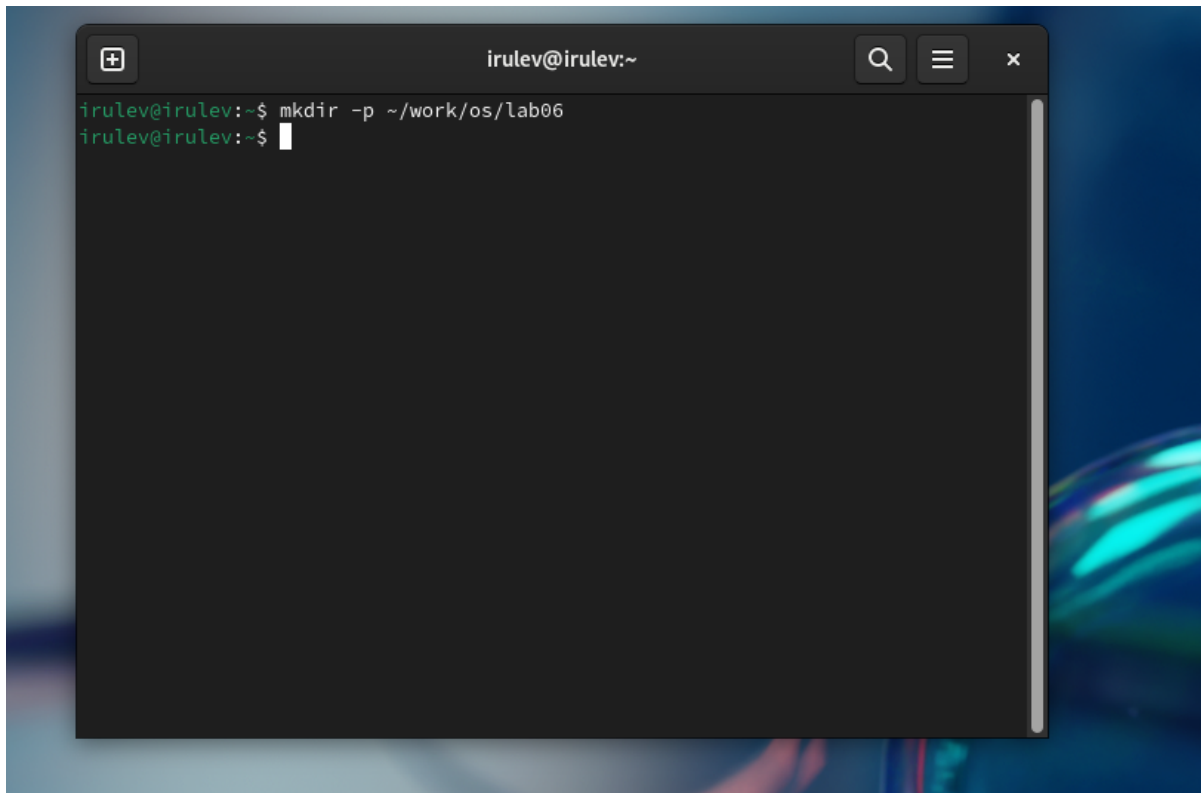


Цель работы

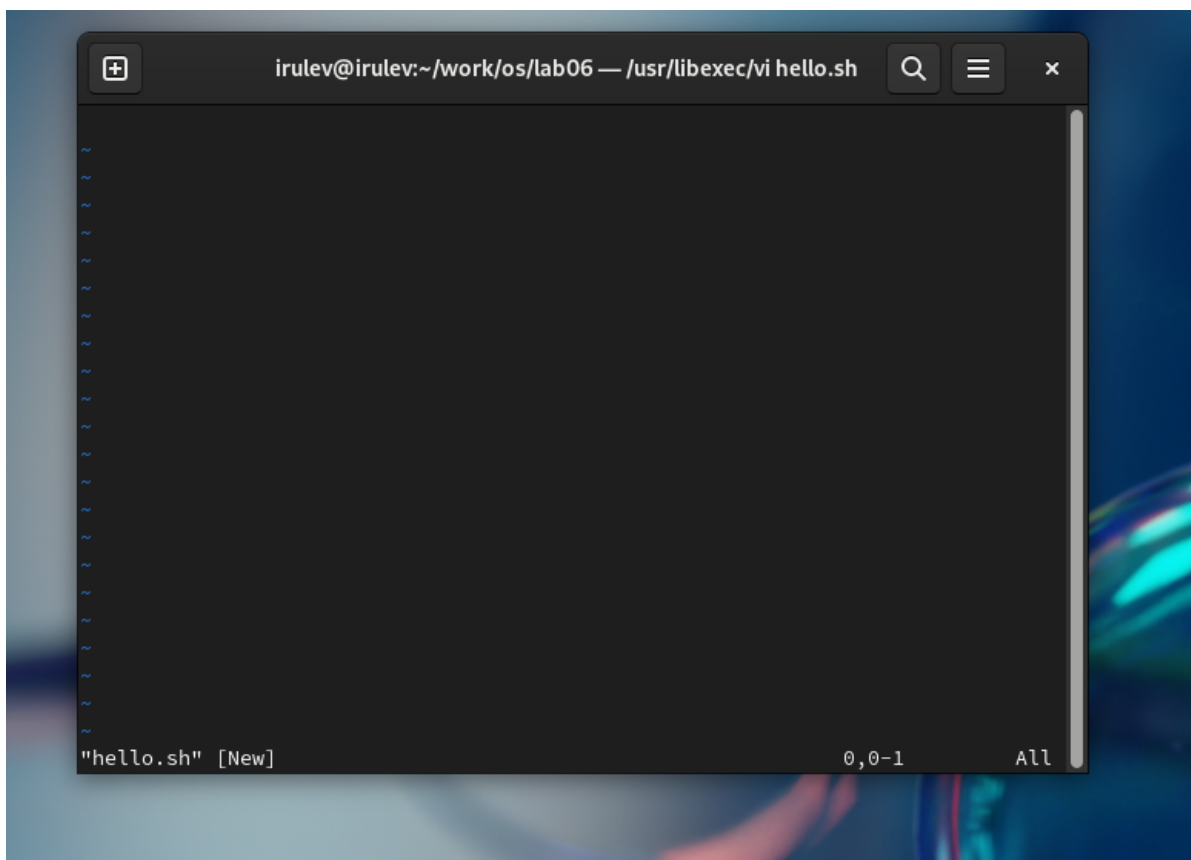
Познакомиться с операционной системой Linux. Получить практические навыки работы с редактором vi, установленным по умолчанию практически во всех дистрибутивах.

