

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра сетей и устройств телекоммуникаций

***КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ
В СЕТЯХ ТЕЛЕКОММУНИКАЦИЙ***

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к лабораторно-практическим занятиям
для студентов специальности 45 01 03 «Сети телекоммуникаций»
дневной, вечерней и заочной форм обучения

В 3-х частях

Часть 3

Операционная система DRS-Unix.
Работа с редактором VI и другими файлами ОС

Минск 2003

УДК 004 + 621.391.(075.8)
ББК 32.973 + 32.811 я 7
К 63

С о с т а в и т е л и:
В.А. Мельников, Я.В. Рощупкин

Компьютерные технологии в сетях телекоммуникаций: Метод.
К 63 указания к лабораторно-практическим занятиям для студ. спец. 45 01 03
«Сети телекоммуникаций» дневной, вечерней и заочной форм обучения:
В 3 ч. Ч. 3. Операционная система DRS-Unix. Работа с редактором VI и
другими файлами ОС / Сост. В.А. Мельников, Я.В. Рощупкин. – Мн.:
БГУИР, 2003. – 20 с.

Методические указания включают основные теоретические положения, методику выполнения работ и подготовки к защите лабораторно-практических работ.

УДК 004+621.391.(075.8)
ББК 32.973+32.811 я 7

Ч. 1. Операционная система DRS-Unix. Основные особенности работы, работа с файлами, регистрация в системе. Ч. 2. Операционная система DRS-Unix. Основные службы. Изданы в БГУИР в 2003 г.

© Мельников В.А., Рощупкин Я.В.,
составление, 2003
© БГУИР, 2003

ЛАБОРАТОРНАЯ РАБОТА № 4 «Текстовый редактор vi»

Редактор **vi** является мощным средством для создания и редактирования файлов. Он предназначен для работы на видеотерминале и использует окна, посредством которых можно просматривать текст файла. Несколько простых команд позволяют сделать серьезные изменения в тексте. Редактор **vi** имеет более 100 команд. Здесь будут описаны основные команды, которые позволят использовать **vi** просто и эффективно.

Редактор **vi** имеет два рабочих режима: режим ввода текста и командный режим.

1. Запуск редактора и выход из него

Для запуска редактора достаточно ввести в командной строке **vi** и имя файла, затем нажать <Enter>:

```
$ vi filename
```

Когда пользователь введет команду **vi** с именем файла, **vi** очистит экран и отобразит окно, в котором можно вводить и редактировать текст.

Пять основных команд могут быть использованы для завершения работы с редактором **vi**. Команды, которым предшествует «:», являются командами построчного редактора:

- :wq<ENTER>** – записать содержимое буфера **vi** в файл, который в данный момент редактируется, и завершить работу с **vi**;
- ZZ** – записать содержимое буфера, если он изменялся с момента последней записи;
- :qw filename<ENTER>** – записать временный буфер в новый файл filename и завершить работу с редактором **vi**;
- :q!<ENTER>** – завершить работу с редактором **vi** без записи содержимого буфера в файл; не учитываются все изменения, проведенные в буфере;
- :q<ENTER>** – завершить работу с редактором **vi** без записи содержимого буфера в файл. Эта команда работает в том случае, если изменений произведено не было. В противном случае **vi** выдаст предупреждающее сообщение о том, что необходимо сохранить буфер или воспользоваться для завершения работы с **vi** командой **:q!<ENTER>**.

Пример:

```
<a>
```

```
This is a text file.<ENTER>
I am adding text to<ENTER>
a temporary buffer and<ENTER>
now it is perfect.<ENTER>
I want to write this file,<ENTER>
and return to the shell.<ESC><ZZ>
```

"stuff" [New file] 6 lines, 135 characters

\$

Если необходимо дать файлу другое имя, например записать в новый файл junk, следует ввести команду:

```
:w junk<ENTER>
```

Затем завершить работу с **vi**:

```
:q<ENTER>
```

Если файл с таким именем уже существует, то программа выдаст предупреждающее сообщение. Например, если попытаться записать в файл johnson, система выдаст сообщение:

```
"johnson" File exists - use "w!johnson" to over-
write
```

Если необходимо заменить содержимое существующего файла на содержимое буфера, то следует воспользоваться командой **:w!**:

```
:w! johnson<ENTER>
```

2. Перемещение курсора по экрану

В этом подразделе приводятся несколько команд, которые позволяют быстро перемещать курсор по экрану. Эти команды выполняются в командном режиме.

