**Лекція № 8**

**3.1. Редактор vim**

Програма, яка призначена для редагування файлів, що містять тільки текст, називається **текстовим редактором**. У дистрибутивах системи Linux встановлено достатньо багато різних текстових редакторів, єдиним редактором, який гарантовано встановлений у будь-якій системі UNIX або Linux є редактор **vi** *(visual editor)*.

Перша версія цього редактора була написана Біллі Джоем, студентом Каліфорнійського університету в місті Берклі, який пізніше став співзасновником Sun Microsystems у 1976 р. Він призначався для редагування на екрані відеотермінала з можливістю переміщення курсору всією його поверхнею, саме від цього і походить його назва. До екранних редакторів існували малі редактори, які дозволяли редагування тексту тільки по одному рядку. З появою відеотерміналів стало можливим візуальне редагування на екрані.

Варто також зазначити, що на сьогодні більшість дистрибутивів Linux мають несправжній редактор **vi**, а його поліпшену заміну, що має назву **vim** (скорочено від *Vi IMproved* – Vi покращений), яка була створена Брамом Моленаром (Bram Moolenaar). **Vim** досконаліше традиційного редактора **vi** і в системах Linux зазвичай використовується під символічним посиланням (або псевдонімом) **vi**.

Для початку роботи з редактором **vi** у командній оболонці використовується однойменна команда: ***vi***.

При виконанні команди без аргументу на початку редагування відкриється редактор з порожнім файлом, а також виведена довідкова інформація щодо нього (рис. 3.1).



**Рис. 3.1.** Вікно редактора **Vim** з довідковою інформацією при створенні порожнього файлу для редагування

Таким чином, ми дізналися що у дистрибутиві Debian встановлено покращену версію редактора **vi** – **vim**, відкрити яку можна за допомогою команди ***vim***, а також псевдоніму ***vi***.

У якості аргументу до команди ***vi*** або ***vim*** можна задати назву файлу, який потрібно відкрити для редагування. Якщо такого файлу до цього не існувало, то він буде створений. У цьому випадку вікно редактора матиме вигляд, зображений на рис. 3.2:

***vi textfile1***



**Рис. 3.2.** Вікно редактора **vim** при відкритті порожнього файлу, назву якого задано в аргументі команди

Якщо задати у якості аргументу назву існуючого файлу, в якому до того ж збережена інформація, то він відкриється для редагування (рис. 3.3).



**Рис. 3.3.** Вікно редактора **vim** на початку редагування існуючого файлу

Однією з основних особливостей редактора **vim** (**vi**) є те, що він може працювати двома режимами: **командному режимі** та **режимі вставки**. Під час запуску він починає працювати у **командному режимі**, в якому можна задавати різні команди для редагування файлу, однак не вводити новий текст. Як правило, команди, які використовуються в командному режимі, складаються з одного або двох символів. Більшість команд будуть розглянуті нижче. Зокрема в **режимі вставки** відбувається основне редагування тексту.

Деякі досвідчені користувачі Linux також виділяють і третій режим редактора **vim** (**vi**) – **режим останнього рядка**. Фактично він є підрежимом режиму командного рядка, оскільки дозволяє ввести спеціальні команди, які починаються з символу‘**:**’ та зазвичай складаються з декількох символів, або навіть слів. Подібні команди відображаються на екрані в останньому рядку тоді як більшість команд, які не починаються з символу ‘**:**’, не відображаються. Типовими їх прикладами є команди виходу з редактора зі збереженням (**:wq**) та без збереження змін (**:q!**). Приклад режиму останнього рядка представлений на рис. 3.4.



**Рис. 3.4.** Режим останнього рядка в редакторі **vim** (**vi**)

**3.1.1. Робота редактора в режимі вставки**

Для початку редагування тексту необхідно перейти у режим вставки.

Такий перехід дозволяють здійснити три основні команди: **i** – розпочати вставку тексту на поточній позиції курсору; **a** – розпочати вставку після поточної позиції курсору; **o** – розпочати вставку на новому рядку нижче поточого абзацу (можна вводити команду, знаходячись у будь-якому місці поточного абзацу).