Выполнение лабораторной работы

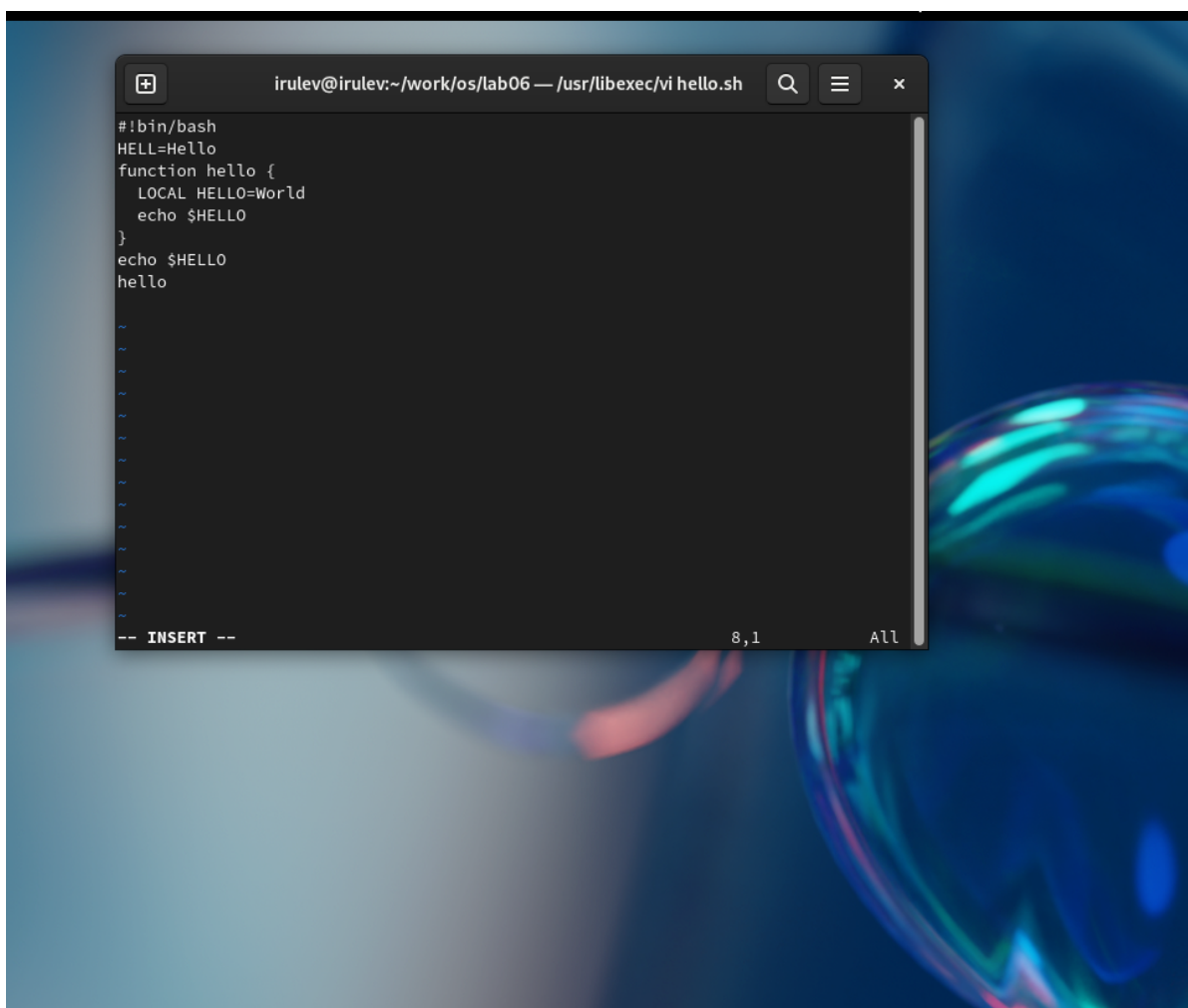
Создал каталог с именем ~/work/os/lab06.



