

# **Отчёта по лабораторной работе 10**

**Текстовой редактор vi**

Дельгадильо Валерия

# Содержание

Цель работы	5
Задание	6
Лабораторной работы	7
Ответы на контрольные вопросы	12
Выводы	14
Список литературы	15

## Список иллюстраций

1	Командой <code>vi hello.sh</code> откроем одноименный файл . . . . .	7
2	Введем в файл текст . . . . .	8
3	Выйдем из редактора с сохранением изменений и изменим права на исполнение . . . . .	9
4	Проверим содержание сохраненного файла . . . . .	9
5	Отредактируем файл и сохраним изменения . . . . .	10
6	Проверим содержание измененного файла . . . . .	11

## Список таблиц

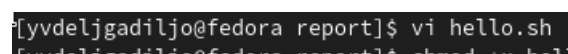
# Цель работы

Освоить интерфейс текстового редактора vi.

# Задание

- Выполнить задания по текстовому редактору vi
- Создать отчет и презентацию в md
- Загрузить скринкасты

# Лабораторной работы

A terminal window with a dark background. The prompt is '[yvdeljgadiljo@fedora report]\$'. The command 'vi hello.sh' has been entered and is highlighted in white text on a dark background.

```
[yvdeljgadiljo@fedora report]$ vi hello.sh
```

Рис. 1: Командой vi hello.sh откроем одноименный файл

[illegible]

Рис. 2: Введем в файл текст



```
[yvdeljgatiljo@fedora report]$ chmod +x hello.sh
```

Рис. 3: Выйдем из редактора с сохранением изменений и изменим права на исполнение

```
[yvdeljgatiljo@fedora work]$ cat hello.sh
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
```

Рис. 4: Проверим содержание сохраненного файла

[illegible]

Рис. 5: Отредактируем файл и сохраним изменения

```
[yvdeljgadiljo@fedora report]$ cat hello.sh
#!/bin/bash
HELLO=Hello
function hello {
    local HELLO=World
    echo $HELLO
}
echo $HELLO
hello
echo $HELLO
[yvdeljgadiljo@fedora report]$
```

Рис. 6: Проверим содержание измененного файла

## Ответы на контрольные вопросы

1. Дайте краткую характеристику режимам работы редактора vi.

Ответ: командный режим (command mode) - в этом режиме можно давать команды для редактирования файлов или перейти в другой режим, режим ввода (insert mode) - можно вводить текст в то место, куда указывает курсор, режим последней строки (last line mode) - это специальный режим, в котором редактору даются сложные команды.

2. Как выйти из редактора, не сохраняя произведённые изменения? Ответ: использовать в командном режиме команду :q!
3. Назовите и дайте краткую характеристику командам позиционирования. Ответ: есть несколько команд: 0 — переход в начало строки, \$ — переход в конец строки, G — переход в конец файла, kG — переход на строку с номером k
4. Что для редактора vi является словом? Ответ: символ, отделённый пробелом.
5. Каким образом из любого места редактируемого файла перейти в начало (конец) файла? Ответ: использовать в командном режиме команду G.
6. Назовите и дайте краткую характеристику основным группам команд редактирования. Ответ: Вставка текста, Вставка строки, Удаление текста, Отмена и повтор произведённых изменений, Копирование текста в буфер, Вставка текста из буфера, Замена текста, Поиск текста
7. Необходимо заполнить строку символами \$. Каковы ваши действия? Ответ: терпеливо заполняем

8. Как отменить некорректное действие, связанное с процессом редактирования?  
Ответ: клавиша u.
9. Назовите и дайте характеристику основным группам команд режима последней строки. Ответ: Копирование и перемещение текста, Запись в файл и выход из редактора.
10. Как определить, не перемещая курсора, позицию, в которой заканчивается строка? Ответ: перейти в конец строки.
11. Выполните анализ опций редактора vi (сколько их, как узнать их назначение и т.д.). Ответ: Команда vi имеет специальные опции. Они позволяют: восстановить потерянный файл, если работа vi была прервана; поместить в буфер редактирования несколько файлов и редактировать их по порядку; просмотр файла без риска случайного внесения изменений в него.
12. Как определить режим работы редактора vi? Ответ: режим работы отображается внизу экрана.

# Выводы

Я освоила интерфейс текстового редактора vi, работая с файлом и командной строкой.

# Список литературы

- GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
- GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
- Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
- NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
- 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>.
- Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
- The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
- Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
- Колдаев В. Д., Lupin С. А. Архитектура ЭВМ. — М. : Форум, 2018.
- Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
- Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
- Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.

- Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
- 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](http://www.stolyarov.info/books/asm_unix).
- Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
- Таненбаум Э., Бос Х. Современные операционн