоначально создан не был, вызовом данной команды мы не только откроем его, но и создадим. Видим, что открылся пустой файл `hello.sh` (рисунок 3)

Рисунок 2: вызов `vi` и создание файла `hello.sh`:

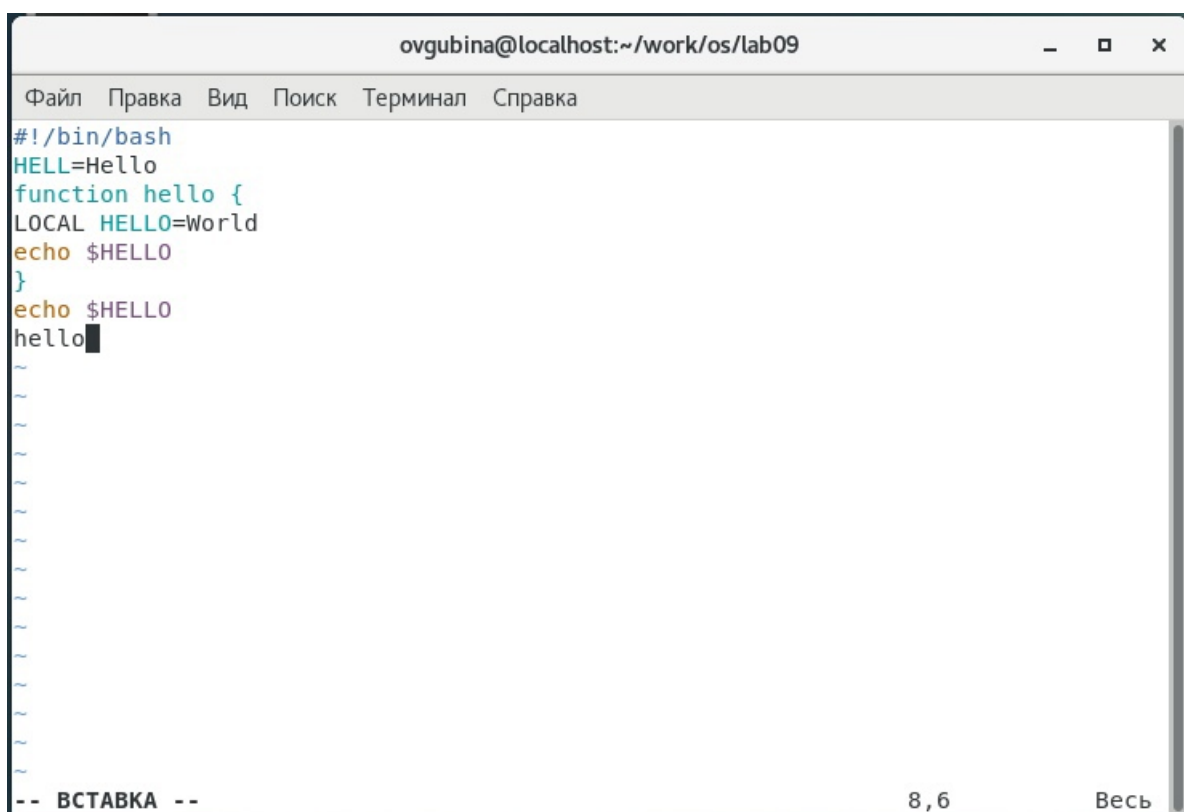
```
[ovgubina@localhost lab09]$ vi hello.sh █
```

Рисунок 3: открытие нового файла `hello.sh`:



4. Нажимаем клавишу `i`, тем самым входя в режим вставки (отображается в левом нижнем углу), и введем предложенный текст (рисунки 4).