Перешел во вновь созданный каталог, вызвал vi и создал файл hello.sh.



Нажал клавишу i и ввел текст.

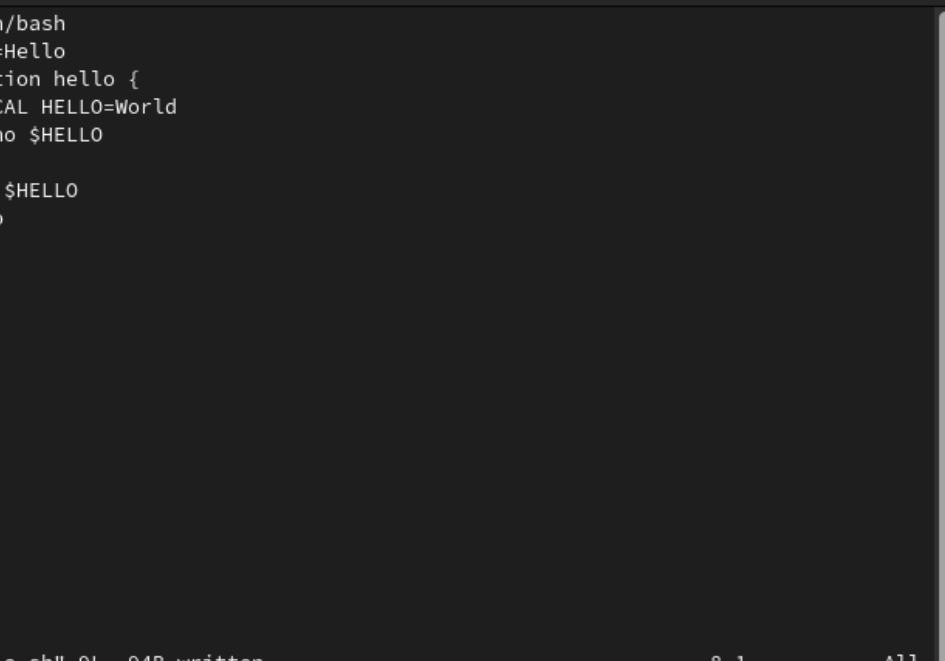


Нажал клавишу Esc для перехода в командный режим после завершения ввода текста.

A screenshot of a terminal window titled "irulev@irulev:~/work/os/lab06 — /usr/libexec/vi hello.sh". The terminal displays a shell script:

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

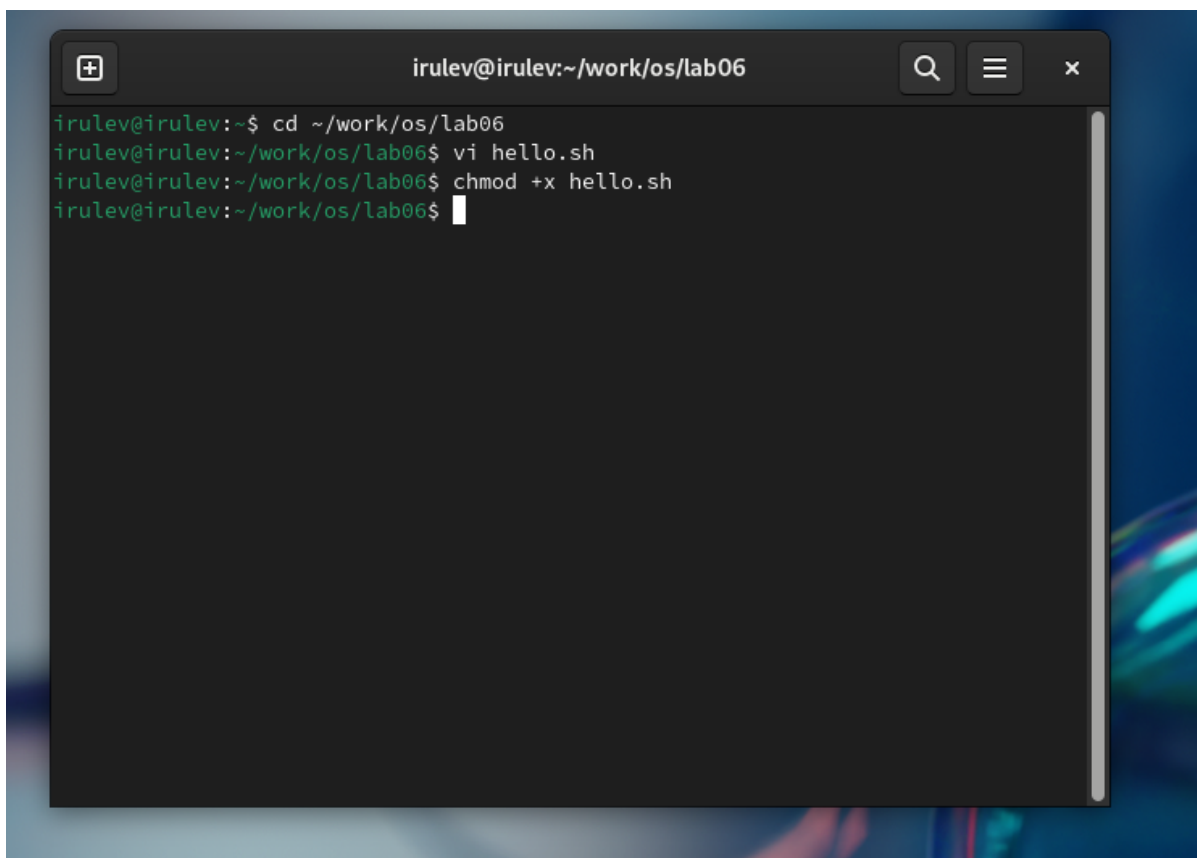
The script has been executed multiple times, as indicated by the repeated tilde (~) prompts. A red arrow points to the prompt at the bottom of the terminal.