2.1. Позиционирование курсора в окне

Редактор **vi** предоставляет 3 команды, которые позволяют позиционировать курсор в окне. Эти команды должны быть введены на верхнем регистре:

H – переместить курсор к первой строке экрана;

M – переместить курсор к средней строке экрана;

L – переместить курсор к последней строке экрана.

Ниже приведены команды перемещения курсора для позиционирования его на символе, строке, предложении, параграфе или позиционирования на экране.

Позиционирование на символе

Существуют команды, которые проводят поиск по шаблону: **f**, **F**, **t**, **T**, **:** (двоеточие) и **,** (запятая). После **f**, **F**, **t** и **T** необходимо указать символ:

fx – осуществляет поиск указанного символа **x** справа от текущей позиции курсора;

Fx – осуществляет поиск указанного символа **x** слева от текущей позиции курсора;

tx – осуществляет перемещение курсора вправо на символ до указанного символа **x**;

Tx – осуществляет перемещение курсора влево на символ до указанного символа **x**;

; – повторяет предыдущий поиск символа. Команда **;** запоминает символ и ищет следующее появление символа в текущей строке;

, – повторяет предыдущий поиск символа, но в обратном направлении. Команда **,** запоминает символ и ищет следующее появление символа в текущей строке.

Команда перемещения – (знак минус)

Команда – (минус) перемещает курсор на одну строку вверх, позиционируя его на первом ненулевом символе. Чтобы переместить курсор более чем на один символ, укажите перед знаком минус количество строк. Например, чтобы переместить курсор вверх на 13 строк, введите:

13-

Если эти строки находятся выше отраженного экрана, то окно переместится так, чтобы открыть их. Этот способ помогает быстро перемещаться по файлу. Если выше окажется меньше 13-ти строк, то звуковой сигнал подскажет, что сделана ошибка и курсор останется в своем прежнем положении.

Команда перемещения + (знак плюс)

Команда + (плюс) или **<ENTER>** перемещает курсор вниз на одну строку и позиционирует на первый непустой символ. Перед командой + можно указать количество строк, на которое необходимо переместить курсор. Например, чтобы переместить курсор на 9 строк вниз, следует ввести:

9+

Если некоторые строки расположены ниже текущего экрана, то окно переместится вниз, чтобы открыть их.

Позиционирование на слове

В редакторе **vi** предполагается, что слово – это строка символов, которая может включать в себя буквы, цифры и символы подчеркивания. В редакторе **vi** существует 6 команд позиционирования: **w**, **b**, **e**, **W**, **B** и **E**. Команды **w**, **b**, **e** рассматривают любой символ, отличный от буквы, цифры или подчеркивания, как разделитель, означающий начало или конец слова. Пунктуация до или после пустого пространства является разделителем слов. Символ начала или конца строки также является разделителем.

Команды **W**, **B**, **E** рассматривают пунктуацию как часть слова. Слова разделяются пустым пространством, которое состоит из символов пробел, табуляция и новая строка.

Функции команд позиционирования:

- w** – перемещает курсор к началу следующего слова;
- nw** – перемещает курсор к началу n-го слова; конец строки не останавливает движение курсора. Он продолжает подсчет слов с начала следующей строки;
- W** – игнорирует все знаки пунктуации и перемещает курсор на следующее после пробела слово;
- e** – перемещает курсор на последний символ в следующем слове;
- E** – перемещает курсор на последний символ в слове. Игнорирует все символы пунктуации, за исключением пробела. Слова разделяются пробелами;
- b** – перемещает курсор к первому символу предыдущего слова;
- nb** – перемещает курсор к первому символу n-го предыдущего слова. Команда **b** не останавливается в начале строки и продолжает движение к началу строки;
- B** – используется подобно команде **b**, за исключением того, что слова разделяются только символами пробел и новая строка. Она рассматривает все другие символы пунктуации как буквы.

Позиционирование на строке

Приведенные ниже команды позволяют:

- \$** – переместить курсор на последний символ в строке;
- 0(ноль)** – переместить курсор на первый символ в строке;
- ^** – переместить курсор на первый ненулевой символ в строке.

2.2. Позиционирование курсора на предложении

Редактор **vi** также распознает предложения. В редакторе **vi** предложение заканчивается знаками «!», «.» или «?»). Если эти разделители появляются в се-

редине строки, то за ними должны стоять два пробела, чтобы **vi** мог их распознать.