Кожна із зазначених команд також має модифікацію у вигляді команди з тією самою літерою, але у верхньому регістрі:

* **I** – розпочати вставку тексту на початку поточного абзацу перед першим ненульовим символом;
* **A** – розпочати вставку в кінці поточного абзацу;

 **O** – розпочати вставку на новому рядку нижче поточного абзацу (можна вводити команду, знаходячись у будь-якому місці поточного абзацу).

Будь-яка з описаних вище команд переведе редактор у режим вставки. Існують і інші команди, які переводять редактор у режим вставки (наприклад, під час заміни символу або слова), які будуть розглянуті далі. Для того, щоб повернутись у командний режим, потрібно натиснути клавішу **<ESC>**. Натиснення цієї клавіші у командному режимі, ніяк не впливає на текст у файлі, навіть якщо натиснути її декілька разів.

**3.1.2. Робота редактора у командному режимі**

У командному режимі редактора **vi** можна виконувати наступні дії: редагування та заміна існуючого тексту; видалення, переміщення та копіювання тексту; пересування всім файлом; інші завдання.



**Команди переміщення курсору**

Існує багато команд, що дозволяють зробити переміщення поточної позиції курсору від простих до більш складних.

Найбільш простим є переміщення курсору на сусідні позиції з 4 сторін. Для цього призначені наступні команди, які виконуються за допомогою клавіш, що розташовані поруч:

 **h** – перемістити курсор на один символ ліворуч; **j** – перемістити курсор вниз на один абзац; **k** – перемістити курсор вгору на один абзац; **l** – перемістити курсор на один символ праворуч.

Додамо, що команди переміщення курсору вниз і вгору (**j** та **k** відповідно) зберігають його позицію на абзаці. Наприклад, якщо курсор знаходився на 10-му символі абзацу, при здійсненні команди **j** або **k**, він переміститься на 10-й символ абзацу, що розташований нижче або вище відповідно. У випадку, якщо в абзаці, на який переходить курсор, кількість символів є меншою, то він переміщатиметься на останній символ абзацу.

Деякі термінали мають спеціальні символи управління курсором, помічені стрілками. Їх можна використовувати замість описаних вище команд. Однак більш універсальними все ж є команди, що представляють літеру. Зокрема, можна перемістити курсор більш ніж на один символ в одну зі сторін. Для цього перед відповідною клавішею, що означає команду переміщення курсору, необхідно вказати число, яке означає, на яку кількість символів потрібно перемістити курсор.

Наприклад:

 **5l** – перемістити курсор на 5 символів праворуч;

 **4k** – перемістити курсор на 4 абзаци вгору.

І так далі.

Якщо певна команда з числом не може перемістити курсор на зазначену кількість символів абзаців у відповідному напрямі, редактор подасть звуковий сигнал і курсор залишиться на колишньому місці.

Додатково для переміщення курсору ліворуч та праворуч використовують клавіші **<Spacebar>** та **<Backpace>**, причому разом з ними також можуть використовуватися числа, які вказують на скільки символів потрібно переміститися. Використання клавіш без чисел здійснить переміщення курсору на один символ у відповідному напрямі.

**Більш складні команди переміщення курсору**

Окрім переміщення курсору на сусідні символи також існують більш складні варіанти переміщення, яким відповідають спеціальні команди. Їх можна класифікувати таким чином:

 команди позиціонування за символами; команди позиціонування за абзацами (рядками); команди позиціонування за словами; команди позиціонування за реченнями; команди позиціонування за розділами (параграфами).



**Позиціонування за символами**

Крім розглянутих команд переміщення сусідніми символами, у **vi** існує багато додаткових команд переміщення на конкретний символ, якщо він не є сусіднім з поточною позицією курсору. Подібні команди дозволяють швидше переміщуватися текстом у межах файлу.