The screenshot shows a terminal window with a dark background. The title bar at the top reads "irulev@irulev:~/work/os/lab06 — /usr/libexec/vi hello.sh". The terminal content shows the execution of a shell script:

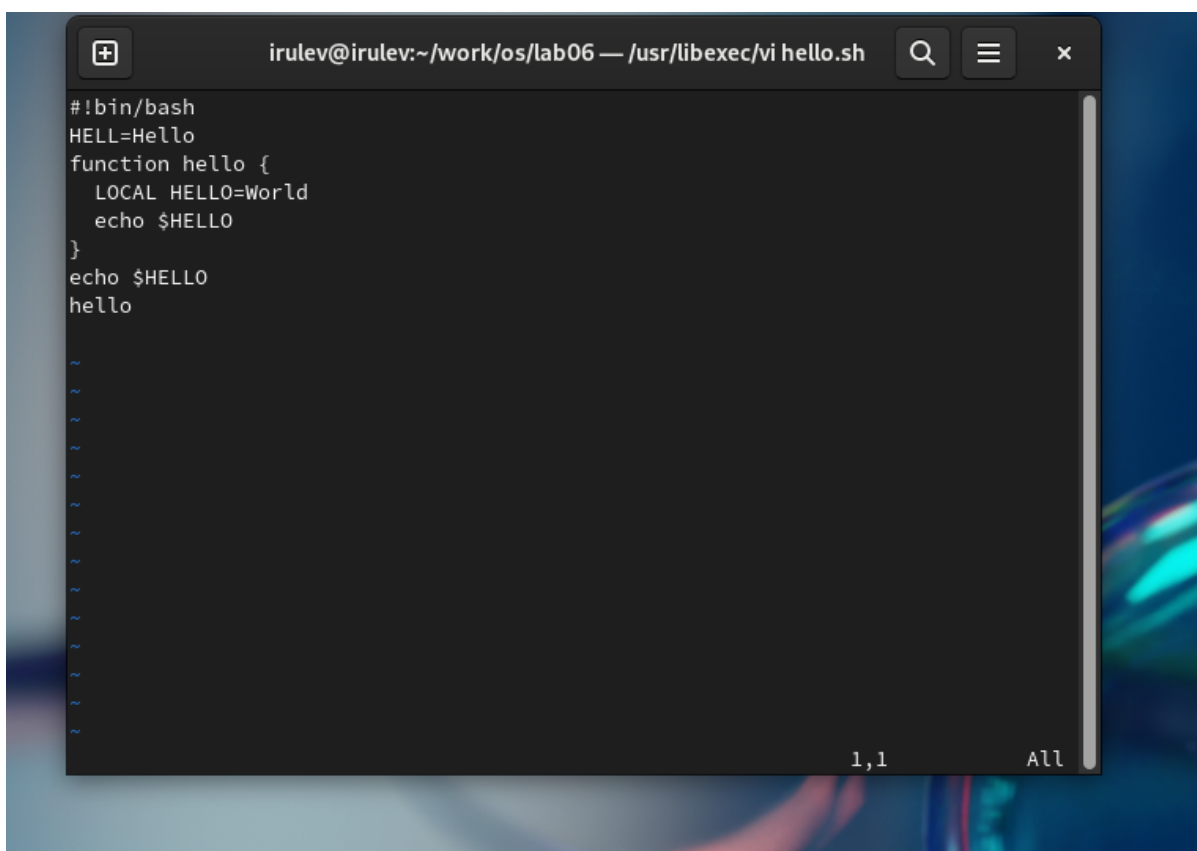
```
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
```

Below the script execution, there are several tilde (~) characters, likely representing the output of the script. At the bottom of the terminal, a status bar indicates: "hello.sh" 9L, 94B written 8,1 All.

