

Федеральное государственное автономное образовательное учреждение высшего  
образования  
«Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики»  
Кафедра Вычислительной Техники

**Системное программное обеспечение**  
**Дополнительное задание №1**  
*vim*

Выполнила: Калугина Марина  
Группа: Р3202

г. Санкт-Петербург  
2019 г.

## Vi vs Vim

Vi - многорежимный редактор, vim - расширенная версия редактора vi, поэтому всё, что есть в vi доступно и в vim.

Некоторые улучшения vi, доступные в vim:

1. Работа с многими файлами. Разбить окно для работы можно многократно и как по горизонтали, так и по вертикали.
  2. Поддержка Unicode.
  3. Режим сравнения двух файлов.
  4. Подсветка синтаксиса. Автоматическое определение длины отступа.
  5. Возможность писать расширения.
  6. Автоматический вызов внешних команд (например, распаковка файла).
  7. Поддержка языков с письмом справа налево.
  8. Поддержка визуального режима для операций над блоками текста.
  9. Хорошо конфигурируется и настраивается под нужды пользователя.
- и т.п.

## Набор текста в интерактивном режиме.

Режим ввода текста в vim - не основной. Основным режимом в vim`е считается режим, в котором клавиатура используется для управления редактором.

Переход из обычного режима в <Insert>:

- i - перейти в режим ввода с текущей позиции
- a - перейти в режим ввода после курсора
- I - переместиться в начало строки и перейти в режим ввода
- A - переместиться в конец строки и перейти в режим ввода
- o - перейти в режим ввода с новой строки под курсором
- O - перейти в режим ввода с новой строки над курсором
- s - заменяет символ и вводит.
- S - удаляет всю текущую строку и переходит в режим ввода.
- R — перейти в режим ввода с заменой текста (аналог insert).
- г — заменить символ. При этом не происходит выхода из командного режима.

В командах могут применяться числовые аргументы. Например, команда 20iПривет<esc> вставляет 20 слов "Привет".

## Вставка текста из файла

- Команда :r file вставит текст из файла <file> на место курсора. :23r file вставит текст начиная с 23 строки.
- Открыть другой файл, скопировать строку(или несколько) в буфер "qY, вернуться в исходный файл, вставить скопированную строку "qP

- Для записи в другой файл из текущего используется команда `:1,10 w outfile` - запишет первые 10 строчек в *outfile*. `:1,10 w >> outfile` - добавит в *outfile* эти строчки.

## Открытие файла из редактора

`:e hello.txt` — открывает файл для редактирования

`CTRL-w n` - создает новое окно

`:new file` - открыть *file* в новом окне

## Открытие файла при запуске редактора

В самом простом случае для открытия редактора нам понадобится команда `vim file`, которая открывает файл *file*. При добавлении `+n` файл откроется на строке под номером *n*.

`+{command} -c {command}` после открытия файла сразу выполнит внутреннюю команду *command*.

`-d` - откроет vimdiff

`-r` - ключ для read-only

`vim file1 file2` откроет *file1*, а после выхода из него откроет *file2*. С ключем `-o/-O`, откроет файлы одновременно в разных окнах.

## Навигация по тексту, абзацам, строкам, словам, символам

`Ctrl-f` — на страницу (экран) вниз

`Ctrl-b` — на страницу (экран) вверх

`Ctrl-y` — на строку вниз, без изменения положения курсора

`Ctrl-e` — на строку вверх, без изменения положения курсора

`0` — в начало текущей строки;

`$` — в конец текущей строки

`w` — на слово вправо

`b` — на слово влево

`W` — до пробела вправо

`B` — до пробела влево

`}` — абзац вниз

`{` — абзац вверх

`gg` — перейти в начало файла

`G` — перейти в конец файла

`<number>G` — перейти на конкретную строку `<number>`

`/<text>` — перейти к `<text>`

`?<text>` — то же самое, но искать назад

`n` — повторить поиск

`N` — повторить поиск назад

## Копирование, перемещение и удаление блоков текста

*x* - удалить символ под курсором

*X* - удалить символ перед курсором

*d* - удалить. Используется совместно с командами перемещения или удаляет выделенный участок (автоматически помещает удаленный текст в свой буфер, так что после можно вставить текст при помощи команды *p* или *P*)

*dd* - удалить строку

*dw* - удалить слово после курсора

*db* - удалить слово перед курсором

*d0* - удаление символов с начала строки

*d\$* - удаление символов до конца строки

*v* - режим выделения

*V* - режим выделения для строк

*y* - копировать выделенный фрагмент в буфер

*p* - вставить из буфера после курсора

*P* - вставить из буфера перед курсором

## Поиск и замена блоков текста

*/<text>* — перейти к *<text>*

*?<text>* — то же самое, но искать назад

*n* — повторить поиск

*N* — повторить поиск назад

*:%s/<str1>/<str2>/g* Заменить все вхождения *<str1>*, на *<str2>*

## Изменение параметров редактора: вывод нумерации строк, вывод невидимых метасимволов

Изменить параметры редактора можно двумя способами. Установить параметр для текущей сессии или изменить в конфигурационном файле *.vimrc*

*:set number* - нумерации строк

*:set list* - отображение непечатных символов

*:set nolist* - отключение отображения непечатных символов

*:set encoding* - задать кодировку файла

*:set wrap* - переносить длинные строки

*:set ignorecase* - установить регистронезависимый поиск

*:set shiftwidth=n* - установить размер отступа *n*

*:set syntax=perl* - установить режим подсветки

## Выход из редактора различными способами

`:q` - выход из текущего окна  
`:qa` - выход из всех окон  
`:wq` - записать изменения и выйти  
`:wqa` - записать изменения и выйти для всех окон  
`:x` - сохранить если есть изменения и выйти  
`:q!` - выйти без сохранения изменений

## Работа с несколькими файлами

В vim`е можно открыть несколько файлов. Vim поддерживает многооконную работу. Разбить окно можно как по вертикали, так и по горизонтали. Для горизонтальной разбивкой окон на экране: `vim -o file1 file2`, для вертикальной нужно пользоваться ключем `-O`.

Команды переключения между окнами:

`CTRL-w w` - быстрое переключение на следующее окно  
`CTRL-w влево` - активация левого окна  
`CTRL-w вправо` - активация правого окна  
`CTRL-w вверх` - активация верхнего окна  
`CTRL-w вниз` - активация нижнего окна  
`CTRL-w n` - создать новое окно ("`:new` файл" - открыть заданный файл)  
`CTRL-w q` - закрыть текущее окно  
`CTRL-w s` - разбить текущее окно на 2 части для синхронного редактирования разных частей  
`CTRL-w +`, `CTRL-w --` - увеличение или уменьшение текущего размера окна  
`ESC 5 CTRL-w -`, `ESC 5 CTRL-w +` - изменение размера сразу на 5 строк/столбцов  
`CTRL-w h` - переместить окно влево  
`CTRL-w j` - переместить окно ниже  
`CTRL-w k` - переместить окно выше  
`CTRL-w l` - переместить окно вправо  
`CTRL-w t` - переместить окно на самый верх  
`CTRL-w b` - переместить окно в самый низ  
`:only` - Закрыть все окна, кроме текущего

Все вышеперечисленное касалось работы в разных окнах. Кроме того, можно просто работать с несколькими файлами и при необходимости переключаться между ними используя `:e file`

## Работа с командным интерпретатором

`:sh` - открыть shell

*exit* - вернуться в редактор из shell

*!command* - выполнить *command* и вернуться

*!!command* - выполнить *command* и вставить результат в файл