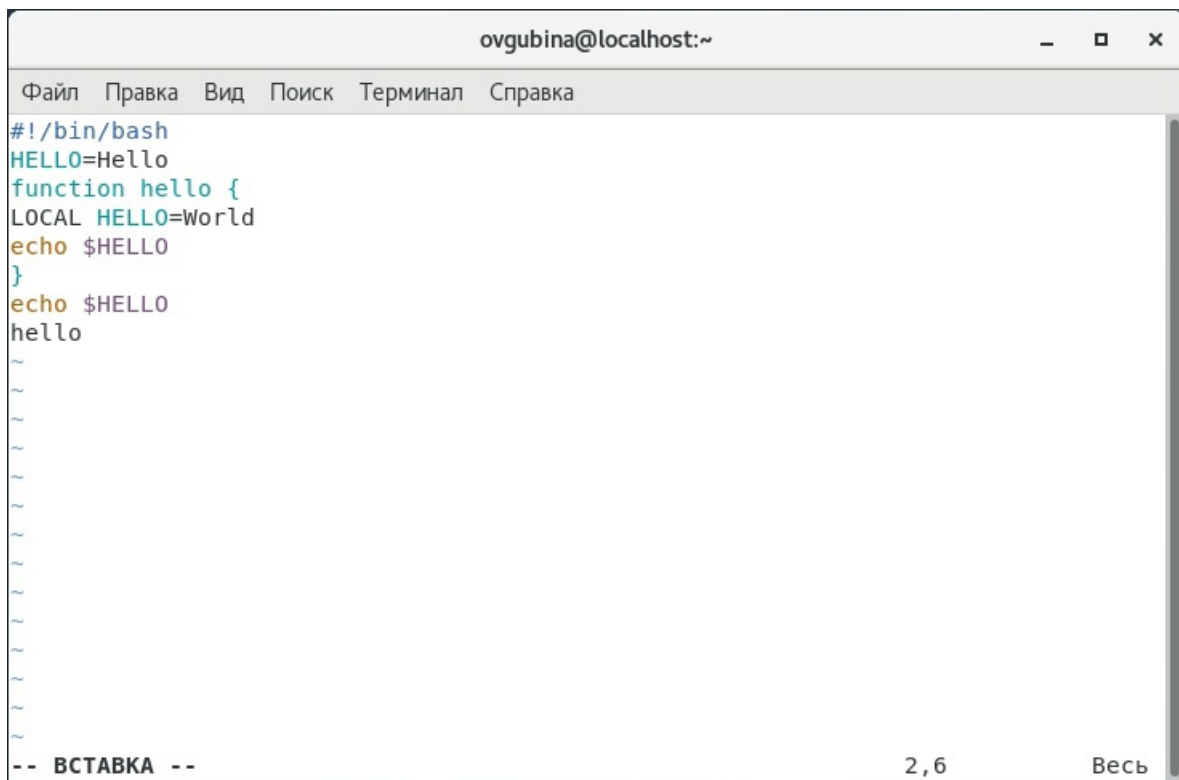
Рисунок 4: редактирование нового файла - ввод текста:



5. Нажмем клавишу `Esc` для перехода в командный режим после завершения ввода текста.
6. Нажмем `:` для перехода в режим последней строки. Видим, что внизу нашего экрана появляется приглашение в виде двоеточия (рисунки 5).
7. Вводим после двоеточия `w` (записать изменения) и `q` (выйти из редактора) (рисунки 5). Теперь нажмем клавишу `Enter` для сохранения нашего текста и завершения работы.

Рисунок 5: режим последней строки:

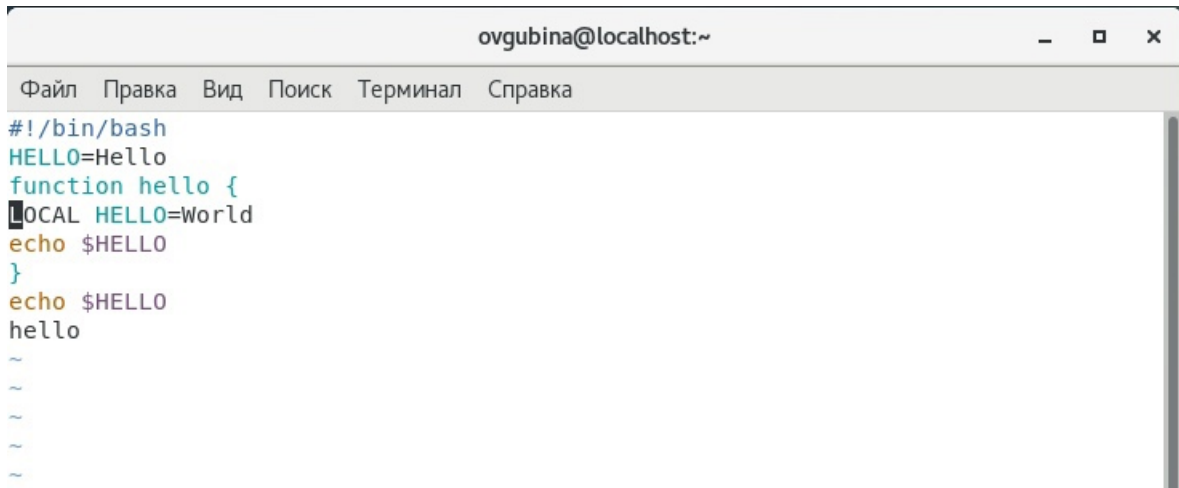




```
ovgubina@localhost:~  
Файл Правка Вид Поиск Терминал Справка  
#!/bin/bash  
HELLO=Hello  
function hello {  
LOCAL HELLO=World  
echo $HELLO  
}  
echo $HELLO  
hello  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
-- ВСТАВКА -- 2,6 Весь
```