A terminal window titled 'irulev@irulev:~/work/os/lab06'. The user has executed the following commands: 'cd ~/work/os/lab06', 'vi hello.sh', and 'chmod +x hello.sh'. The cursor is now at the prompt 'irulev@irulev:~/work/os/lab06\$'.

```
irulev@irulev:~$ cd ~/work/os/lab06
irulev@irulev:~/work/os/lab06$ vi hello.sh
irulev@irulev:~/work/os/lab06$ chmod +x hello.sh
irulev@irulev:~/work/os/lab06$
```

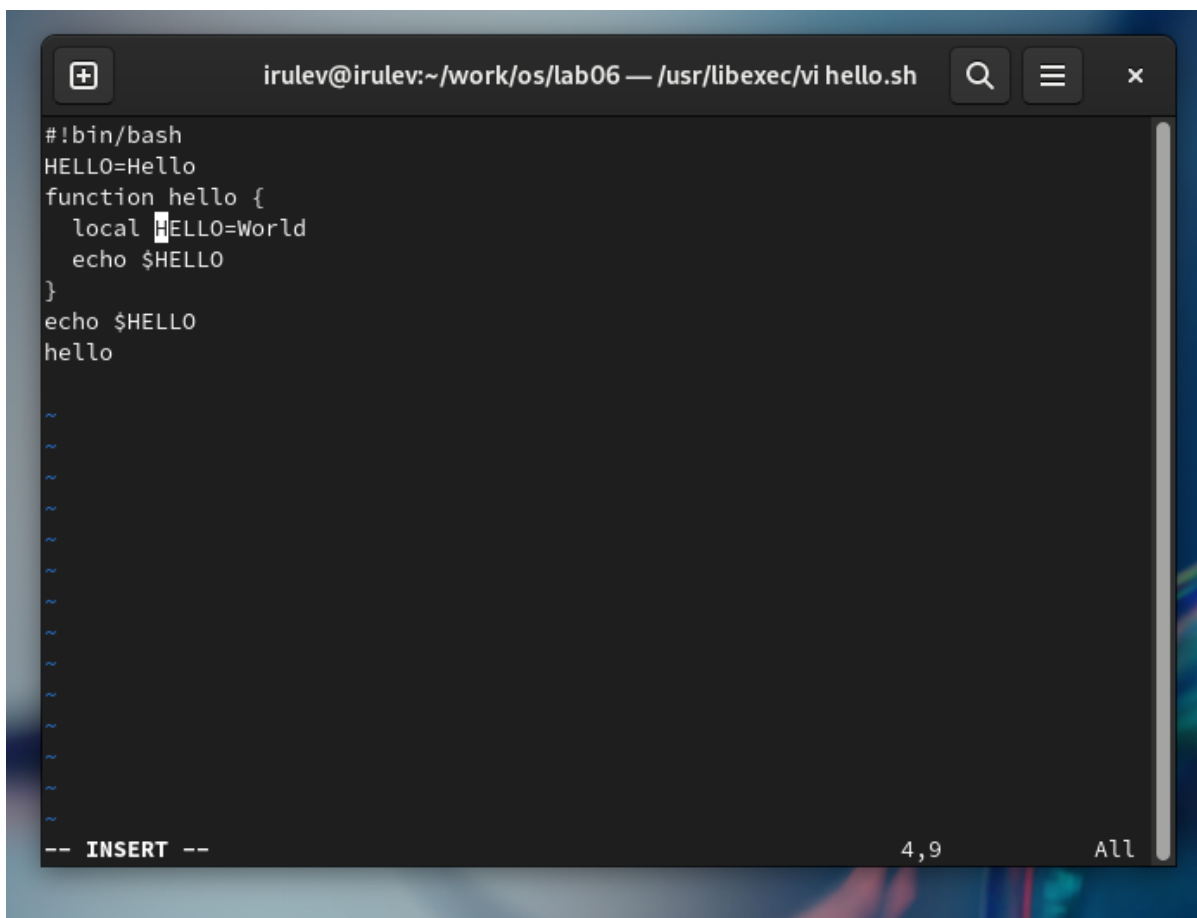
Вызвал vi на редактирование файла

The terminal window now shows the contents of 'hello.sh' being edited in the vi editor. The title bar includes the file path: 'irulev@irulev:~/work/os/lab06 — /usr/libexec/vi hello.sh'. The script content is as follows:

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

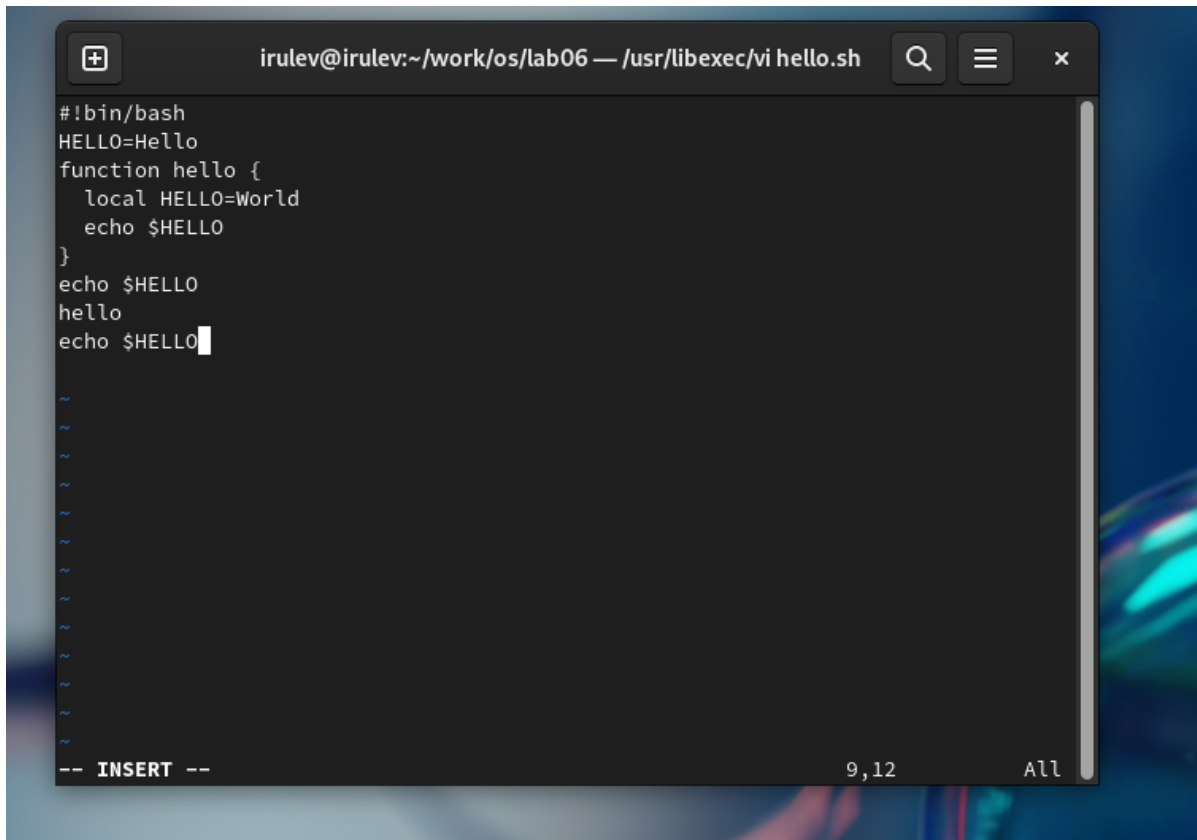
Below the script content, there are several tilde (~) characters representing empty lines. At the bottom right, the status bar shows '1,1' and 'All', indicating the cursor is at the first line and first column in insert mode.

Установил курсор в конец слова HELL второй строки. Перешел в режим вставки и заменил на HELLO. Нажал Esc для возврата в командный режим.



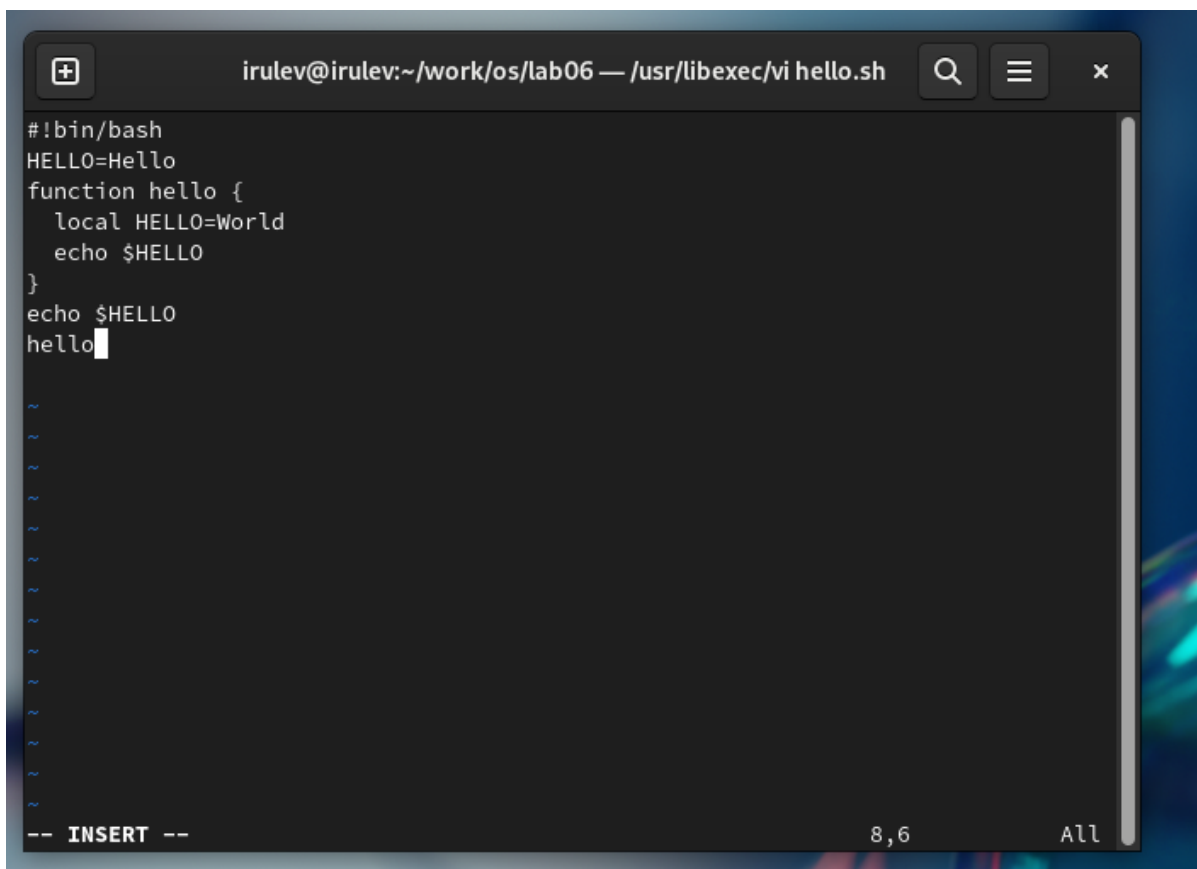
```
irulev@irulev:~/work/os/lab06 — /usr/libexec/vi hello.sh
#!/bin/bash
HELLO=Hello
function hello {
    local HELLO=World
    echo $HELLO
}
echo $HELLO
hello
~
~
~
~
~
~
~
~
~
~
-- INSERT -- 4,9 All
```

