

Отчёт по лабораторной работе №10

Дисциплина: Операционные системы

Верниковская Екатерина Андреевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Задание 1. Создание нового файла с использованием vi	7
3.2	Задание 2. Редактирование существующего файла	10
4	Ответы на контрольные вопросы	17
5	Выводы	21
6	Список литературы	22

Список иллюстраций

3.1	Создание каталога	7
3.2	Переход в созданный каталог	7
3.3	Вызов vi (1)	7
3.4	Открытый редактор vi	8
3.5	Режим вставки	8
3.6	Введённый текст	9
3.7	Переход в командный режим	9
3.8	Переход в режим последней строки	9
3.9	Завершение работы (1)	10
3.10	Добавление прав на исполнение	10
3.11	Вызов vi (2)	10
3.12	Открытый файл	11
3.13	Курсор в конце слова HELL	11
3.14	Замена HELL на HELLO	12
3.15	Без слова LOCAL	12
3.16	Замена на local	13
3.17	Вставка echo \$HELLO	14
3.18	Удаление последней строки	14
3.19	Отмена последнего действия	15
3.20	Завершение работы (2)	16
4.1	Граф взаимосвязи режимов работы редактора vi	20

Список таблиц

1 Цель работы

Ознакомиться 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 Выполнение лабораторной работы

3.1 Задание 1. Создание нового файла с использованием vi

Создаём каталог с именем ~/work/os/lab10 (рис. 3.1)

```
[eavernikovskaya@eavernikovskaya ~]$ mkdir ~/work/  
[eavernikovskaya@eavernikovskaya ~]$ mkdir ~/work/os/  
[eavernikovskaya@eavernikovskaya ~]$ mkdir ~/work/os/lab10  
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 3.1: Создание каталога

Переходим в только что созданный каталог (рис. 3.2)

```
[eavernikovskaya@eavernikovskaya ~]$ cd ~/work/os/lab10  
[eavernikovskaya@eavernikovskaya lab10]$
```

Рис. 3.2: Переход в созданный каталог

Далее вызываем vi и создаём файл hello.sh командой *vi hello.sh* (рис. 3.3), (рис. 3.4)

```
[eavernikovskaya@eavernikovskaya lab10]$ vi hello.sh  
[eavernikovskaya@eavernikovskaya lab10]$
```

Рис. 3.3: Вызов vi (1)

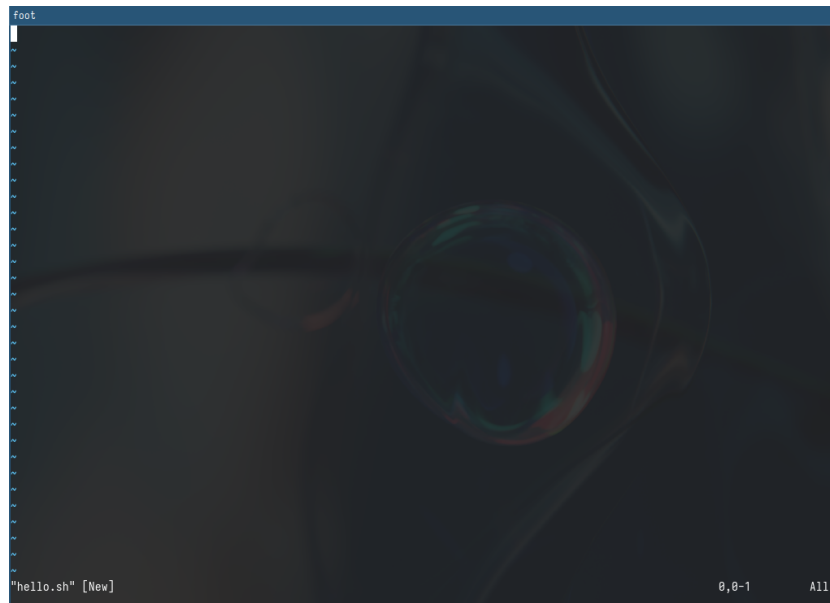


Рис. 3.4: Открытый редактор vi

После нажимаем на кнопку *i* чтобы перейти в режим вставки (рис. 3.5)



Рис. 3.5: Режим вставки

Вводим нужный текст: (рис. 3.6)

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



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

Рис. 3.6: Введённый текст

Далее нажимаем *esc* для перехода в командный режим (рис. 3.7)

```
~
~
~
```

Рис. 3.7: Переход в командный режим

Далее с помощью *:* мы переходим в режим последней строки. Внизу экрана появляется приглашение в виде двоеточия (рис. 3.8)

```
~
~
:
:
```

Рис. 3.8: Переход в режим последней строки

После вводим *w*(записать) и *q*(выйти), а затем нажимаем *enter* для сохранения нашего текста и завершения работы (рис. 3.9)



Рис. 3.9: Завершение работы (1)

Далее делаем файл исполняемым введя *chmod +x hello.sh* (рис. 3.10)

```
[eavernikovskaya@eavernikovskaya lab10]$ chmod +x hello.sh
[eavernikovskaya@eavernikovskaya lab10]$ ls
hello.sh
[eavernikovskaya@eavernikovskaya lab10]$
```

Рис. 3.10: Добавление прав на исполнение

3.2 Задание 2. Редактирование существующего файла

Вызываем *vi* на редоктирование файла (рис. 3.11), (рис. 3.12)

```
[eavernikovskaya@eavernikovskaya os]$ vi ~/work/os/lab10/hello.sh
```

Рис. 3.11: Вызов *vi* (2)


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

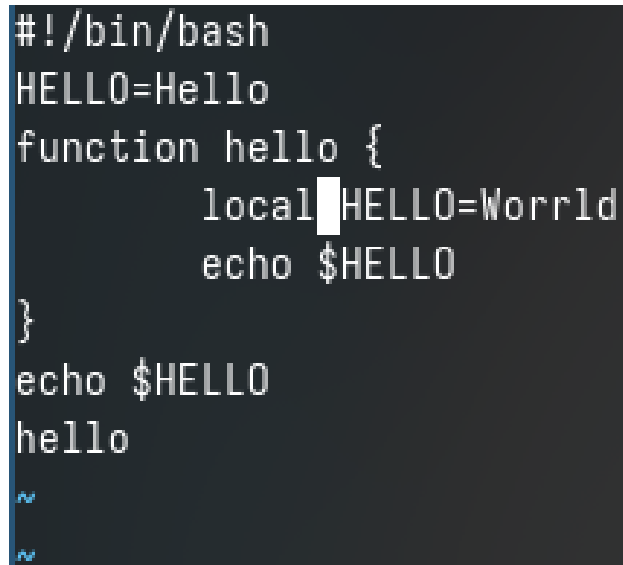
Рис. 3.14: Замена HELL на HELLO

После устанавливаем курсор на четвёртую строку и стираем слово LOCAL (рис. 3.15)

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

Рис. 3.15: Без слова LOCAL

Переходим в режим вставки и набираем слово `local`. После переходим опять в командный режим (рис. 3.16)



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

Рис. 3.16: Замена на `local`

Далее устанавливаем курсор на последней строке файла и вставляем туда следующий текст: `echo $HELLO` (рис. 3.17)

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

Рис. 3.17: Вставка echo \$HELLO

Переходим снова в командный режим. Далее удаляем последнюю строку (рис. 3.18)

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

Рис. 3.18: Удаление последней строки

После вводим команду отмены изменений нажав на *u* (рис. 3.19)

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

Рис. 3.19: Отмена последнего действия

Вводим : для перехода в режим последней строки. Записываем произведённые изменения и выходим из vi (рис. 3.20)



Рис. 3.20: Завершение работы (2)

4 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 (или командной) строки — используется для записи изменений в файл и выхода из редактора.

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

Можно нажимать символ q (или q!), если требуется выйти из редактора без сохранения.

3. Назовите и дайте краткую характеристику командам позиционирования.

- 0 (ноль) — переход в начало строки;
- \$ — переход в конец строки;
- G — переход в конец файла;
- n G — переход на строку с номером n.

4. Что для редактора vi является словом?

Редактор vi предполагает, что слово - это строка символов, которая может включать в себя буквы, цифры и символы подчеркивания.

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

С помощью G — переход в конец файла.

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; n i — 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; d w — удалить одно слово в буфер; d \$ — удалить в буфер текст от курсора до конца строки; d 0 — удалить в буфер текст от начала строки до позиции курсора; d d — удалить в буфер одну строку; n d d — 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; n Y — скопировать n строк в буфер; y w — скопировать слово в буфер.
- Вставка текста из буфера: p — вставить текст из буфера после курсора; P — вставить текст из буфера перед курсором.
- Замена текста: c w — заменить слово; n c w — заменить n слов; c \$ — заменить текст от курсора до конца строки; r — заменить слово; R — заменить текст.
- Поиск текста: / текст — произвести поиск вперёд по тексту указанной строки символов текст; ? текст — произвести поиск назад по тексту указанной строки символов текст.

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

Перейти в режим вставки.

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

С помощью `u` — отменить последнее изменение.

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

Режим последней строки — используется для записи изменений в файл и выхода из редактора.

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

`$` — переход в конец строки.

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

Опции редактора `vi` позволяют настроить рабочую среду. Для задания опций используется команда `set` (в режиме последней строки): `- : set all` — вывести полный список опций; `- : set nu` — вывести номера строк; `- : set list` — вывести невидимые символы; `- : set ic` — не учитывать при поиске, является ли символ прописным или строчным.

12. Как определить режим работы редактора `vi`?

В редакторе `vi` есть два основных режима: командный режим и режим вставки. По умолчанию работа начинается в командном режиме. В режиме вставки клавиатура используется для набора текста. Для выхода в командный режим используется клавиша `Esc` или комбинация `Ctrl + c`.

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

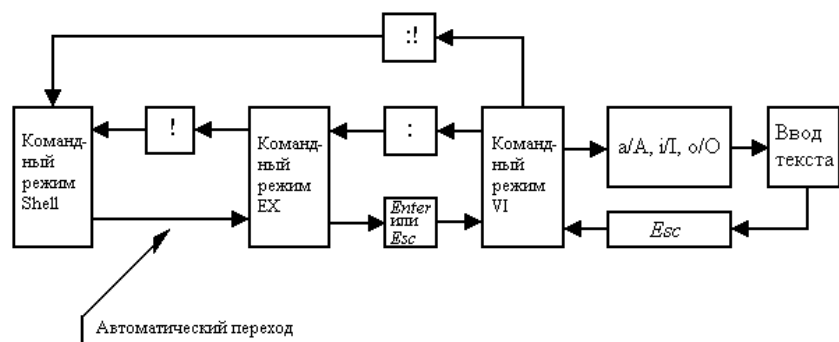


Рис. 4.1: Граф взаимосвязи режимов работы редактора vi

5 Выводы

В ходе выполнения лабораторной работы мы ознакомились с операционной системой Linux а также получили практические навыки работы с редактором vi, установленным по умолчанию практически во всех дистрибутивах.

6 Список литературы

Не пользовалась сайтами.