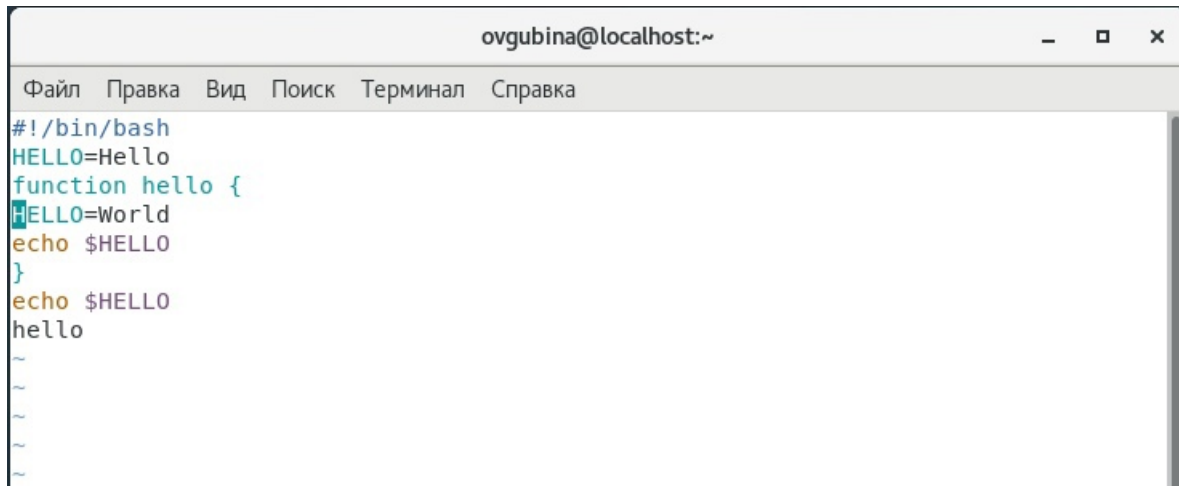
4. Установим курсор на начало четвертой (рисунки 10) строки и командой `d+w`, позволяющей стирать первое слово строки, сотрем слово `LOCAL` (рисунки 11).

Рисунок 10: курсор на начало четвертой строки:



```
ovgubina@localhost:~  
Файл Правка Вид Поиск Терминал Справка  
#!/bin/bash  
HELLO=Hello  
function hello {  
LOCAL HELLO=World  
echo $HELLO  
}  
echo $HELLO  
hello  
~  
~  
~  
~  
~
```