Розглянемо спочатку команди переміщення на певний символ в межах поточного рядка. Для цього призначені наступні команди:

 ***0*** (нуль) – перемістити курсор на перший символ в абзаці; ***^*** – перемістити курсор на перший ненульовий символ в абзаці;

 ***$*** – перемістити курсор на останній символ в абзаці.

Крім розглянутих вище команд, що є так званими командами безумовного переміщення символу, існують також спеціальні команди переміщення курсору на відповідний символ, який потрібно спочатку знайти. Якщо на поточному рядку шуканий символ не знайдено, редактор видасть звуковий сигнал попередження і курсор залишиться на місці. Для пошуку символу всередині рядка з подальшим переміщенням курсору редактор **vi** має 4 команди:

* ***f<x>*** – шукати зазначений символ ***x*** праворуч від поточної позиції

курсору та перемістити курсор на нього;

* ***F<x>*** – шукати зазначений символ ***x*** ліворуч від поточної позиції курсору та перемістити курсор на нього;
* ***;*** – повторити попередній пошук символу: запам’ятати попередній шуканий символ і шукати наступну появу символу в поточному абзаці праворуч з подальшим переміщенням курсору на нього;
* ***,*** – повторити попередній пошук символу, але у зворотному напрямі: запам’ятати попередній шуканий символ і шукати наступну появу символу в поточному абзаці ліворуч з подальшим переміщенням курсору на нього.

До того ж існують додаткові команди переміщення за символами вниз та вгору, які доповнюють команди **j** та **k**. Різниця тільки у тому, що ці команди позиціонують курсор на першому ненульовому символі. Такими командами є ***+*** та ***-***, які також можна використовувати з числами перед ними, що вказують кількість символів, на які потрібно зробити переміщення.

**Наприклад:**

* ***5-*** – перемістити курсор на 5 символів угору на ненульовий символ;
* ***7+*** – перемістити курсор на 7 символів вниз на ненульовий символ.

**Позиціонування за абзацами**

У редакторі **vi** існують також спеціальні команди, що дозволять перейти на потрібний абзац. Це наступні команди:

* ***gg*** – перемістити курсор на початок документу (перший символ

першого абзацу);

* ***G*** – перемістити курсор в кінець документу (перший символ останнього абзацу);
* ***<n>G*** – перемістити курсор на перший символ абзацу, порядковим номером якого в документі є ***n***.

**Позиціонування за словами**

З точки зору редактора **vi** словом є рядок символів, у який можуть входити букви, цифри та символи підкреслення. У редакторі є 6 команд позиціонування за словами, які у якості роздільника розглядають будь-який символ, який відрізняється від букви, цифри або символу підкресленення. Роздільниками слів є символи початку та кінця рядка, а також знаки пунктуації до, та після порожнього простору.

Відповідні команди позиціонування за словами:

 ***b*** – переміcтити курсор до першого символу попереднього слова; ***w*** – перемістити курсор на початок наступного слова;

 ***e*** – перемістити курсор до останнього символу наступного слова.

Перераховані вище команди можуть використовуватися з цілими числами попереду них. У цьому випадку буде здійснено переміщення курсору на відповідну кількість слів від його поточної позиції.

Для зазначених команд також існують модифікації, які у якості роздільників слів приймають тільки символ пробілу та переходу на новий рядок, ігноруючи при цьому всі знаки пунктуації. Такі команди представляють ті ж самі букви, але у верхньому регістрі: ***B***, ***W*** та ***E*** відповідно.

**Позиціонування за реченнями**

Редактор **vi** також розпізнає речення. У якості розділювачів речень можуть виступати символи ‘***.***’, ‘***!***’ або ‘***?***’. Однак, якщо ці символи з’являються всередині рядка, то за ними мають стояти два пробіли, щоб редактор **vi** міг їх розпізнати.

Командами позиціонування за реченнями є команди ***(*** та ***)***, які можна застосовувати з цілими числами перед ними, що буде означати переміщення на вказану кількість речень:

* ***(*** – перемістити курсор на початок поточного (попереднього) речення;
* ***)*** – перемістити курсор на початок наступного речення.

