Шаблон отчёта по лабораторной работе

Простейший вариант

Хань Цзянтао

Содержание

# 1 Цель работы

Познакомиться с операционной системой Linux. Получить практические навыки рабо- ты 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. # 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; – 𝑛 G — переход на строку с номером 𝑛. ####**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; – 𝑛 W или 𝑛 w — перейти на 𝑛 слов вперёд; – b или B — перейти на слово назад; – 𝑛 b или 𝑛 B — перейти на 𝑛 слов назад.

## 3.1 **4.2оманды редактирования**

####**4.2.2.1Команды перемещения по файлу** – а — вставить текст после курсора; – А — вставить текст в конец строки; – i — вставить текст перед курсором; – 𝑛 i — вставить текст 𝑛 раз; – I — вставить текст в начало строки. ####**4.2.2.2Вставка строки** – о — вставить строку под курсором; – О — 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. ####**4.2.2.4Отмена и повтор произведённых изменений** – u — отменить последнее изменение; – . — повторить последнее изменение. ####**4.2.2.5Копирование текста в буфер** – Y — скопировать строку в буфер; – 𝑛 Y — скопировать 𝑛 строк в буфер; – y w — скопировать слово в буфер. ####**4.2.2.6Вставка текста из буфера** – p — вставить текст из буфера после курсора; – P — вставить текст из буфера перед курсором. ####**4.2.2.7Замена 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. ####**4.2.2.8Поиск текста** – / текст — произвести поиск вперёд по тексту указанной строки символов текст; – ? текст — произвести поиск назад по тексту указанной строки символов текст. ## **4.2.3Команды редактирования в режиме командной строки**

####**4.2.3.1Копирование и перемещение текста**

– : 𝑛,𝑚 d — удалить строки с 𝑛 по 𝑚; – : 𝑖,𝑗 m 𝑘 — переместить строки с 𝑖 по 𝑗, начиная со строки 𝑘; – : 𝑖,𝑗 t 𝑘 — копировать строки с 𝑖 по 𝑗 в строку 𝑘; – : 𝑖,𝑗 w имя-файла — записать строки с 𝑖 по 𝑗 в файл с именем имя-файла.

####**4.2.3.2Запись 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.

## 3.2 **4.2.4Опции**

Опции редактора vi позволяют настроить рабочую среду. Для задания опций использу- ется 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. ## **4.3Последовательность выполнения работы** ####**4.3.1ЗЗадание 1. Создание нового файла с использованием vi** 1. Создайте каталог с именем ~/work/os/lab06. 2. Перейдите во вновь созданный каталог. 3. Вызовите vi и создайте файл hello.sh

vi hello.sh

1. Нажмите клавишу i и вводите следующий текст. #!/bin/bash HELL=Hello function hello { LOCAL HELLO=World echo $HELLO } echo $HELLO hello
2. Нажмите клавишу Esc для перехода в командный режим после завершения ввода текста.
3. Нажмите : для перехода в режим последней строки и внизу вашего экрана появится приглашение в виде двоеточия.
4. Нажмите w (записать) и q (выйти), а затем нажмите клавишу Enter для сохранения вашего текста и завершения работы.
5. Сделайте файл исполняемым chmod +x hello.sh ####**4.3.2Задание 2. Редактирование существующего файла**
6. Вызовите vi на редактирование файла vi ~/work/os/lab06/hello.sh
7. Установите курсор в конец слова HELL второй строки.
8. Перейдите в режим вставки и замените на HELLO. Нажмите Esc для возврата в команд- ный режим.
9. Установите курсор на четвертую строку и сотрите слово LOCAL.
10. Перейдите в режим вставки и наберите следующий текст: local, нажмите Esc для возврата в командный режим.
11. Установите курсор на последней строке файла. Вставьте после неё строку, содержащую следующий текст: echo $HELLO.
12. Нажмите Esc для перехода в командный режим.
13. Удалите последнюю строку.
14. Введите команду отмены изменений u для отмены последней команды.
15. Введите символ : для перехода в режим последней строки. Запишите произведённые изменения и выйдите из vi.

# 4 Выводы

# Список литературы

::: {#refs}

1. GDB: The GNU Project Debugger. — URL: https://www.gnu.org/software/gdb/.
2. GNU Bash Manual. — 2016. — URL: https://www.gnu.org/software/bash/manual/.
3. Midnight Commander Development Center. — 2021. — URL: https://midnight

commander. org/.

1. NASM Assembly Language Tutorials. — 2021. — URL: https://asmtutor.com/.
2. 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.

1. Robbins A. Bash Pocket Reference. — O’Reilly Media, 2016. — 156 с. — ISBN

978-1491941591.

1. The NASM documentation. — 2021. — URL: https://www.nasm.us/docs.php.
2. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN
3. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
4. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. :

Солон-Пресс, 2017.

1. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
2. Расширенный ассемблер: NASM.— 2021.—URL: https://www.opennet.ru/docs/RUS/nasm/.
3. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е

изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.

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.:::