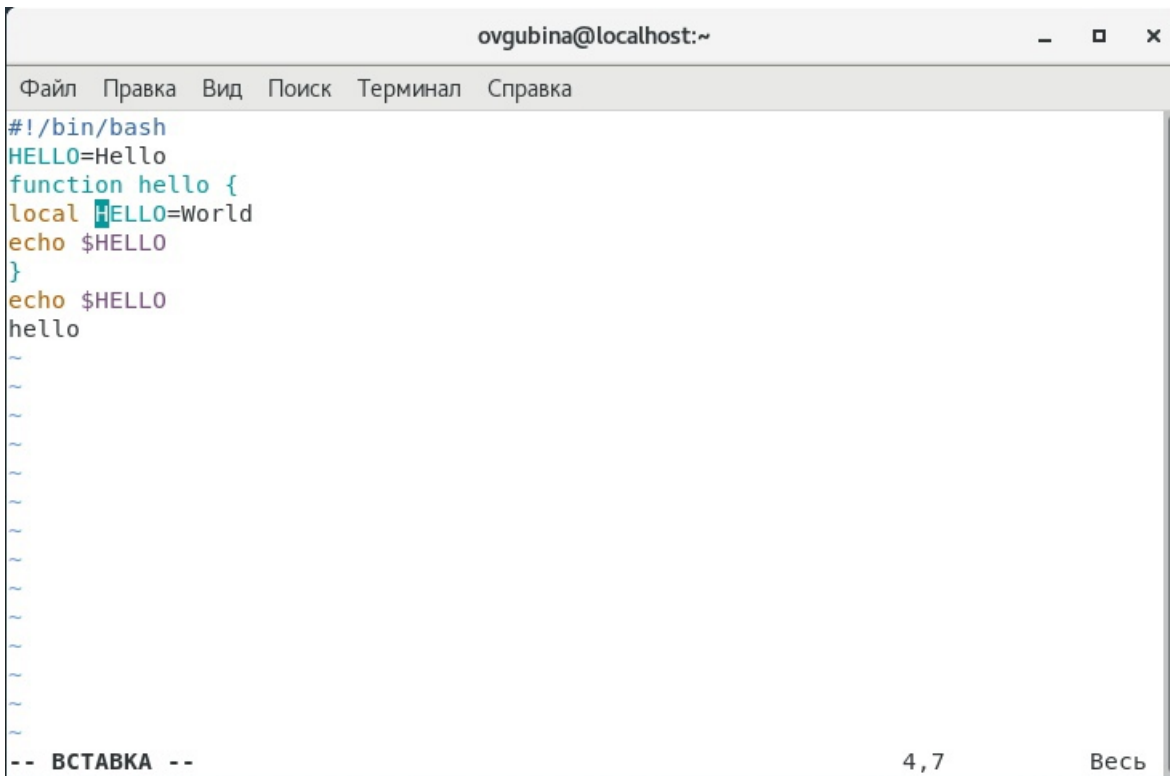
Рисунок 11: стираем первое слово строки:



```
ovgubina@localhost:~  
Файл Правка Вид Поиск Терминал Справка  
#!/bin/bash  
HELLO=Hello  
function hello {  
HELLO=World  
echo $HELLO  
}  
echo $HELLO  
hello  
~  
~  
~  
~  
~
```

5. Перейдем в режим вставки `i` и наберем следующий текст: `local` (рисунки 12); и нажмем `Esc` для возврата в командный режим.

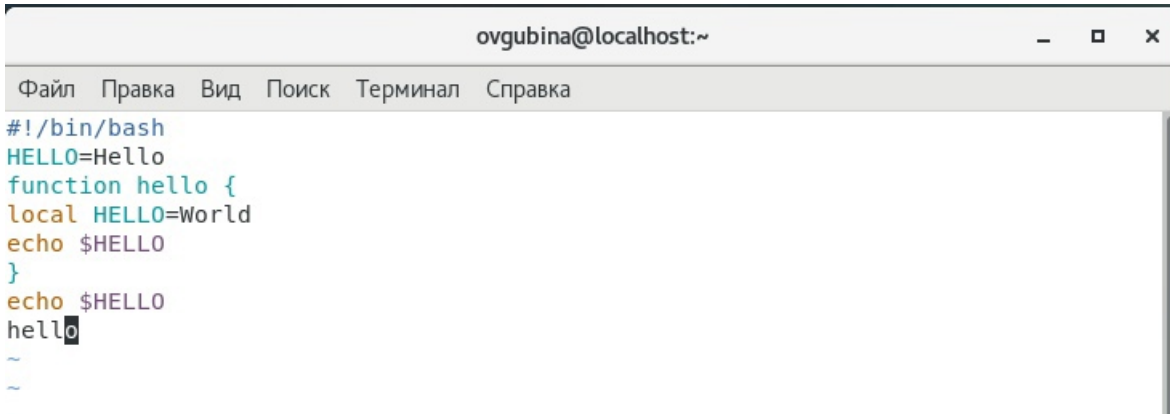
Рисунок 12: ввод local:



```
ovgubina@localhost:~
Файл  Правка  Вид  Поиск  Терминал  Справка
#!/bin/bash
HELLO=Hello
function hello {
local HELLO=World
echo $HELLO
}
echo $HELLO
hello
~
~
~
~
~
~
~
~
~
~
-- ВСТАВКА --                                4,7                Весь
```