**Позиціонування за параграфами**

Редактор **vi** також вміє розпізнавати параграфи. Для цього вони мають бути розділені порожнім рядком. Для позиціонування курсору за параграфами використовуються команди ***{*** та ***}***, перед якими як і у випадках інших видів позиціонування можна вказати ціле число для переміщення на певну кількість параграфів:

 ***{*** – перемістити курсор на початок поточного параграфа;

 ***}*** – перемістити курсор на початок наступного параграфа.

**Позиціонування у вікні**

У редакторі **vi** є три досить цікаві команди, які дозволяють позиціонувати курсор у вікні. Ці команди представляють букви верхнього регістру:

 ***H*** – перемістити курсор до першого абзацу на екрані; ***M*** – перемістити курсор до середнього абзацу на екрані;

 ***L*** – перемістити курсор до останнього абзацу на екрані.

Усі команди переміщують курсор до першого символу відповідного абзацу.

Для пошуку символів за шаблоном ви можете використовувати команди ***/***, ***?***, ***n*** і ***N***, які дозволяють здійснювати пошук вперед і назад у буфері наступної появи зазначеного шаблону символів. Команди ***n*** і ***N*** дозволяють вам повторити останній пошук.

**Пошук символів та слів за шаблоном**

Окремим способом позиціонування курсору є пошук символів та слів за шаблоном. Для такого пошуку в редакторі **vi** використовуються команди ***/***,***?***, ***n*** та ***N***. Перші дві команди дозволяють здійснювати пошук наступної появи зазначеного шаблону символів у буфері вперед і назад, а останні дві повторити попередній пошук.

Команда ***/*** і наступний шаблон (***/pattern***) здійснює пошук появи символів, що зазначені у шаблоні та переміщує курсор на перший символ з відповідного шаблону. Наприклад, команда ***/hello world*** знаходить першу появу слів **‘hello world’** у буфері і переміщує курсор на букву **h**.

Відповідно команда ***?*** здійснює пошук шаблону в зворотному порядку. Наприклад, команда ***?data set design*** знаходить попередню появу слів **‘data set design’** і переміщує курсор під буквою **d** у слові **data**.

Важливим є те, що зазначені вище команди не виконують циклічного повернення у разі закінчення абзацу під час пошуку двох або більше слів. Наприклад, якщо потрібно знайти появу рядка ***„Hello world‟*** і виявиться, що слово ***„Hello‟*** знаходиться в кінці попереднього рядка, а слово ***„world‟*** є першим словом наступного, то ці команди відповідний рядок не знайдуть.

Водночас, зазначені команди пошуку виконують циклічне повернення в кінці або на початку буфера для продовження пошуку. Наприклад, якщо ми знаходимося майже в кінці буфера, а символи, які відповідають шаблону, на початку, то команда пошуку ***/*** знайде шаблон.

Натомість команди ***n*** та ***N*** дозволяють повторити попередній запит пошуку ***/pattern*** або ***?pattern***.

Варто звернути увагу на те, що з командами ***/*** та ***?*** не можна використовувати цілі числа на відміну від багатьох інших розглянутих команд. Тобто ми не можемо, наприклад, знайти третю порядкову появу шаблону, починаючи з поточної позиції.

**Команди модифікації тексту**

Одним зі способів модифікації тексту є команди видалення і команди введення тексту. Іншим способом модифікації є використання команди, яка дозволяє видалення і створення тексту одночасно. Це команди заміни, підстановки та зміни тексту.

**Заміна тексту**

Для заміни тексту в редакторі **vi** призначені наступні команди:

* ***r<x>*** – замінити символ, на який показує курсор, на символ **x**. Команда не переводить редактор у режим вставки, тому немає необхідності натиснення після неї клавіші ***<ESCAPE>***;
* ***<n>r<x>*** – замінити ***n*** символів на символ ***x***. Команда не переводить редактор у режим вставки, а отже не вимагає після її використання повернення у командний режим;
* ***R*** – перевести редактор у режим вставки, однак при введенні нового тексту символи, що знаходяться після курсору, не зміщуються, а замінюються новим текстом. Після завершення заміни цією командою потрібно повернутися у командний режим за допомогою клавіші ***<ESCAPE>***. Якщо під час редагування було досягнуто кінця рядка, то ця команда додасть введення як нового тексту.

**Приклади**

1. Замінити у реченні **‘The circus has many acts.’** слово **‘acts’** на

**‘ants’**.

Для цього потрібно перемістити курсор на букву ‘c’ у слові ‘acts’, а потім ввести команду ***rn***. Речення набуде наступного вигляду: **The circus has many ants.**

1. У тому самому реченні замінити слово **‘many’** на **‘7777’**.

Для цього переміщуємо курсор під буквою ‘m’ у слові ‘many’ і вводимо команду ***4r7***, яка виконає наступну заміну:

**The circus has 7777 ants.**

**Підстановка тексту**

Команди підстановки тексту замінюють символи, але потім дозволяють продовження введення тексту з того місця, на якому було натиснуто ***<ESCAPE>***. Є наступні команди підстановки тексту:

 ***s*** – видалити символ, на який вказує курсор і додати текст, тим

самим перевівши редактор у режим вставки;

 ***<n>s*** – видалити n символів і додати текст, так само перевівши редактор у режим вставки;

 ***S*** – замінити всі символи в рядку (абзаці). При цьому весь абзац буде видалений.

**Зміна тексту**

Команди зміни замінюють об’єкт тексту, а потім продовжують додавати його допоки не буде натиснута клавіша ***<ESCAPE>***, тобто всі відповідні команди як і команди підстановки також переводять редактор у режим вставки. Для зміни тексту призначені наступні команди:

 ***cw*** – замінити слово або символи, які залишились у слові, на новий

текст;

 ***<n>cw*** – замінити ***n*** слів;

 ***cl*** – замінити символ на новий текст; ***<n>cl*** – замінити ***n*** символів; ***cc*** – замінити всі символи в рядку; ***C*** – замінити решту символів у рядку від курсору до кінця рядка.



**Команди видалення тексту**

Редактор **vi** дозволяє видалити текст як у режимі вставки, так і у командному режимі.

Якщо редактор перебуває в режимі вставки, то для видалення символу потрібно скористатися клавішею ***<BACKSPACE>***, яка видалить символ ліворуч від курсору. Ця клавіша переміщує курсор назад під час введення літер і видаляє кожен символ, на який він переходить. У режимі вставки також можна застосувати комбінацію клавіш ***<CTRL>+<w>***, яка скасовує введення поточного слова.

Для скасування останньої введеної команди потрібно у командному режимі застосувати одну з наступних команд:

 ***u*** – скасувати останню команду;

 ***U*** – відновити поточний рядок у стан до проведених змін.

Для скасування випадково видалених рядків доцільно виконати команду ***u***, у результаті чого видалені рядки знову з’являться на екрані. На відміну від команди ***u***, команда ***U*** анулює всі зміни в поточному рядку, поки курсор на ньому буде перебувати.

Однак, потрібно зауважити, що при виконанні команди ***u*** двічі, друга команда скасує дію першої. Тобто при помилковому видаленні рядка з подальшим його відновленням командою ***u***, при виконанні другої команди ***u*** рядок знову буде видалений.

Для видалення певного тексту також призначено багато корисних команд, які потрібно виконувати у командному режимі. Вони одразу ж видаляють вказаний текст з екрана, а решту відображують в оновленому вигляді.

Для видалення тексту в командному режимі редактора **vi** існують наступні команди:

 ***x*** – видалити поточний символ;

 ***dw*** – видалити слово над курсором і наступний за ним пробіл, або

знак пунктуації;

 ***dW*** – видалити слово і знак пунктуації над курсором з наступним за

ним пробілом; ***dd*** – видалити поточний рядок;

 ***d)*** – видалити поточне речення від поточної позиції курсору до кінця; ***d}*** – видалити поточний параграф від поточної позиції до кінця;

 ***<n>d<x>*** – видалити ***n***-ий об’єкт тексту ***<x>***, у якості якого може бути: ***d*** – рядок; ***w*** – слово; ***)*** – речення; ***}*** – параграф;

 ***D*** – видалити частину рядка праворуч від курсору.

**Команди витягу і приєднання тексту**

Редактор **vi** також надає набір команд, які розбивають текст у файлі, копіюють частину тексту та поміщають його в інше місце у файлі.

