

Занятие 2. Пользовательские средства ОС Linux. Часть первая.

2.1. Работа с файлами

2.1.1. Vim

Vim – режимный текстовый редактор, созданный на основе более старого vi.

Режимы работы:

- **нормальный/обычный/основной** – режим перемещения по файлу, стирание текста и другое редактирование

переход в него из любого другого режима можно осуществить нажатием клавиши <ESC> (в некоторых случаях дважды) или <Ctrl-]>

среди команд нормального режима есть односимвольные команды:

основные команды перемещения:

h – перейти на символ влево

j – спуститься на строчку вниз

k – подняться на строчку вверх

l – перейти на символ вправо

Удалить символ позволяет символ x – непосредственно выделенный символ, X – до него.

Символ г – позволяет заменить текущий символ на следующий введённый, R – включает режим “забоя” (аналогично нажатию Insert).

Символ р – производит вставку из буфера после выделенного, Р – до.

Так, чтобы поменять местами 2 символа (часто встречающаяся опечатка), достаточно набрать хр.

Символ u – отменяет последнее действие. Возвращает отменённое действие - <Ctrl-r>.

Символ . (точка) повторяет последнее совершённое действие.

0 – переходит на первый символ строки, \$ - на последний символ в строке, ^ - первый непробельный символ.

Символ w – переходит вперёд на слово, останавливаясь на первом символе слова, e – на последнем; b – назад.

Последовательность gg – переводит на начало файла, G – в конец файла.

Символ d – производит удаление (в помещением в буфер), согласно следующему символу за ним. Так d0 –

удаляет все символы до начала строки, d\$ - до конца строки, dG – до конца документа, dw – удалит до конца слова.

Последовательность dd – позволяет удалить всю текущую строку.

Символ у – производит копирование в буфер. Аналогично команде d:

y0 – копирует все символы до начала строки, yw – копирует слово и т.д.

Последовательность yy – копирует всю текущую строку полностью.

- **Режим ввода текста**

перейти в него из нормального режима можно разными способами в зависимости от того, где удобнее “встать” в данном случае:

“a” – после текущего символа

“i” - до текущего символа

“A” - после последнего символа в строке

“I” - до первого символа в строке

“o” - добавить строку снизу от текущей и перейти в режим ввода текста в её начале

“O” - добавить строку над текущей и перейти на её начало

“s” - стирает символ и переходит в режим ввода текста

“c” - удаление (работает аналогично d) и переходит в режим ввода текста

- **Командный** – режим операций с файлом, поиска и замены, настройки редактора

перейти из обычного режима в него - “:”.

Примеры команд

:w – сохранить текущий файл

:wa – сохранить все открытые файлы

:q – закрыть файл

:wq – закрыть файл с сохранением

:q! - закрыть файл без сохранения изменений

:e filename – открыть файл с именем filename (может быть путь до файла – как относительный, так и

абсолютный)

- **Режим поиска**

переход из обычного:

/ - активирует поиск вперед

? - активирует поиск назад

после одного из этих символов набираем строчку, которую ищем, и нажимаем <Enter>

Чтобы перейти на следующее вхождение – n, следующее в обратном направлении от текущего – N.

Примеры:

/^Warning: - ищет строки, которые начинаются с “Warning:”

- **Визуальный режим**

перейти в него из обычного:

v – позволяет выделить текст визуально, а затем можно его удалить(нажатием символа d) или копировать(y)

V – выделяет целыми строками

Vim имеет достаточно удобную и подробную справочную систему, и систему навигации и перехода между её разделами. Чтобы открыть любой её раздел достаточно воспользоваться командой :help в обычном режиме.

2.1.2. Вывод текстовой информации

Информация внутри регулярных файлов в Линукс может содержаться в двух видах: текстовом (ASCII) или бинарном.

Для каждого из них существуют программы, позволяющие просмотреть содержимое файла.

Для вывода текстовой информации наиболее часто используются команды cat, tac, more, less, tail и head.

Команда cat выводит на экран содержимое текстового файла. Наиболее часто используются следующие ключи этой команды:

Ключ	Значение
-b	пронумеровать все непустые строки
-n	пронумеровать все строки
-s	отобразить несколько подряд идущих пустых строк как одну пустую строку
-T	показать символы табуляции как последовательность " ^ "
-E	показать символы конца строки как \$

Пример использования команды:

```
[user@localhost ~]$ cat -E /selinux/enforce
0[user@localhost ~]$
```

Команда tac используется для отображения текста на экране в обратном порядке. Ключ -s (--separator) позволяет указать символ, которым разделяются записи. По умолчанию используется символ конца строки (\n).

