

# Занятие 10.

**Тема:** Текстовые редакторы. Редактор vi.

**Вид занятия:** лекция, практическое занятие.

**Учебные вопросы:**

1. Редактор vi.

**Время:** 90 минут

**Литература:**

1. Системная справочная служба Linux Man
2. Команда vintutor.ru.
3. Шевель А. “Linux. Обработка текстов. Специальный справочник.” - Спб.: Питер, 2001. - 384 с.:ил.

## Ход занятия.

В Linux широко используются около десятка текстовых редакторов. Каждый из них хорош для своих целей. Но наиболее часто используемыми все же являются редакторы vi и emacs.

1. Редактор vi – один из старейших и мощнейших редакторов в Linux. Он работает даже на таких аппаратных терминалах как VT100 и XTERM. Современный вариант этого редактора называется vim (Vi IMproved – улучшенный vi). vim эмулирует все команды vi, и в то же время добавляет много дополнительных возможностей, таких как:

- многоуровневый процесс отмены ранее выполненных действий (undo);
- использование нескольких окон редактирования;
- редактирование командной строки;
- встроенная справочная система (команда **:help**) и многое другое.

Мы с вами будем обсуждать именно vim – расширенный вариант vi.

Запуск редактора производится командой vim:

```
vim <имя файла>
```

Если вы запустили vim без указания имени файла, то Вы можете или начать редактирование, или открыть файл командой **:edit полное\_имя\_файла<Enter>**.

Сохранить текущий файл можно с помощью команды **:w <имя\_файла><Enter>**. Параметр имя\_файла в команде сохранения необязателен. Если он не указан, что файл будет сохранен с тем же именем, с которым и был открыт. Если же вы редактируете новый файл, то имя указать обязательно.

Выйти из vim можно по команде **:q<Enter>**. Для того, чтобы выйти из редактора, сохранив все сделанные изменения, используйте комбинацию команд **:qw <Enter>**, а для отмены сохранения всех сделанных изменений - **:q!<Enter>**.

vim может работать в нескольких режимах: просмотра и ввода команд(по умолчанию), редактирования и выделения.

Из-за того, что редактор обеспечивает совместимость с разными типами терминалов, кроме стандартных клавиш управления, он поддерживает клавиши буквенной клавиатуры для управления курсором:

```
      ^  
      k  
  < h   l >  
      j  
      v
```

Такой режим перемещения работает только в режиме просмотра и режиме выделения.

Для того, чтобы узнать, в каком месте файла Вы сейчас находитесь, нажмите **<Ctrl+g>**. Внизу экрана появится строка состояния, в которой будет указано имя редактируемого файла и текущая позиция в нем.

Для перехода к концу файла, Вам необходимо нажать комбинацию **<Shift+G>**. Если же

нужно перейти к какой-то определенной строке в файле, нажмите **<N\_строку Shift+G>**, например для перехода к 35 строке: **<3><5><Shift+G>**

Поиск по файлу производится с помощью команды **/** (аналогично поиску в less). По умолчанию поиск регистрозависимый (то есть маленькие и большие буквы считаются разными). Изменить это можно с помощью команды **:set ic<Enter>**. Подсветить цветом все искомые совпадения можно командой **:set hls<Enter>**.

Продолжить поиск до следующего вхождения вниз по тексту можно командой **<n>**, а вверх по тексту - **<Shift+N>**.

Для удаления символов в режиме ввода команд, Вам необходимо нажать **<Del>** или **<x>**, а для удаления сразу целой строки – **<dd>**. Если понадобилось удалить сразу несколько строк, то перед вводом команды **<dd>**, необходимо ввести число удаляемых строк (например: **<5dd>** удалит 5 строк, включая текущую вниз по тексту). Для удаления всех символов до конца слова необходимо набрать команду **<dw>**, а до конца строки – **<d\$>**. Если же необходимо удалить строку до ее начала – используйте комбинацию **<d^>** (Если в начале строки есть пустые символы, то удалить до начала строки вместе с пустыми символами поможет команда **<d0>**).

Для начала редактирования (вставки текста) необходимо нажать **<Insert>** или **<i>**. Перевести vim в режим замены символов в режиме редактирования также может повторно нажатая клавиша **<Insert>**, а в обычном режиме - **<Shift+R>**. Выйти из режима редактирования можно с помощью нажатия **<Esc>**

Если в процессе редактирования Вы сделали ошибку, и Вам необходимо выполнить откат выполненных действий, то:

- перейти в режим ввода команд;
- нажать **<u>**.

Откат выполненной операции также действует на операции удаления символов, строк, слов и частей строк. Для отката отката (возврата выполненных команд) используется комбинация **<Ctrl+r>**.

При выполнении операций удаления, удаленный текст помещается в буфер. Этот буфер можно вставить потом в любое место в редактируемом тексте. Делается это нажатием клавиши **<p>**. Вставляемый текст будет помещен после текущей строки.

Для копирования текста в буфер используются команды **<y>**(1 символ), **<y\$>**(до конца строки), **<y^>**(до начала строки), **<yw>**(слово), **<yy>**(вся строка).

Для изменения части слова используется команда **<cw>**. Встав над словом, которое необходимо исправить, нажмите **<cw>**. После этого все знаки слова, начиная с позиции курсора будут удалены и vim перейдет в режим редактирования.

Команда `<c>` используется с теми же параметрами, что и команда удаления, за исключением начала строки:

`<cw>` – редактировать до конца слова

`<c$>` - редактировать до конца строки

Можно применять цифры, например:

`<2cw>` - редактировать до конца второго слова, включая текущее

`<5c$>` - редактировать до конца пятой строки, включая текущую.

Vim позволяет производить замену:

`:s/было/стало/g` – сделать замену в текущей строке

`:1,20s/было/стало/g` – сделать замену в строках с 1 по 20-ю

`:%s/было/стало/g` – сделать замену во всем файле

`:30,100s/было/стало/gc` – сделать замену в строках с 30 по 100 и спрашивать при каждой замене разрешения сделать ее.

`:/слово/s/было/стало/g` – сделать замену в строках содержащих "слово"

Для вставки файла (слияния) используется команда `:r имя_файла <Enter>`. В результате выполнения этой команды файл `имя_файла` будет прочитан и вставлен после текущей строки.

Во время работы с редактором часто возникает необходимость выполнить команду оболочки без выхода из редактора. Vim предоставляет такую возможность:

`:!команда` – выполнить команду “команда”

`:%!команда` – пропустить редактируемый текст через внешнюю команду “команда”

Во время выполнения команд нужно учитывать, что все команды действуют в пределах вновь вызванного командного интерпретатора, и потому не меняют системного окружения текущего, т.е. в результате выполнения команд не изменится рабочий каталог редактора или какие-либо переменные окружения.

Помните, что все команды, начинающиеся с двоеточия, необходимо заканчивать нажатием `<Enter>`.