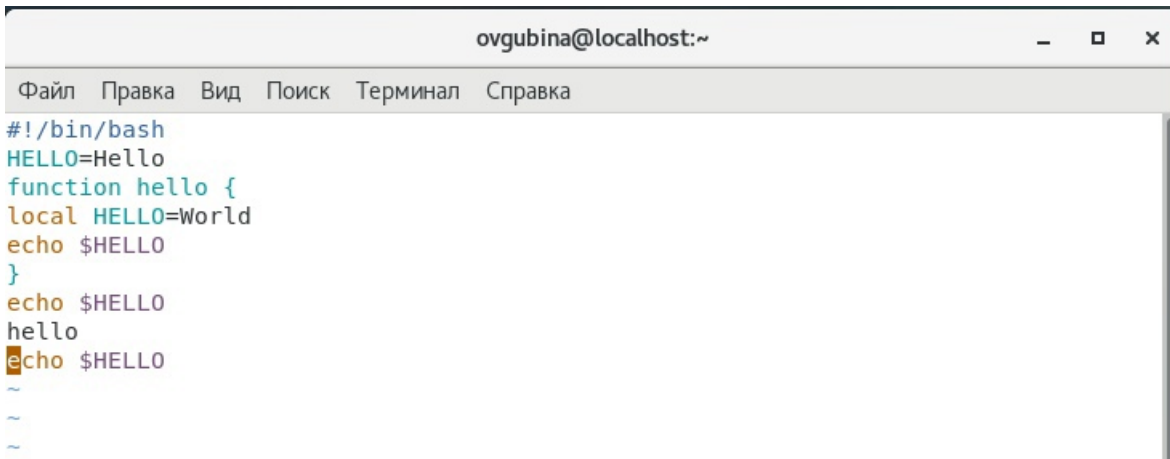
6. Установим курсор на последней строке файла. Вставим после неё строку, содержащую следующий текст: echo \$HELLO. Для этого сначала установим курсор на строке echo \$HELLO и скопируем ее командой `Y`, которая копирует строку, после клавишами стрелок перейдем на последнюю строку, а командой `$` перейдем в ее конец (рисунки 13), обратимся к команде `p`, позволяющей вставить текст из буфера обмена после курсора. Таким образом мы скопировали строку echo \$HELLO в конец текста (рисунки 14).

Рисунок 13: курсор на конец последней строки:



```
ovgubina@localhost:~
Файл  Правка  Вид  Поиск  Терминал  Справка
#!/bin/bash
HELLO=Hello
function hello {
local HELLO=World
echo $HELLO
}
echo $HELLO
hello
~
~
~
```

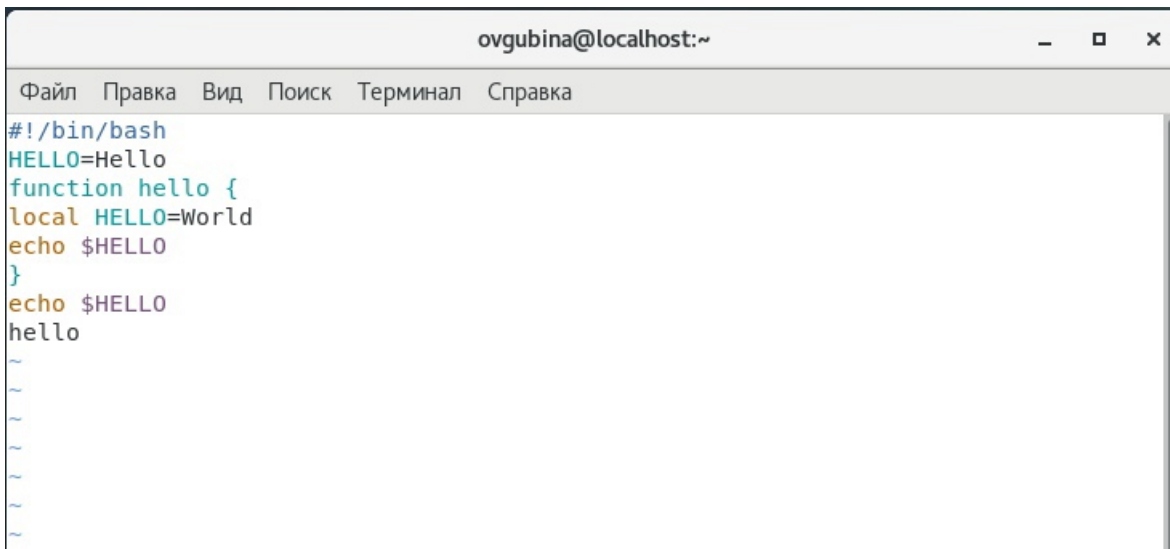
Рисунок 14: копирование-вставка предпоследней строки:



```
ovgubina@localhost:~
Файл  Правка  Вид  Поиск  Терминал  Справка
#!/bin/bash
HELLO=Hello
function hello {
local HELLO=World
echo $HELLO
}
echo $HELLO
hello
echo $HELLO
~
~
~
```

- 7. Нажмем Esc для перехода в командный режим (хотя мы из него и не выходили).
- 8. Удалим последнюю строку (рисунки 15) - наведем на нее курсор и вызовем команду `dd` - удаление строки в буфер.