Установите курсор на последней строке файла. Вставьте после неё строку, содержащую следующий текст: echo \$HELLO.



```
irulev@irulev:~/work/os/lab06 — /usr/libexec/vi hello.sh
#!/bin/bash
HELLO=Hello
function hello {
    local HELLO=World
    echo $HELLO
}
echo $HELLO
hello
echo $HELLO
~
~
~
~
~
~
~
~
~
~
-- INSERT -- 9,12 All
```

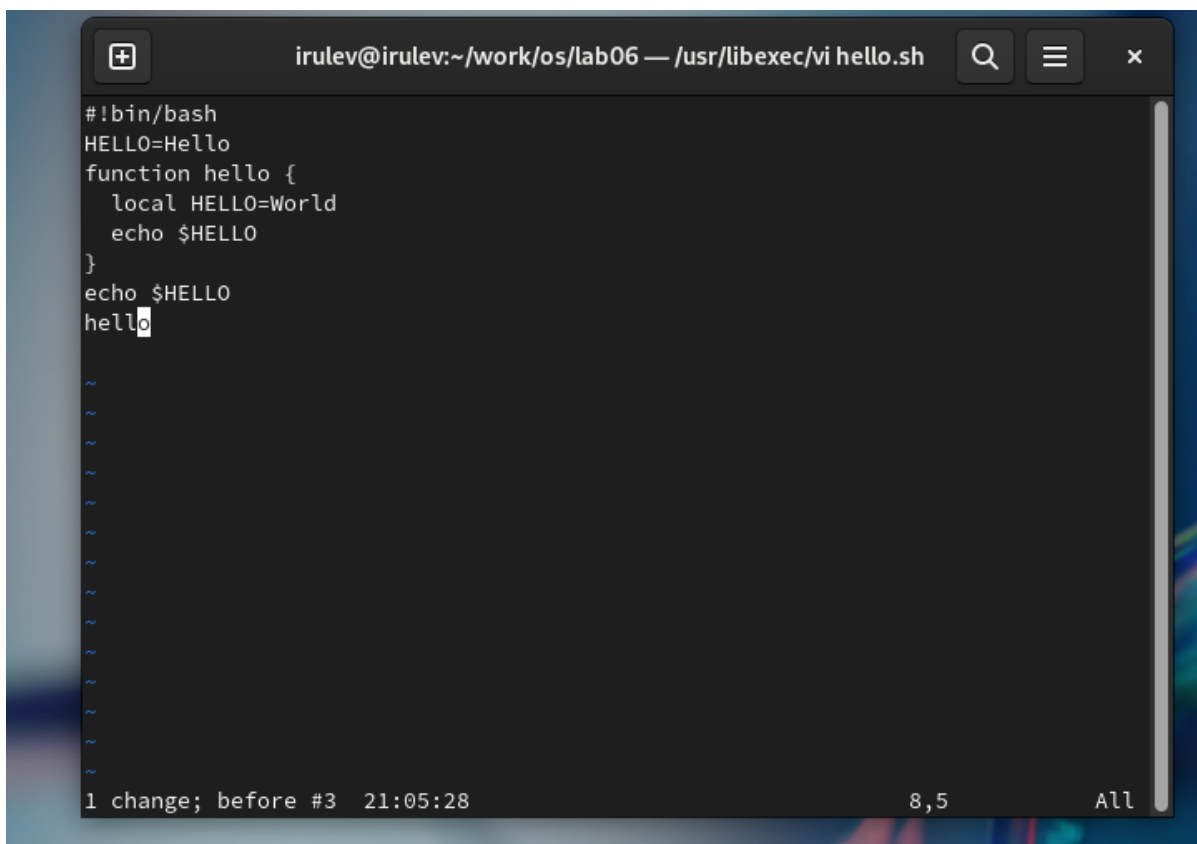
Удалил последнюю строку.



```
irulev@irulev:~/work/os/lab06 — /usr/libexec/vi hello.sh
#!/bin/bash
HELLO=Hello
function hello {
    local HELLO=World
    echo $HELLO
}
echo $HELLO
hello
```