Можно перемещать курсор от предложения к предложению в файле с помощью команд (– открывающая скобка и) – закрывающая скобка:

- (– перемещает курсор в начало текущего предложения;
- n**(– перемещает курсор в начало n-го предложения выше текущего предложения;
-) – перемещает курсор в начало следующего предложения;
- n**) – перемещает курсор в начало n-го предложения, следующего за текущим предложением.

2.3. Позиционирование курсора на параграфе

Параграфы распознаются редактором **vi**, если они начинаются после пустой строки. Если необходимо перемещать курсор по параграфам, то при создании текста следует в конце каждого параграфа вставить пустую строку. Варианты команд позиционирования:

- { – переместить курсор в начало текущего параграфа, который ограничивается пустой строкой;
- n**{ – переместить курсор в начало n-го параграфа, следующего за текущим параграфом;
- } – переместить курсор в начало следующего параграфа;
- n**} – переместить курсор в начало n-го параграфа ниже текущей строки.

2.4. Позиционирование курсора на неотображаемом тексте

Переместить курсор на текст, который не отображен на текущем экране, можно следующими способами:

1. Прокрутка текста

Четыре команды редактора **vi** позволяют осуществлять прокрутку текста файла. Команды **^f** (<CTRL>f) и **^d** (<CTRL>d) проводят прокрутку экрана вперед, а команды **^b** (<CTRL>b) и **^u** (<CTRL>u) – прокрутку экрана назад. Команда **^f** осуществляет прокрутку вперед на одно окно текста, расположенного ниже текущего окна. Редактор **vi** очищает экран и выводит новое окно. Две последние строки текущего окна отображаются в верхней части нового окна. Если окно дополнено пустыми строками, то они помечаются знаком ~ (тильда).

Команда **^d** осуществляет прокрутку на половину экрана, чтобы отобразить текст, расположенный ниже окна. Если курсор находится на последней строке файла и вы нажмете **^d**, то **vi** подаст звуковой сигнал.

Команда **^b** осуществляет прокрутку экрана назад на полное окно, чтобы отобразить текст, который находится выше текущего экрана. Редактор **vi** очи-

щает экран и отображает новое окно. В отличие от команды **^f** команда **^b** не оставляет в новом окне строки из текущего окна.

Команда **^u** осуществляет прокрутку половины экрана, чтобы отобразить текст, расположенный выше окна. Если курсор достигнет начала файла, то **vi** подаст звуковой сигнал.

2. Переход на указанную строку

Команда **G** позиционирует курсор на указанную строку в окне. Если эта строка не находится в текущем экране, то **G** очищает экран и отображает окно, содержащее эту строку. Если не указана строка, то команда **G** перемещает курсор на последнюю строку файла:

G – переместить курсор на последнюю строку в файле;

nG – переместить курсор на строку *n*.

3. Основные операции с текстом

3.1. Создание текста

После запуска редактора **vi** пользователь сразу попадает в командный режим, и **vi** ожидает ввода команд. Чтобы создать текст, необходимо:

- ввести **a**, чтобы перейти к режиму ввода (не нажимая клавишу **<ENTER>**);
- ввести нужный текст;
- перейти на новую строку нажатием клавиши **<ENTER>**.

После завершения создания текста следует нажать клавишу **<ESCAPE>**, чтобы перейти из режима ввода к командному режиму. Теперь можно редактировать текст:

```
<a>  
Entereate some text<ENTER>  
in the sEntereeen editor<ENTER>  
and return to<ENTER>  
command mode.<ESC>
```

Звуковой сигнал после нажатия клавиши **<ESCAPE>** означает переход в командный режим. Нажатие клавиши **<ESCAPE>** во время нахождения в командном режиме никак не влияет на текст в файле, даже если нажать ее несколько раз.

В редакторе **vi** существуют две команды для добавления текста: **insert(i)** и **append(a)**. Чтобы добавить текст с помощью команды **I**, нажмите **i** и введите текст. Введенный текст появится на экране слева от символа, на который указывает курсор. Этот символ и все остальные справа от курсора будут передвинуты вправо, чтобы освободить место для нового текста. Редактор **vi** будет хранить символы, вводимые до тех пор, пока пользователь не нажмет клавишу

<ESCAPE>. Если необходимо, передвинутые символы будут перенесены на следующую строку.

Таким же образом можно использовать команду **a**. Отличие состоит в том, что новый текст появится справа от символа, к которому подведен курсор.

3.2. Удаление текста

Удалить текст можно с помощью нескольких команд в командном режиме, а небольшой объем – в режиме ввода.

Удаление текста в режиме ввода текста

Если необходимо удалить символ в режиме ввода текста, воспользуйтесь клавишей <BACKSPACE>, которая перемещает курсор назад в режиме ввода текста и удаляет каждый символ. Однако удаленный символ не стирается с экрана до тех пор, пока пользователь не нажмет клавишу <ESCAPE> для возврата в командный режим.

Также удалить текст можно с помощью следующих ключей:

^w – отменить ввод текущего слова;

@ – удалить на текущей строке весь введенный во время данного режима ввода текст;

^v – удалить специальное значение, если оно есть, следующих вводимых символов.

Когда пользователь вводит **^w**, курсор возвращается на последнее введенное слово и ожидает на первом символе этого слова. Он не стирает слово с экрана до тех пор, пока не будет нажата клавиша <ESCAPE> или не будут введены новые символы на месте этого слова.

Отмена последней команды

Отменить последнюю введенную команду можно с помощью следующих команд:

u – отменить последнюю команду;

U – восстановить текущую строку в состояние до произведенных изменений.

Если пользователь случайно удалил строки, то для отмены команды удаления следует ввести команду **u**. Удаленные строки вновь появятся на экране. Также команда **u** отменяет и ввод неверной команды. Команда **U** аннулирует все изменения в текущей строке.

Ввод команды **u** во второй раз отменит первый ввод. Например, если пользователь удалил по ошибке строку и восстановил ее с помощью команды **u**, то нажав второй раз **u**, он вновь удалит строку.

Удаление объектов в командном режиме

Многие команды редактора **vi**, такие как команды удаления и изменения, позволяют использовать в командной строке команды перемещения курсора.

Команды перемещения курсора могут указывать на объект текста: слово, строка, предложение, параграф. Команды удаления, выдаваемые в командном режиме, сразу же удаляют указанный текст с экрана и оставшуюся часть отображают на экране в обновленном виде. Команда удаления имеет следующий формат:

[number]dtext_object

где **d** – команда удаления;
text_object – объект удаления.

Удаление слов. Удалить слово или часть слова можно с помощью команды **d**. Следует переместить курсор к первому удаляемому символу и ввести команду **dw**. Символ, расположенный над курсором, и последующие за ним будут стерты. Команда **dw** удаляет одно слово или знак пунктуации и пробелы, которые следуют за ним. Можно удалить несколько слов или знаков пунктуации за один раз, если указать перед командой соответствующее число. Например, чтобы удалить 3 слова и две запятые, следует ввести команду **5dw**.

Удаление параграфов. Чтобы удалить параграф, следует воспользоваться командой **d{** или **d}**. Следует помнить, что можно восстановить удаленный текст с помощью команды **u**.

Удаление строк. Чтобы удалить строку, следует ввести:

dd

Для удаления нескольких строк необходимо указать перед командой количество удаляемых строк. Например, команда

10dd

сотрет 10 строк.

Если необходимо удалить несколько строк, то **vi** отображает в нижней части экрана сообщение типа:

10 lines deleted

Если ниже текущей строки осталось меньше 10 строк, то прозвучит сигнал и строки не удалятся.

Удаление текста после курсора. Для удаления всего текста на строке, расположенного после курсора, следует поместить курсор на первый удаляе-

мый символ и ввести **D** или **d\$**. Эта команда удаляет текст только на текущей строке.

3.3. Поиск и замена текста

Поиск и замена слов

Следующие команды позволяют осуществить поиск и замену слов:

- :g/pattern/command<ENTER>** – для каждой строки, содержащей **pattern**, выполнить команду **command** редактора. Построчный редактор напечатает две строки, содержащие **pattern**;
- :s/pattern/new_words<ENTER>** – это команда замены. Построчный редактор отыскивает первое появление символов **pattern** в текущей строке и заменяет его на **new_words**;
- :s/pattern/new_words/g<ENTER>** – редактор заменяет каждое появление **pattern** в текущей строке;
- :g/helix/s//double helix/g<ENTER>** – эта команда осуществляет поиск слова **helix**. Каждое найденное слово **helix** заменяется на **double helix**. Ограничители после команды «s» позволяют не вводить слово **helix** каждый раз. Команда запоминает слово между ограничителями после команды «g»;
- :s/text/new_words/<ENTER>** – заменить первое появление символов **text** в текущей строке на **new_words**;
- :s/text/new_words/g<ENTER>** – заменить каждое появление символов **text** в текущей строке на **new_words** ;
- :g/text/s//new_words/g<ENTER>** – заменить каждое появление символов **text** в файле на **new_words**.