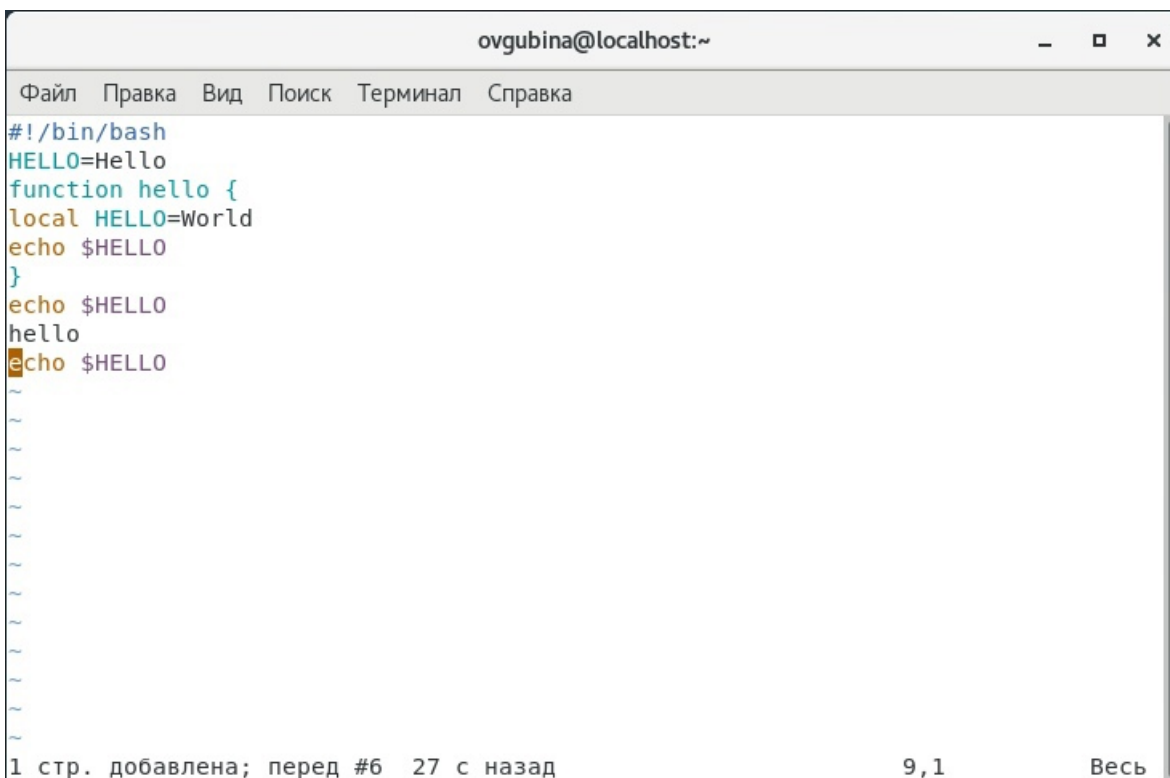
Рисунок 15: удаление последней строки:



```
ovgubina@localhost:~  
Файл Правка Вид Поиск Терминал Справка  
#!/bin/bash  
HELLO=Hello  
function hello {  
  local HELLO=World  
  echo $HELLO  
}  
echo $HELLO  
hello  
~  
~  
~  
~  
~  
~
```

9. Введем команду отмены изменений и для отмены последней команды.Видим, что изменение было успешно удалено - удаленная строка вернулась на место(рисунок 16).

Рисунок 16: отмена последнего действия:



```
ovgubina@localhost:~  
Файл Правка Вид Поиск Терминал Справка  
#!/bin/bash  
HELLO=Hello  
function hello {  
  local HELLO=World  
  echo $HELLO  
}  
echo $HELLO  
hello  
echo $HELLO  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
1 стр. добавлена; перед #6 27 с назад 9,1 Весь
```

10. Введем символ : для перехода в режим последней строки. Запишите произведённые изменения и выйдите из vi. Для этого сначала укажем w и нажмем Enter для записи. После этого наберем :q для выхода из редактора. Или же сразу :wq (рисунок 17) - комбинация, позволяющая поочередно за один раз сначала сохранить изменения, а потом выйти. Ну и, конечно же, Enter. Видим, что мы вернулись к привычной командной строке (рисунок 18).

Рисунок 17: сохранение и выход через режим последней строки:

