

Отчёт по лабораторной работе №10

Текстовый редактор vi

Сергей Витальевич Павлюченков

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	11
5	Контрольные вопросы	12

Список иллюстраций

3.1	Создание каталога	7
3.2	Работа с vi	7
3.3	Делаю файл исполняемым	8
3.4	Редактирование слова	8
3.5	Стирание слова	8
3.6	Набирание текста	9
3.7	Вставляю строку	9
3.8	Удаление строки	9
3.9	Отмена последних изменений	10

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 Выполнение лабораторной работы

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

```
[svpavliuchenkov@fedora ~]$ mkdir work/os/lab10
mkdir: невозможно создать каталог «work/os/lab10»: Нет такого файла или каталога
[svpavliuchenkov@fedora ~]$ mkdir work/os
[svpavliuchenkov@fedora ~]$ mkdir work/os/lab10
[svpavliuchenkov@fedora ~]$ cd work/os/lab10/
[svpavliuchenkov@fedora lab10]$
```

Рис. 3.1: Создание каталога

Вызвал vi и создал файл hello.sh, после чего нажал клавишу i и ввел нужный текст(рис. 2).

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

Рис. 3.2: Работа с vi

После закрытия vi делаю файл исполняемым.

```
[svpavliuchenkov@fedora lab10]$ vi hello.sh
[svpavliuchenkov@fedora lab10]$ chmod +x hello.sh
[svpavliuchenkov@fedora lab10]$
```

Рис. 3.3: Делаю файл исполняемым

Вызываю `vi` на редактирование файла и устанавливаю курсор в конец слова `HELL` второй строки. Перехожу в режим вставки и заменяю на `HELLO`. Нажимаю `Esc` для возврата в командный режим.

```
foot
#!/bin/bash
HELLOHello
function hello {
    LOCAL HELLO=world
    echo $HELLO
}
echo $HELLO
hello
```

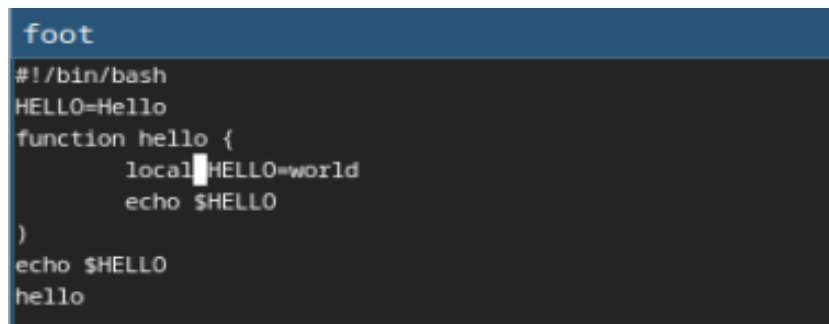
Рис. 3.4: Редактирование слова

Устанавливаю курсор на четвертую строку и стираю слово `LOCAL`

```
foot
#!/bin/bash
HELLO=Hello
function hello {
    HELLO=world
    echo $HELLO
}
echo $HELLO
hello
```

Рис. 3.5: Стирание слова

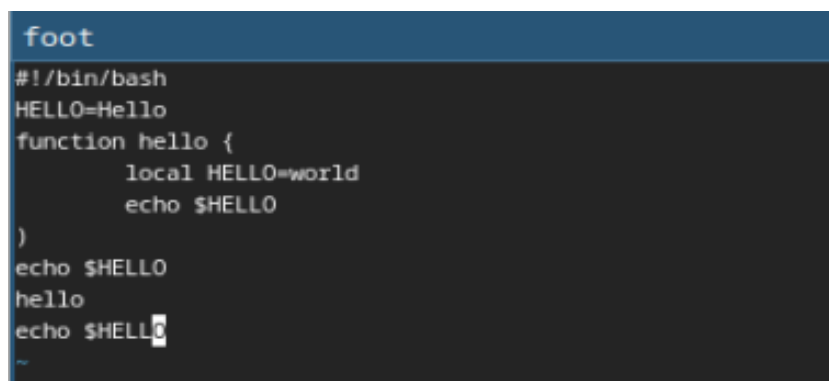
Перехожу в режим вставки и набираю следующий текст: `local`, нажимаю `Esc` для возврата в командный режим.



