

Шаблон отчёта по лабораторной работе

Простейший вариант

Хань Цзянтао

Содержание

1	Цель работы.....	2
2	Задание.....	2
3	Теоретическое введение.....	2
4.	Выполнение лабораторной работы.....	3
4.2.1	Основные группы команд редактора.....	3
4.2.1.1.	Команды управления курсором Команды управления	3
-	3
4.2.1.3	Команды перемещения по файлу	3
4.2.1.4	Команды перемещения по файлу	3
4.2	Команды редактирования	3
4.2.2.1	Команды перемещения по файлу	3
4.2.2.2	Вставка строки	3
4.2.2.3	Удаление текста.....	3
4.2.2.4	Отмена и повтор произведённых изменений	4
4.2.2.5	Копирование текста в буфер	4
4.2.2.6	Вставка текста из буфера.....	4
4.2.2.7	Замена текста	4
4.2.2.8	Поиск текста.....	4
4.2.3	Команды редактирования в режиме командной строки	4
4.2.3.1	Копирование и перемещение текста.....	4
4.2.3.2	Запись в файл и выход из редактора.....	5

4.2.4	Опции	5
4.3.13	Задание 1. Создание нового файла с использованием vi	5
4.3.2	Задание 2. Редактирование существующего файла	7
4	Выводы.....	9
	Список литературы.....	9

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 наборе (восприятии) команд.

4.Выполнение лабораторной работы

4.2.1Основные группы команд редактора

4.2.1.1. Команды управления курсором Команды управления курсором приведены в табл Курсор влево Курсор вправо Курсор вверх Курсор вниз **4.2.1.2. Команды позиционирования – 0 (ноль) — переход в начало строки;**

- \$ — переход в конец строки;
- G — переход в конец файла;
- \square G — переход на строку с номером \square .

4.2.1.3Команды перемещения по файлу

- Ctrl-d — перейти на пол-экрана вперёд; – Ctrl-u — перейти на пол-экрана назад; – Ctrl-f — перейти на страницу вперёд; – Ctrl-b — перейти на страницу назад.

4.2.1.4Команды перемещения по файлу

- W или w — перейти на слово вперёд;
- \square W или \square w — перейти на \square слов вперёд;
- b или B — перейти на слово назад;
- \square b или \square B — перейти на \square слов назад.

4.2оманды редактирования

4.2.2.1Команды перемещения по файлу

- a — вставить текст после курсора;
- A — вставить текст в конец строки;
- i — вставить текст перед курсором;
- \square i — вставить текст \square раз;
- I — вставить текст в начало строки.

4.2.2.2Вставка строки

- o — вставить строку под курсором;
- O — вставить строку над курсором.

4.2.2.3Удаление 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.

4.2.2.4 Отмена и повтор произведённых изменений

- u — отменить последнее изменение;
- . — повторить последнее изменение.

4.2.2.5 Копирование текста в буфер

- Y — скопировать строку в буфер;
- [n] Y — скопировать [n] строк в буфер;
- y w — скопировать слово в буфер.

4.2.2.6 Вставка текста из буфера

- p — вставить текст из буфера после курсора;
- P — вставить текст из буфера перед курсором.

4.2.2.7 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.

4.2.2.8 Поиск текста

- / текст — произвести поиск вперёд по тексту указанной строки символов текст;
- ? текст — произвести поиск назад по тексту указанной строки символов текст.

4.2.3 Команды редактирования в режиме командной строки

4.2.3.1 Копирование и перемещение текста

- : [n],[n] d — удалить строки с [n] по [n];
- : [n],[n] m [n] — переместить строки с [n] по [n], начиная со строки [n];

- : \square, \square t \square — копировать строки с \square по \square в строку \square ;
- : \square, \square w имя-файла — записать строки с \square по \square в файл с именем имя-файла.

4.2.3.23 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;
- : e ! — вернуться в командный режим, отменив все изменения, произведённые со времени последней записи.

4.2.4 Опции

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. ## **4.3 Последовательность выполнения работы**

4.3.13 Задание 1. Создание нового файла с использованием vi

1. Создайте каталог с именем ~/work/os/lab06.

```
dskulyabov@dskulyabov:~$ cd work
dskulyabov@dskulyabov:~/work$ ls
lab8 study
dskulyabov@dskulyabov:~/work$ mkdir os
```

2. Перейдите во вновь созданный каталог.

```
dskulyabov@dskulyabov:~/work$ mkdir os
dskulyabov@dskulyabov:~/work$ cd os
dskulyabov@dskulyabov:~/work/os$ mkdir lab06
dskulyabov@dskulyabov:~/work/os$ cd lab06
```

3. Вызовите vi и создайте файл hello.sh

vi hello.sh

```
dskulyabov@dskulyabov:~/work/os/lab06$ vi hello.sh
[1]+  Остановлен    vi hello.sh
```

4. Нажмите клавишу i и вводите следующий текст. #!/bin/bash HELL=Hello
function hello { LOCAL HELLO=World echo \$HELLO } echo \$HELLO hello


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

8. Сделайте файл исполняемым `chmod +x hello.sh`

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

4.3.Задание 2. Редактирование существующего файла

1. Вызовите `vi` на редактирование файла `vi ~/work/os/lab06/hello.sh`

```
dskulyabov@dskulyabov:~/work/os/lab06$ vi ~/work/os/lab06/hello.sh
[4]~ Остановлен vi ~/work/os/lab06/hello.sh
```

2. Установите курсор в конец слова `HELL` второй строки.

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

3. Перейдите в режим вставки и замените на `HELLO`. Нажмите `Esc` для возврата в команд- ный режим.

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

4. Установите курсор на четвертую строку и сотрите слово LOCAL.

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

5. Перейдите в режим вставки и наберите следующий текст: local, нажмите Esc для возврата в командный режим.

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

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

```
Файл  Правка  Вид  Терминал  Вкладки  Справка
#!/bin/bash
HELLO=hello
function hello{
    local HELLO=World
    echo $HELLO
}
echo $HELLO
echo $HELLO
~
~
~
~
~
~
~
~
~
~
~
~
```

7. Нажмите Esc для перехода в командный режим.

8. Удалите последнюю строку.

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


4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM.— 2021.—URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix.— 2-е изд.—М. : МАКС Пресс, 2011.—URL: http://www.stolyarov.info/books/asm_unix.::