Один из способов модификации текста представляют команды удаления и команды ввода текста. Другой способ, с помощью которого можно изменить текст, это задать команду, позволяющую одновременно удалять и создавать текст. В редакторе **vi** существуют 3 основные команды изменения текста: **r**, **s** и **c**.

Замена символов

- rx** – заменить текущий символ (символ, на который показывает курсор) на **x**. Эта команда не устанавливает режим ввода текста и, следовательно, нет необходимости нажимать после нее клавишу <ESCAPE>;
- nrx** – заменить **n** символов на **x**. После этой команды также не надо нажимать клавишу <ESCAPE>;
- R** – заменяет только те символы, которые введены в режиме <ESCAPE>. Если будет достигнут конец строки, то эта команда добавит ввод как новый текст.

Например, необходимо заменить слово **acts** на **ants** в следующем предложении:

The circus has many acts,

следует поместить курсор под буквой **c** в слове **acts** и ввести команду:

rn

Предложение будет иметь вид

The circus has many ants.

В следующем примере, чтобы заменить слово **many** на **7777**, следует поместить курсор под буквой **m** в слове **many** и ввести команду **4r7**. Команда **r** заменит слово **many** на четыре цифры **7**:

The circus has 7777 ants.

Подстановка символов

Команда подстановки текста заменяет символы, но затем позволяет продолжать ввод текста с того места, где была нажата клавиша **<ESCAPE>**:

s – удалить символ, на который указывает курсор, и добавить текст. После окончания ввода текста необходимо нажать клавишу **<ESCAPE>**;

ns – удалить **n** символов и добавить текст;

S – заменить все символы в строке.

Когда пользователь вводит команду **s**, последний символ в строке символов, который будет заменен, переписывается со знаком **\$**. Символ не стирается с экрана, пока пользователь не введет на его место новый символ или не нажмет клавишу **<ESCAPE>**.

Изменение текста

Команда подстановки заменяет символы. Команда изменения заменяет объект текста и затем продолжает добавлять текст с этого места до тех пор, пока пользователь не нажмет клавишу **<ESCAPE>**.

Команда изменения может содержать аргумент **text_object**. Ниже приводятся команды замены:

ncx – замена **n**-го объекта текста типа **x** (предложение, параграф);

cw – замена слова или оставшихся символов в слове на новый текст. Редактор **vi** печатает знак **\$**, чтобы показать последний символ, который будет заменен;

ncw – заменить **n** слов;

cc – заменить все символы в строке;

ncc – заменить все символы в текущей строке и **n** верхних строк;

C – заменить оставшиеся символы в строке от курсора до конца строки;

dfn nC – заменить оставшиеся символы, начиная от курсора в текущей строке и во всех **n** верхних строках. Команда **c** использует знак **\$**, чтобы отметить последний символ, который должен быть замещен;

- r** – заменить текущий символ;
- R** – заменить те символы, которые введены в режиме ESCAPE;
- s** – удалить символ, под которым расположен курсор, и добавить текст;
- S** – заменить все символы в строке;
- псх** – заменить n объектов текста типа x (предложение, параграф).

Поиск символов по шаблону: команды / и ?

Для поиска символов по шаблону необходимо использовать команды /, ?, **n** и **N**. Эти команды позволяют осуществлять поиск вперед и назад в буфере следующего появления указанного шаблона символов. Команды **n** и **N** позволяют повторить последний поиск.

Команда / и следующий за ней шаблон (/pattern) ведут поиск вперед по буферу следующего появления символов, указанных в шаблоне, и помещают курсор на первый символ из шаблона. Например, командная строка:

/Hello world<ENTER>

находит первое появление в буфере слов Hello world и помещает курсор под буквой H.

Команда ? и следующий за ней шаблон (?pattern) ведет поиск в обратном направлении в буфере первого появления символов, указанных в шаблоне, и помещает курсор на первый из этих символов. Например, командная строка:

?data set design<ENTER>

находит предыдущее появление в буфере слов data set design и помещает курсор под буквой d в слове data.

Эти команды поиска не выполняют циклический возврат в случае окончания строки во время поиска двух или более слов. Например, если необходимо найти появление Hello world и окажется, что слово Hello находится в конце одной строки, а слово world – первое слово последующей строки, то эти команды не найдут Hello world. В то же время эти команды поиска выполняют циклический возврат в конце или начале буфера для продолжения поиска. Например, если курсор находится почти в конце буфера, а символы, которые соответствуют шаблону, – в начале буфера, то команда поиска / найдет шаблон.