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

Рис. 3.6: Набирание текста

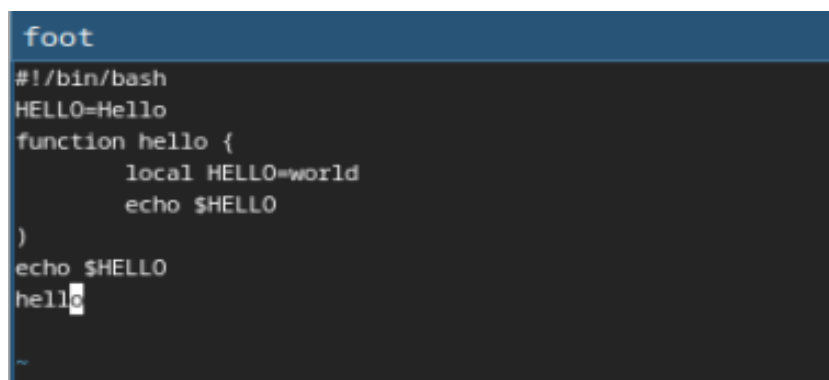
Устанавливаю курсор на последней строке файла. Вставляю после неё строку, содержащую следующий текст: echo \$HELLO.



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

Рис. 3.7: Вставляю строку

Нажимаю Esc для перехода в командный режим. Удаляю последнюю строку.



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

Рис. 3.8: Удаление строки

Ввожу команду отмены изменений и для отмены последней команды.

4 Выводы

Я научился работать с текстовым редактором vi. Повторил знания по работе в терминале линукс.

5 Контрольные вопросы

1. Дайте краткую характеристику режимам работы редактора vi.

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.

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

- набрать символ q (или q!), если требуется выйти из редактора без сохранения.

3. Назовите и дайте краткую характеристику командам позиционирования.

- Команды позиционирования – 0 (ноль) — переход в начало строки; – \$ — переход в конец строки; – G — переход в конец файла; – ☒ G — переход на строку с номером ☒

4. Что для редактора vi является словом?

- Пробелы, знаки препинания и другие непечатные символы разделяют слова в vi. Таким образом, слово в vi - это любая непрерывная последовательность символов

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

- – G — переход в конец файла; – ☒ G — переход на строку с номером ☒. – переход на первую строку - начало

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

8.2.2.1. Вставка текста – a — вставить текст после курсора; – A — вставить текст в конец строки; – i — вставить текст перед курсором; – ☒ i — вставить текст ☒ раз; – I — вставить текст в начало строки. 8.2.2.2. Вставка строки – o — вставить строку под курсором; – O — вставить строку над курсором. 8.2.2.3. 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; – ☒ d d — удалить в буфер ☒ строк. 8.2.2.4. Отмена и повтор произведённых изменений – u — отменить последнее изменение; – . — повторить последнее изменение. 8.2.2.5. Копирование текста в буфер – Y — скопировать строку в буфер; – ☒ Y — скопировать ☒ строк в буфер; – y w — скопировать слово в буфер. 8.2.2.6. Вставка текста из буфера – p — вставить текст из буфера после курсора; – P — вставить текст из буфера перед курсором. 8.2.2.7. Замена текста – c w — заменить слово; – ☒ c w — заменить ☒ слов; – c \$ — заменить текст от курсора до конца строки; – r — заменить слово; – R — заменить текст. 8.2.2.8. 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?

– ☒ i — вставить текст ☒ раз, где i - \$

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

Перейти в командный режим и ввести команду для отмены изменений - u

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

Команды редактирования в режиме командной строки 8.2.3.1. Копирование и перемещение текста – : `▣,▣ d` — удалить строки с `▣` по `▣`; – : `▣,▣ m ▣` — переместить строки с `▣` по `▣`, начиная со строки `▣`; – : `▣,▣ t ▣` — копировать строки с `▣` по `▣` в строку `▣`; – : `▣,▣ w` имя-файла — записать строки с `▣` по `▣` в файл с именем имя-файла. 8.2.3.2. 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; – : `w q` — записать изменения в файл и выйти из `vi`; – : `q` — выйти из редактора `vi`; – : `q !` — выйти из редактора без записи;

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

- с помощью команды `$` — 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. Если вы хотите отказаться от использования опции, то в команде `set` перед именем опции надо поставить `no`.

12. Как определить режим работы редактора `vi`?

Если последняя строка начинается с “:”, то вы в режиме командной строки, если внизу строка отсутствует, то вы в режиме вставки, и иначе в режиме последней строки

13. Постройте граф взаимосвязи режимов работы редактора vi.

командный режим -> режим вставки -> режим последней строки