

Отчёт по лабораторной работе №10

Операционные системы

Луангсуваннавонг Сайпхачан

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Создание нового файла с использованием vi	8
4.2	Редактирование существующего файла	10
5	Выводы	15
6	Ответы на контрольные вопросы	16
	Список литературы	19

Список иллюстраций

4.1	Создание каталогов	8
4.2	Вызов vi	8
4.3	Ввод текстового кода	9
4.4	Сохранение файла и выход из него	9
4.5	Добавление разрешения на выполнение файла	10
4.6	Вызов vi	10
4.7	Редактирование текстового кода	10
4.8	Редактирование текстового кода	11
4.9	Добавление текстового кода	12
4.10	Удаление текстового кода	13
4.11	Отмена последней команды	13
4.12	Сохранение файла и выход из него	14

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. Ознакомиться с теоретическим материалом.
2. Ознакомиться с редактором vi.
3. Выполнить упражнения, используя команды 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 (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

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:

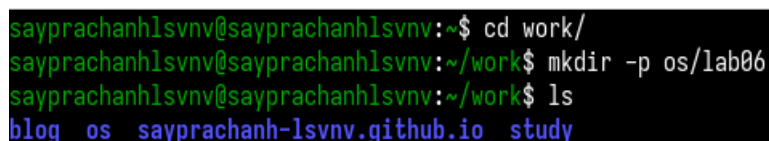
- набрать символы wq, если перед выходом из редактора требуется записать изменения в файл;
- набрать символ q (или q!), если требуется выйти из редактора без сохранения.

Замечание. Следует помнить, что vi различает прописные и строчные буквы при наб

4 Выполнение лабораторной работы

4.1 Создание нового файла с использованием vi

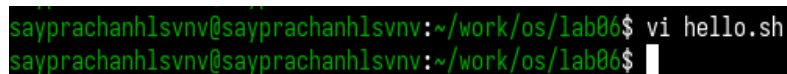
Сначала я создаю новый каталог в каталоге work и называю его os, затем создаю вложенный подкаталог lab06, после чего перехожу в созданный каталог (рис. 4.1)



```
sayprachanhlsnv@sayprachanhlsnv:~$ cd work/  
sayprachanhlsnv@sayprachanhlsnv:~/work$ mkdir -p os/lab06  
sayprachanhlsnv@sayprachanhlsnv:~/work$ ls  
blog os sayprachanhlsnv.github.io study
```

Рис. 4.1: Создание каталогов

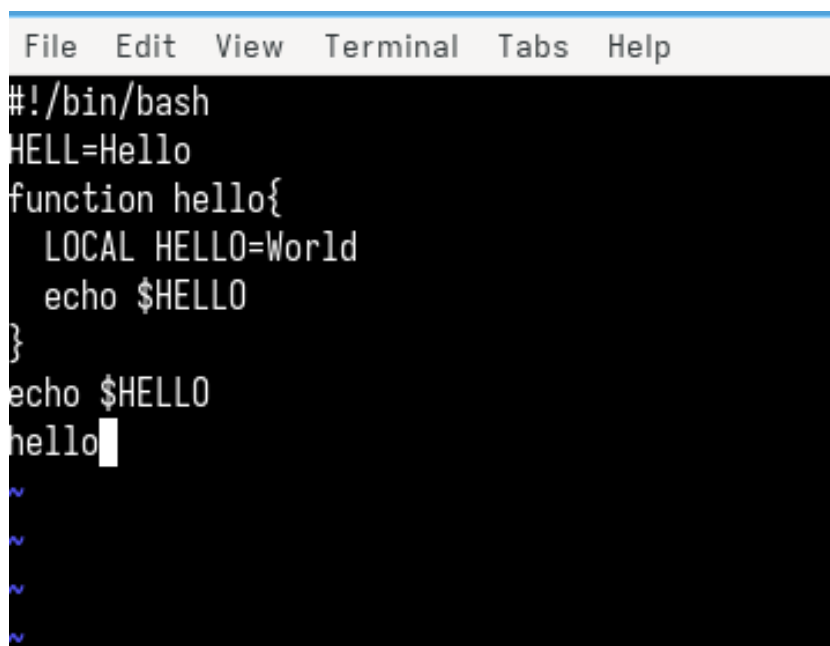
С помощью команды vi я запускаю текстовый редактор vi, затем создаю новый файл, давая ему имя hello.sh (рис. 4.2)



```
sayprachanhlsnv@sayprachanhlsnv:~/work/os/lab06$ vi hello.sh  
sayprachanhlsnv@sayprachanhlsnv:~/work/os/lab06$
```

Рис. 4.2: Вызов vi

Я перехожу в режим вставки в текстовом редакторе vi, затем ввожу bash-код в файл, нажав клавишу i на клавиатуре — это переводит меня из командного режима в режим вставки для ввода текста в файл (рис. 4.3)

A screenshot of a terminal window with a menu bar at the top containing 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal content shows the following commands and output:

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

The cursor is positioned at the end of the 'hello' command on the last line. Below the terminal window, there are four blue tilde (~) symbols.