В случае, когда длина текстового файла такова, что целиком он не умещается на экране, можно использовать команду more. Эта команда позволяет просматривать файл по частям, перемещаясь к следующей строке по нажатию клавиши Enter и к следующему экрану по Space. Команда more не позволяет вернуться к ранее просмотренному экрану. Более гибкую навигацию по текстовому файлу предоставляет команда less. Эта команда позволяет прокручивать текст в любом направлении, а также искать вхождения групп символов. Переход в режим поиска осуществляется клавишей '/. Команды head и tail позволяют просмотреть часть файла, не открывая его полностью. По умолчанию они выводят по 10 строк, соответственно, первых и последних в файле. Флаг -n N позволяет указать другое число выводимых строк (N). Флаг -с N заставляет команду вывести N байт.

Флаг -f команды tail вводит ее в режим непрерывного считывания конца файла. При дописывании в файл новой информации она автоматически отображается на экране в реальном времени. Эта функция часто используется для просмотра файлов журналов.

Для вывода на экран данных из бинарных файлов в читаемом формате можно использовать команду od. Эта команда выводит на экран так называемый дамп бинарных данных в формате десятичных, восьмиричных или шестнадцатиричных чисел. Ключ -A указывает систему счисления для отображения адресов в файле. Ключ -d выводит дамп данных в десятичной системе, -o - в восьмиричной (по умолчанию), -x - в шестнадцатиричной системе.

2.1.3. Перенаправление ввода/вывода

Большинство программ Линукс оперируют тремя потоками данных: стандартным вводом, выводом и потоком ошибок.

Все эти потоки автоматически открываются всеми процессами. По умолчанию для получения данных используется

стандартный ввод, для вывода - стандартный вывод и поток ошибок.

stdin - стандартное устройство ввода. Номер файлового указателя для этого потока 0.

stdout - стандартное устройство вывода. Номер указателя 1.

{stderr - стандартный поток ошибок. Номер указателя 2.

Эти потоки можно перенаправлять друг в друга и в файлы. Для перенаправления используются специальные операторы:

> - перенаправляет стандартный поток в файл или другой поток. При этом если файл не существует, он создается, если существует - перезаписывается.

>> - перенаправляет стандартный поток в файл. При этом если файл не существует, он создается, если существует - данные добавляются в конец файла.

< - перенаправляет содержимое файла в стандартный ввод программы.

>& - перенаправляет стандартный вывод в поток ошибок или наоборот, например: 2>&1.

Также возможно перенаправление стандартного вывода программы на стандартный ввод другой программы. Для этого используются каналы (pipes). Канал - это односторонний программный интерфейс, позволяющий процессами обмениваться данными. Для управления каналом используется оператор |. Пример использования канала для передачи информации от одной команды к другой:

```
[user@localhost ~]$ cat /var/log/messages | tail -n 3
```

```
Dec 15 13:26:25 localhost pseudo: [ID 129642 kern.info] pseudo-device: llc10
```

```
Dec 15 13:26:25 localhost genunix: [ID 936769 kern.info] llc10 is /pseudo/llc1@0
```

```
Dec 15 13:26:46 localhost zfs: [ID 327478 kern.notice] NOTICE: The property is  
nms:dedup-dirty
```

2.1.4. Фильтрация строковых данных, регулярные выражения.

Для того, чтобы найти в текстовых файлах вхождения известных подстрок или данные в известном формате, можно использовать фильтры и регулярные выражения.

Основным средством для фильтрации файлов в Линукс является программа grep. В качестве аргументов для этой команды должны быть указаны регулярное выражение и имя файла или каталога: grep regexp filename.

Регулярными выражениями называются средства указания шаблона для поиска его в тексте. Это может быть просто слово или его часть, а может быть намного более сложная структура, позволяющая выбирать данные различного формата в рамках одного выражения. Наиболее часто используются следующие специальные символы:

^ – начало строки;

\$ – конец строки;

[] – любой символ из заключенных в скобки. Поддерживает диапазоны, например, [1-6] - любая цифра от 1 до 6, и т.п.;

[^] – любой символ, кроме указанных в скобках после ^;

\} – превращает управляющий символ в обычный, например, {\\$ означает символ '\$', а не конец строки;

. – любой символ;

* – предыдущий шаблон встречается 0 или более раз в тексте.

Если аргумент filename не указан, команда grep осуществляет поиск в тексте, поступающем на стандартный вход. Это можно использовать для поиска в выводах других команд, перенаправляя их вывод на вход grep через канал (pipe).

Пример использования команды grep:

```
[user@localhost ~]$ cat /var/log/messages | grep -i "err"
```

```
[user@localhost ~]$ grep "[Ee]rr" file.log
```

Поиск по регулярным выражениям также поддерживается большинством текстовых редакторов в системе Линукс.