Команды **n** и **N** позволяют повторить предыдущий запрос поиска /pattern или ?pattern:

- n** – повторить последнюю команду поиска;
- N** – повторить последнюю команду поиска в обратном направлении.

Команды / и ? не дают возможности указать конкретное появление шаблона с помощью числа, то есть, нельзя, например, запросить поиск 3-го появления шаблона, начиная с текущей позиции.

3.4. Извлечение и присоединение текста

Редактор **vi** предоставляет набор команд, которые разбивают текст на части, и набор команд, которые копируют часть текста и помещают в другое место в файле.

Перемещение текста

Текст можно переместить из одного места в буфере редактора **vi** в другое, удалив строки, а затем поместив их в требуемое место. Удаленный текст сохраняется во временном буфере. Переместив курсор на то место, куда необходимо поместить текст, следует нажать клавишу «**p**». Удаленные строки будут добавлены ниже текущей строки.

Удаленная с помощью команды удаления строка может быть помещена в середине другой строки. Следует поместить курсор между двумя словами, затем нажать клавишу «**p**». Удаленная строка будет помещена после курсора.

Так как временный буфер сохраняет результат только последней команды, то команда **p** должна использоваться сразу же после команды удаления. Команда **p** используется также для копирования текста, помещенного во временный буфер командой **y**.

Перемещение букв

Быстрым способом перемещения букв является комбинация команд **x** и **p** (**xp**). Команда **x** удаляет букву, а **p** помещает ее после следующего символа. Например, в следующем предложении есть ошибка:

A line of tetx

Следует поместить курсор под буквой «**t**» в комбинации «**tx**» и затем нажать клавиши «**x**» и «**p**». В результате получим:

A line of text

Копирование текста

При необходимости можно копировать одну или несколько строк текста во временный буфер и затем поместить копию этого текста в определенное место в файле. Чтобы поместить текст на новое место, следует ввести команду **y**. Команда **y** имеет следующий общий формат:

[number]y[text_object]

Копирование строк с помощью команды **y** не удаляет их с прежнего места в файле, т.е. можно поместить один и тот же кусок текста в нескольких местах:

- пх** – копирование n-го объекта текста типа х (предложение или параграф);
- уw** – создание копии слова;
- уу** – создание копии текущей строки;
- пуу** – копирование n строк;
- у)** – создание копии всего текста до конца предложения;
- у}** – создание копии всего текста до конца параграфа.

Следует обратить внимание, что эта команда позволяет указывать номер объекта текста, который будет копироваться.

3.5. Копирование и перемещение текста с использованием регистров

Перемещение и копирование нескольких разделов текста в различные части файла являются утомительной работой. Редактор **vi** предлагает использовать так называемые регистры, в которых можно хранить текст до тех пор, пока он вам не понадобится. Сохранить текст можно либо с помощью команды **у**, либо с помощью команды удаления. Использование регистров особенно полезно, если часть текста должна появиться во многих местах в файле. Извлекаемый текст остается в указанном регистре пока пользователь не завершит работу с ним, либо закончит работу с редактором, либо поместит в этот регистр другой текст.

Общий формат команды:

[number] ["X] command [text_object]

где **X** – имя регистра, состоящее из одной буквы верхнего регистра. Имени регистра должны предшествовать двойные кавычки.

Команды извлечения и присоединения текста:

- р** – помещает содержимое временного буфера, полученное с помощью команды **у** и команд удаления, в текст после курсора;
- уу** – создает копию строки текста и помещает во временный буфер;
- пх** – создает копию n-го объекта текста типа х и помещает ее во временный буфер;
- "хун** – помещает копию объекта текста типа n в в регистр с именем х;
- "хр** – помещает содержимое регистра х после курсора.

4. Специальные функции

В этом подразделе приведены специальные команды и функции редактора **vi**, необходимые пользователю.

Команда **.** (точка) повторяет последнюю введенную команду. Очень часто она используется с командой поиска.