Рис. 4.3: Ввод текстового кода

Затем я возвращаюсь в командный режим, нажав клавишу ESC на клавиатуре, затем ввожу символ :, за которым следуют буквы w и q, чтобы сохранить изменения в файле и выйти из него (рис. 4.4)

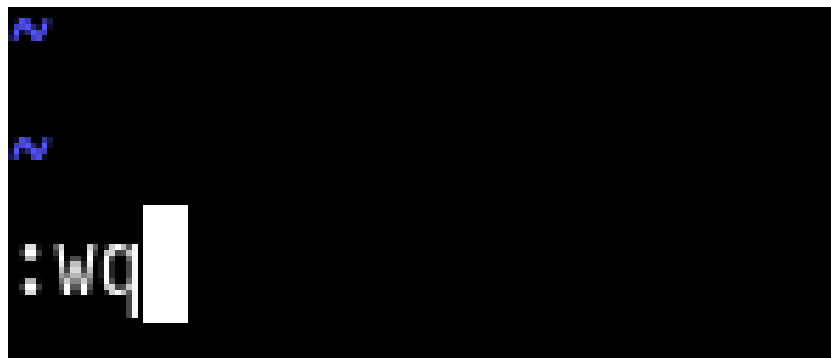
A screenshot of a terminal window showing the command ':wq' being entered. The cursor is at the end of the command. Above the terminal window, there are two blue tilde (~) symbols.

Рис. 4.4: Сохранение файла и выход из него

С помощью команды `chmod` я изменяю права доступа к файлу `hello.sh`, добавляя право на выполнение, чтобы сделать файл исполняемым (рис. 4.5)

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

Рис. 4.5: Добавление разрешения на выполнение файла

4.2 Редактирование существующего файла

Я снова открываю файл `hello.sh` в текстовом редакторе `vi` (рис. 4.6)

```
sayprachanhlsnv@sayprachanhlsnv:~$ vi ~/work/os/lab06/hello.sh
```

Рис. 4.6: Вызов `vi`

Я перемещаю курсор с помощью стрелок к концу слова “HELL” во второй строке, перехожу в режим вставки и изменяю слово на “HELLO”, затем нажимаю `ESC`, чтобы вернуться в командный режим (рис. 4.7)

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

Рис. 4.7: Редактирование текстового кода

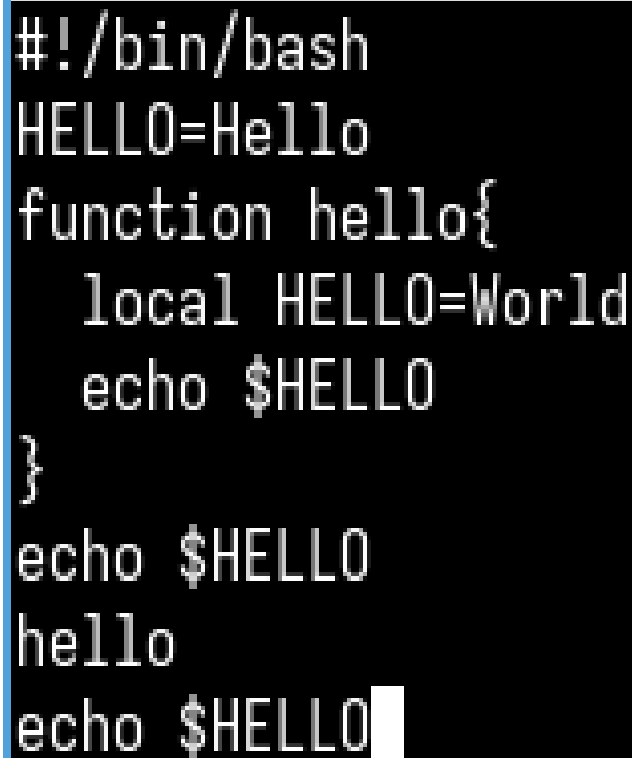
Перехожу на следующую строку и меняю слово “LOCAL” на “local” в режиме вставки. После завершения редактирования возвращаюсь в командный режим (рис. 4.8)



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

Рис. 4.8: Редактирование текстового кода

В командном режиме, нажав клавишу G, я перемещаюсь в конец файла, затем перехожу в режим вставки и добавляю новую строку в файл (echo \$HELLO). После завершения ввода возвращаюсь в командный режим (рис. 4.9)

A terminal window with a black background and white text. The text is a shell script. The first line is a shebang. The second line is an assignment. The third line is a function definition with a local variable. The fourth line is an echo command inside the function. The fifth line is the closing brace of the function. The sixth line is an echo command outside the function. The seventh line is a function call. The eighth line is an echo command. A white cursor is at the end of the last line.