-- INSERT -- 8,6 All

Введите команду отмены изменений и для отмены последней команды.



```
irulev@irulev:~/work/os/lab06 — /usr/libexec/vi hello.sh
#!/bin/bash
HELLO=Hello
function hello {
    local HELLO=World
    echo $HELLO
}
echo $HELLO
hello
```

1 change; before #3 21:05:28 8,5 All

Записал произведённые изменения и вышел из vi.

Выводы

При выполнении данной работы я ознакомился с текстовым редактором vi (Visual display editor) и его функционалом.

Ответы на контрольные вопросы

1. Редактор vi имеет несколько режимов работы:

- **Command mode:** Режим команд, в котором можно выполнять команды для перемещения по тексту, удаления, копирования и других операций над текстом.
- **Insert mode:** Режим вставки, в котором можно вводить текст.
- **Line mode (Ex mode):** Режим редактирования команд, в котором можно выполнять расширенные операции, такие как сохранение файла, поиск и замена, управление настройками и т. д.

2. Для выхода из редактора vi, не сохраняя изменения, в командном режиме следует нажать клавиши `:q!` и затем Enter.

3. Команды позиционирования в командном режиме включают:

- **h:** Переместить курсор влево.
- **j:** Переместить курсор вниз.
- **k:** Переместить курсор вверх.
- **l:** Переместить курсор вправо.
- **0:** Переместить курсор в начало строки.
- **^:** Переместить курсор в начало строки (игнорируя пробелы).
- **\$:** Переместить курсор в конец строки.
- **G:** Переместить курсор в конец файла.
- **gg:** Переместить курсор в начало файла.

4. Для редактора vi "словом" является последовательность символов, разделенных пробелами или другими символами пунктуации.

5. Для перемещения в начало файла используйте команду `gg`, а для перемещения в конец файла - `G`.

6. Основные группы команд редактирования включают в себя:

- **Вставка и удаление текста:** Команды для вставки и удаления символов, строк и блоков текста.
- **Копирование и перемещение текста:** Команды для копирования и перемещения текста из одного места в другое.
- **Поиск и замена:** Команды для поиска и замены текста в файле.
- **Отмена и повторение:** Команды для отмены предыдущих действий и их повторения.
- **Сохранение и выход:** Команды для сохранения изменений и выхода из редактора.

7. Для заполнения строки символами `$` можно использовать команду `A`, чтобы перейти в конец строки, а затем нажать клавишу `Esc` для перехода в режим команд и ввести нужное количество символов `$`.

8. Для отмены некорректного действия в редакторе vi можно использовать команду `u` в режиме команд, чтобы отменить последнее действие.

9. Основные группы команд режима последней строки (Ex mode) включают:

- **Сохранение и выход:** Команды для сохранения изменений и выхода из файла.
- **Поиск и замена:** Команды для поиска и замены текста в файле.

- **Управление настройками:** Команды для управления настройками редактора, такими как отображение номеров строк, установка отступов и т.д.

10. Для определения позиции, в которой заканчивается строка, можно использовать команду `$` в режиме команд.

11. В редакторе vi существует множество опций, позволяющих настраивать его поведение и внешний вид. Некоторые из них:

- `number` - отображение номеров строк
- `syntax` - подсветка синтаксиса
- `autoindent` - автоматическое выравнивание строк
- `tabstop` - размер табуляции
- `wrap` - перенос строк

Для получения полного списка опций и их назначения можно использовать команду `:set all`.

12. Режим работы в редакторе vi можно определить по тому, какие команды он ожидает от пользователя и как ведет себя курсор. Основные режимы:

- **Command mode (режим команд):** В этом режиме можно выполнять команды, такие как удаление, копирование, вставка, перемещение по тексту.
- **Insert mode (режим вставки):** В этом режиме можно вводить текст.
- **Visual mode (визуальный режим):** В этом режиме можно выделять текст для выполнения над ним операций.

Обычно, когда вы открываете vi, вы начинаете в режиме команд.

13. Пользователь может переключаться между режимами с помощью соответствующих команд (например, нажатие клавиши `i` переводит в режим вставки, а нажатие клавиши `Esc` вернет обратно в режим команд).