Команда **j** объединяет строки. Чтобы ввести эту команду, следует поместить курсор на текущей строке и нажать клавиши **SHIFT** и **j** одновременно. Текущая строка объединяется со следующей строкой. Например, имеются две строки текста:

```
Dear Mr.  
Smith:
```

Для объединения этих двух строк в одну следует поместить курсор под любым символом в первой строке и ввести команду **j**. На экране появится строка:

```
Dear Mr. Smith:
```

Следует обратить внимание, что редактор **vi** автоматически ставит пробел между последним словом в первой строке и первым словом второй строки.

Чтобы быстро заменить букву нижнего регистра на букву верхнего регистра, надо поместить курсор под этой буквой и нажать клавишу «~» (тильда). Можно заменить несколько букв, нажав столько же раз клавишу «~», либо поставив перед тильдой число требуемых замен.

Чтобы добавить текст из файла ниже указанной строки в буфер редактора, следует воспользоваться командой **r**. Например, чтобы перенести содержимое файла **data** в текущий файл, следует поместить курсор на то место, где должен стоять текст, и ввести команду:

```
:r data<ENTER>
```

Вместо перемещения курсора можно указать номер строки. Например, чтобы вставить файл **data** после 56-й строки в буфере, следует ввести:

```
:56r data<ENTER>
```

4.1. Временный возврат к **shell**

Часто во время работы с редактором **vi** возникает необходимость временно вернуться в **shell**. Например, необходимо получить информацию из другого файла для присоединения ее к текущему тексту. Однако процедура завершения работы с редактором и повторный вход в редактор потребуют времени и сил. Редактор **vi** предлагает два способа временного выхода из редактора для того, чтобы ввести команды **shell** (и даже редактировать другие файлы), не покидая сам редактор. Это команды **:sh** и **:!.**

Команда **:!.** позволяет выйти из редактора и запустить команду **shell** в командной строке. Находясь в командном режиме редактора **vi**, необходимо ввести **:!.** Эти символы будут напечатаны внизу экрана. Сразу же после символа **!**

следует ввести команду **shell**, которая запустит команду, даст ответ и напечатает сообщение:

```
Hit return to continue
```

После того как пользователь нажмет клавишу <ENTER>, **vi** обновит экран и курсор примет свое прежнее положение.

Команда редактора **:sh** позволяет делать то же самое, но ведет себя по-другому. При вводе из командного режима редактора **vi** **:sh** и нажатия клавиши <ENTER> на следующей строке появится подсказка **shell**. Следует ввести команды после подсказки как в нормальном режиме **shell**. Когда необходимо вернуться к редактору **vi**, следует ввести **^d** или **exit**. Экран обновит информацию из буфера и курсор вновь переместится на прежнее место.

При временном переходе к **shell** даже изменение справочников не мешает возврату к буферу редактора **vi** при вводе **^d** или **exit**.

4.2. Запись текста в новый файл

Команда **w** позволяет создавать файл, копируя строки текста из файла, который в данный момент редактируется, в файл, который указан. Чтобы создать новый файл, нужно указать строку или строки (номера строк) и имя нового файла в командной строке. Общий формат этой команды:

```
:line_number[, line_number]w filename
```

Например, чтобы записать третью строку буфера в файл **three**, следует ввести

```
:3w three<ENTER>
```

При успешном создании нового файла редактор **vi** напечатает следующую информацию:

```
"three" [New file] 1 line, 20 characters
```

Чтобы записать текущую строку в файл, следует указать **.** (точку) как адрес строки:

```
:.w junk<ENTER>
```

Создастся новый файл с именем **junk**. Он будет содержать только текущую строку буфера. Например, чтобы строки с 23-й по 37-ю записать в файл, следует ввести следующую команду:

```
:23,37w newfile<ENTER>
```

4.3. Восстановление потерянного файла

Если случайно произошло прерывание работы редактора **vi**, то запись текста из буфера обратно в файл не производится. Но **vi** сохраняет копию буфера. Когда пользователь вновь регистрируется в системе UNIX, он может восстановить файл с помощью опции **-r**:

```
$ vi -r filename<ENTER>
```

Все или почти все изменения, которые были сделаны до прерывания, вновь появятся в буфере **vi**, и можно будет снова продолжить редактирование файла или записать буфер в файл и завершить работу с **vi**.

4.4. Редактирование нескольких файлов

Если необходимо редактировать в одном сеансе более одного файла, то следует ввести команду **vi**, перечислив имя каждого файла:

```
$ vi file1 file2<ENTER>
```

после чего **vi** ответит, сколько файлов будет редактироваться. Например:

```
2 files to edit
```

После завершения редактирования первого файла следует записать все изменения из буфера в файл **file1 :w<ENTER>**. Система ответит на эту команду сообщением внизу экрана. В нем она напечатает имя файла, число строк и символов в этом файле. Затем можно перейти к редактированию следующего файла, задав команду **:n<ENTER>**.