**Переміщення тексту**

Існує можливість переміщення тексту з одного місця в буфері редактора **vi** в інше, спочатку видаливши рядки, а потім помістивши їх у потрібне місце. Видалений текст буде зберігатися у тимчасовому буфері. Перемістивши курсор на те місце, куди потрібно вставити текст, і виконавши команду ***p***, видалені рядки будуть додані нижче поточного рядка.

Дистанційний рядок може бути розміщений також усередині іншого рядка. При позиціонуванні курсору між двома словами та виконанні команди ***p*** дистанційний рядок буде поміщений після курсору.

Оскільки у тимчасовому буфері зберігається результат тільки останньої команди, команда ***p*** повинна використовуватися відразу ж після команди видалення. Вона використовується також для копіювання тексту, який поміщений у тимчасовий буфер командою ***y***, яка буде розглянута нижче.

**Переміщення букв**

Швидко перемістити букви можна використавши команду ***xp***, яка насправді представляє комбінацію команд ***x*** та ***p***: ***x*** видаляє букву, а ***p*** поміщує її після наступного символу.

Наприклад, у наступному реченні є помилка: **tetx**. Помістивши курсор під другою літерою **"t"**, та потім натиснувши почергово клавіші ***x*** і ***p***, у результаті отримаємо: **text**.

**Копіювання тексту**

Редактор **vi** також надає набір команд для копіювання тексту в тимчасовий буфер з подальшою його вставкою в інше певне місце у файлі. Для того, щоб помістити текст на нове місце, використовується команда ***y*** в своїх різних модифікаціях. Загальний синтаксис команди ***y*** є наступним: ***[<Число>]y[<Об‟єкт\_тексту>]***

Потрібно звернути увагу, що при копіюванні тексту за допомогою команди ***y***, він не видаляється з колишнього місця в файлі, тобто можна помістити один і той же фрагмент тексту одразу в декількох місцях.

Команда ***y*** в редакторі **vi** має наступні різновиди:

 ***<n>y<x>*** – копіювання ***n***-го об’єкта тексту типу ***<x>***: ***)*** – речення,

***}*** – параграф; ***yw*** – копіювання слова;

 ***yy*** – копіювання поточного рядка;

 ***<n>yy*** – копіювання ***n*** рядків;

 ***y)*** – копіювання всього тексту до кінця речення; ***y}*** – копіювання всього тексту до кінця параграфа.

Утім переміщення і копіювання кількох фрагментів тексту в різні частини файлу іноді є досить кропіткою роботою. Для спрощення таких дій редактор **vi** пропонує використання регістрів, на яких можна зберігати текст, допоки він не знадобиться. Зберегти текст можна за допомогою команди ***y***, або команд видалення. Використання регістрів є особливо корисним, якщо частину тексту потрібно вставити в багатьох місцях у файлі.

Загальний формат команди:

***[<Число>]["<X>]<команда>[<Об‟єкт\_тексту>]***

де ***<X>*** – назва регістра, що складається з однієї літери верхнього регістру. Його назві при цьому мають передувати подвійні лапки.

**Завершення роботи з редактором vim**

Для завершення роботи з редактором **vi** можна використати команди, які починаються зі знаку ‘***:***’, тобто відображуються на екрані у найнижчому рядку редактора у режимі останнього рядка. До того ж, після введення таких команд завжди потрібно натискати ***<ENTER>***.

Перерахуємо ці команди:

* ***ZZ*** – записати буфер в файл, якщо після останнього запису в буфері були зроблені зміни, і завершити роботу з **vi**;

 ***:wq*** – записати буфер у поточний файл і завершити роботу з **vi**;

 ***:w <filename>*** – записати вміст буфера в новий файл ***filename***;

* ***:w! <filename>*** – замінити вміст буфера в уже існуючий файл ***filename***;
* ***:q!*** – завершити роботу з **vi** без запису у файл, навіть якщо вміст буфера змінювався;
* ***:q*** – завершити роботу з **vi** без запису у файл, якщо вміст буфера не змінювався.

Остання з команд буде працювати у тому випадку, якщо у буфері не було ніяких змін. В іншому випадку редактор **vi** виведе попередження про те, що потрібно або зберегти буфер або для завершення роботи виконати команду: ***q!***.