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

Рис. 4.9: Добавление текстового кода

В командном режиме, нажав клавишу d, я удаляю последнюю строку файла, предварительно перейдя на неё с помощью стрелок (рис. 4.10)

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

Рис. 4.10: Удаление текстового кода

Нажав клавишу `u` в командном режиме, я отменяю последнее действие (удаление предыдущей последней строки) (рис. 4.11)

```
echo $HELLO
}
echo $HELLO
hello
```

Рис. 4.11: Отмена последней команды

Я ввожу символ `:`, затем буквы `w` и `q`, чтобы сохранить изменения и выйти из файла (рис. 4.12)

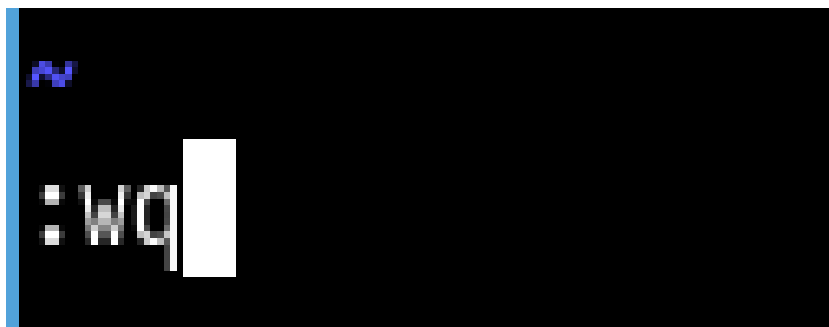


Рис. 4.12: Сохранение файла и выход из него

5 Выводы

Во время этой лабораторной работы я познакомился с операционной системой Linux. Получил практические навыки работы с редактором vi, установленным по умолчанию практически во всех дистрибутивах

6 Ответы на контрольные вопросы

1. Дайте краткую характеристику режимам работы редактора vi.

Редактор vi работает в три режима. Первый — командный режим (Command Mode), где можно двигаться по тексту и использовать команды. Второй — режим вставки (Insert Mode), он нужен чтобы писать текст (включается с i, a, o). Третий — это режим последней строки (Last line Mode), когда пишем : и используем команды для сохранения, выхода, поиска.

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

Надо нажать Esc, потом написать :q! и нажать Enter. Это выход без сохранить. Ещё можно использовать ZQ.

3. Назовите и дайте краткую характеристику командам позиционирования.

h, j, k, l — двигаться по символам (влево, вниз, вверх, вправо).

O, ^, \$ — начало строки, первый символ, конец строки.

w, b, e — переход по словам.

gg, G, nG — начало файла, конец, и строка номер n.

4. Что для редактора vi является словом?

Слово — это буквы, цифры и подчёркивание. Но если использовать W, B, тогда считается как любые символы до пробела. Например, foo-bar — может быть одно слово или два, зависит от команды.

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

Чтобы в начало файла: команда gg или 1G.

Чтобы в конец: команда G.

А если нужно точная строка: пишем :номер.

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

Вставка: i, a, o — для писать текст.

Удаление: x — символ, dd — строка, dw — слово.

Копия и вставка: yy, uw, p.

Замена: r — символ, cw — слово, R — замена режим.

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

Сначала в командный режим (нажать Esc), потом написать 80i, Esc. : s/*/\$\$\$\$\$\$\$/, тогда строка заменится на \$.

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

u — отмена последнего действия.

Ctrl+r — вернуть обратно.

U — отмена изменений в строке.

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

Файл: :w — сохранить, :q — выйти, :q! — выйти без сохранить.

Поиск и замена: :/текст, :%s/старый/новый/g.

Настройки: :set number, :set ignorecase.

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

Можно нажать \$, курсор покажет конец. Если включить :set ruler, тогда всегда видно позиция курсора.

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

Показать все опции: :set all.

Справка по опции: :help 'имя_опции'.

Примеры: :set number, :set nowrap. Опций очень много, они меняют поведение vi.

12. Как определить режим работы редактора vi?

Командный режим — когда нет текста внизу.

Режим вставки — написано – INSERT –.

Режим последней строки — видно : или / внизу.

13. Постройте граф взаимосвязи режимов работы редактора vi.

Командный режим (Command mode)

- i, a, o → Режим вставки (Insert) → Esc → Командный режим

|- :, / → Режим последней строки (Ex) → Enter → Командный режим

Список литературы

Лабораторная работа №10