Система внизу экрана сообщит вам имя следующего файла, который вы будете редактировать, и число строк и символов в этом файле.

4.5. Просмотр файла

Один из способов просмотра файла – воспользоваться командами прокрутки и поиска. Однако можно защитить себя от случайного внесения изменений в файл во время сеанса редактирования. Чтобы воспользоваться опцией «только чтение», следует вызвать редактор следующим образом:

```
$ view file<ENTER>
```

4.6. Номера строк

Каждая строка файла имеет номер, соответствующий ее положению в буфере. Чтобы получить номер конкретной строки, переместите курсор на эту строку и введите команду **^g**.

В нижней части экрана команда **^g** поместит информацию:

- имя файла;
- был ли буфер модифицирован после последней записи его в файл;
- номер строки, на которой находится курсор;
- общее количество строк в буфере.

5. Порядок выполнения работы

1. Изучить методические указания.
2. После предварительной беседы с преподавателем получить у него конкретное задание и номер рабочего места.
3. После запуска сервера преподавателем (лаборантом) приступить к конкретному выполнению задания.
4. После окончания выполнения задания предъявить результаты преподавателю (в электронном либо бумажном виде).
5. Выйти из системы, сдать рабочее место.

6. Содержание отчета

Отчет должен содержать:

1. Описание процедуры входа в систему и регистрации.
2. Описание конкретного задания по набору и форматированию текста.
3. Текст созданных файлов.
4. Описание работы с файлами – создание, копирование, пересылка по электронной почте и т.д.
5. Пункты, конкретно заданные преподавателем.

ЛИТЕРАТУРА

1. Банахан М., Ратгер Э. Введение в операционную систему UNIX. – М.: Радио и связь, 1986. – 341 с.
2. Браун С. Операционная система UNIX. – М.: Мир, 1986. – 462 с.
3. Браун П. Введение в операционную систему UNIX. – М.: Мир, 1987. – 287 с.
4. Готье Р. Руководство по операционной системе UNIX. – М.: Финансы и статистика, 1985. – 232 с.
5. Керниган Б.В., Пайк Р. UNIX – универсальная среда программирования. – М.: Финансы и статистика, 1992. – 304 с.
6. Кристиан К. Введение в операционную систему UNIX. – М.: Финансы и статистика, 1985.
7. Немет Э. UNIX: руководство системного администратора. – Киев: BHV, 1997.
8. Петерсен Р. Linux: руководство по операционной системе. – Киев: BHV, 1997.
9. Тихомиров В.П., Давидов М.И. Операционная система UNIX: инструментальные средства программирования. – М.: Финансы и статистика, 1988. – 206 с.
10. Томас Р., Йейтс Дж. Операционная система UNIX: руководство для пользователей. – М.: Радио и связь, 1986. – 352 с.
11. Топхем Д., Чьюнг Х.В. UNIX и XENIX. – М.: Мир, 1988. – 392 с.
12. Ирвин Дж., Харль Д. Передача данных в сетях: инженерный подход: Учеб. пособие. – СПб.: BHV, 2003.
13. DRS/NX. Desktop Handbook, 1993.
14. DRS/NX. Software and Documentation, 1993.

Учебное издание

**КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ
В СЕТЯХ ТЕЛЕКОММУНИКАЦИЙ**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к лабораторно-практическим занятиям
для студентов специальности 45 01 03 «Сети телекоммуникаций»
дневной, вечерней и заочной форм обучения

В 3 частях

Часть 3

Операционная система DRS-Unix.
Работа с редактором VI и другими файлами ОС

С о с т а в и т е л и :

Мельников Владимир Александрович,
Рощупкин Яков Викторович

Редактор Н.А. Бебель
Корректор Е.Н. Батурчик
Компьютерная вёрстка Т.В. Шестакова

Подписано в печать 19.11.2003.
Печать ризографическая.
Уч.-изд. л. 1,0.

Формат 60x84 1/16.
Гарнитура «Таймс».
Тираж 50 экз.

Бумага офсетная.
Усл. печ. л. 1,395.
Заказ 221.

Издатель и полиграфическое исполнение:
Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Лицензия ЛП № 156 от 30.12.2002.
Лицензия ЛВ № 509 от 03.08.2001.
220013, Минск, П. Бровки, 6.