

## ЛЕКЦІЯ 17. РОБОТА З ТЕКСТОВИМИ РЕДАКТОРАМИ VI I VIM

### Загальна характеристика текстових редакторів

Після основних утиліт для роботи з файлами й текстом перша програма, що знадобиться будь-якому користувачеві Linux – це текстовий редактор (для стислості – просто редактор).

Текстовий редактор буде потрібно користувачеві Linux у першу чергу для того, щоб змінити настроювання системи або свого оточення, наприклад, shell – при цьому потрібно буде редагувати конфігураційні файли, які завжди представлені в текстовому форматі. Але й власні завдання користувача можуть зажадати редагування текстових файлів: наприклад, сценарії й програми, електронні листи, а також замітки для себе, які пише Методій -все це дані в текстовому форматі. Текстові дані, отримані за допомогою стандартних утиліт, теж буває зручно зберігати у файлах і редагувати.

Не варто плутати текстові редактори й текстові процесори. Текстові процесори, наприклад OpenOffice Writer або Microsoft Word, призначені для створення документів, у яких, крім властиво тексту, утримується й різна метаінформація (інформація про оформлення): розміщення тексту на сторінці, шрифт і т.п. Оскільки в текстовому форматі не передбачено засобів для збереження інформації про оформлення (там є тільки символи й рядки), текстові процесори використовують власні формати для зберігання даних. Текст, у якому немає ніякої метаінформації про оформлення, називають "plain text" (тільки текст, "плоский", простий текст).

Однак за допомогою текстових редакторів можна працювати не тільки з форматом plain text. Різна метаінформація (про оформлення, спосіб використання тексту, наприклад, як посилання та ін.) може бути записана й у вигляді звичайних символів (тобто в текстовому форматі), але зі спеціальною угодою, що ці символи потрібно інтерпретувати особливим образом: як інструкції з обробки тексту, а не як текст. Такі інструкції називаються розміткою. У такий спосіб улаштований, наприклад, формат HTML. Для того щоб обробити розмітку HTML і відповідно до неї відобразити текст, потрібна спеціальна програма – броузер, але редагувати файли HTML та інші формати розмітки можна й за допомогою текстового редактора. Крім того, програми на будь-яких мовах програмування й сценарії (програми на shell) теж являють собою текстові файли. Багато текстових редакторів орієнтовані на роботу не тільки з "плоским" текстом, але й з текстом у різних форматах.

Найважливіша умова для *текстового редактора* в Linux – можливість працювати в терміналі, тому що це основний спосіб керування системою. Тому й введення даних, і редагування повинні повністю здійснюватися засобами терміналу, тобто алфавітно-цифровими й деякими функціональними клавішами. Оскільки функціональних клавіш, на які можна розраховувати на будь-якому терміналі, зовсім небагато, а команд, які потрібно віддавати редактору, – дуже багато, потрібен спосіб вводити будь-які команди обмеженими засобами терміналу. Це умова, так само як і вимога зручності при роботі з різноманітними структурованими текстами, виконане у двох "головних" *текстових редакторах* Linux – Vi і Emacs.

## Редактор vi

В Linux широко використовуються біля десятка текстових редакторів. Кожний з них гарний для своїх цілей. Але найбільше часто використовуваними все-таки є редактори vi і emacs.

Редактор vi - один з найстарших і наймогутніших редакторів в Linux. Він працює навіть на таких апаратних терміналах як VT100 і XTERM. Сучасний варіант цього редактора називається vim (Vi IMproved - поліпшений vi). vim емулює всі команди vi, і в той же час додає багато додаткових можливостей, таких як:

- багаторівневий процес скасування раніше виконаних дій (undo);
- використання декількох вікон редагування;
- редагування командного рядка;
- вбудована довідкова система (команда **:help**) і багато чого іншого. Ми будемо обговорювати саме vim - розширений варіант vi.

Під ім'ям Vi, насправді, може ховатися кілька різних програм: з моменту появи Vim в операційній системі UNIX (а це відбулося близько 30 років тому) цей редактор став чимсь начебто стандарту. До теперішнього часу існує ряд програм, або в точності повторюючих вид і поведінку "класичного" Vi (наприклад, nvi), або дуже схожих на нього, але зі значно розширеними можливостями (Vim, elvis). Найбільшою популярністю

користується Vim, можливості якого величезні - для їхнього опису треба було майже сто тисяч рядків документації. Коли користувач Linux набирає в командному рядку Vim, швидше за все, буде запущена "полегшена" версія Vi, що настроєна таким чином, щоб максимально відтворювати поведінку класичного редактора Vim. Природно, у такому режимі частина можливостей Vi недоступна. Всі властивості, який Vi відрізняється від Vim, обов'язково постачені в посібнику з Vi вказівкою "not in vi". Надалі викладі під Vi ми будемо розуміти саме Vim у режимі сумісності, всі можливості, недоступні в цьому режимі, будуть обмовлятися. Щоб викликати Vim у повнофункціональному режимі, досить набрати команду Vim.

Популярність Vi не випадкова: цей *текстовий редактор* дозволяє не тільки робити прості операції редагування текстових файлів – він добре пристосований і для максимально швидкого й ефективного рішення ряду суміжних з редагуванням завдань. Серед найважливіших його можливостей – інструменти для роботи з текстами на різних мовах програмування й у різних форматах *розмітки*. Vim уміє виділяти різними кольорами синтаксичні конструкції мови програмування або *розмітки*, автоматично виставляти відступи, що полегшує сприйняття структури документа. Крім того, в Vim є спеціальні засоби для організації циклу налагодження програми: компіляція-виправлення вихідного текст-компіляція.

## Запуск vi

Щоб почати сеанс редагування в vi, досить виконати команду vi на будь-якому терміналі. Щоб відкрити для редагування вже існуючий файл, *шлях* до

цього файлу потрібно вказати як параметр: "vi шлях\_до\_файлу". Як і всяка поважуюча себе програма UNIX, Vim може бути запущений з безліччю ключів, що модифікують його поведінку, які докладно описані в керівництві. Викликаний без параметра, редактор відкриє порожній *буфер* – чистий аркуш для створення нового тексту. У центрі екрана при цьому може з'явитися коротке привітальне повідомлення, де зазначені версія програми й команди для одержання допомоги й виходу з редактора (що нетривіально). Однак таке повідомлення може й не з'явитися – це залежить від версії Vi, встановленої в системі.

Для відображення тексту й роботи з ним Vi використовує весь екран термінала – тільки останній рядок призначений для діалогу з користувачем: виводу інформаційних повідомлень і введення команд. Поки *буфер* не заповнений текстом, на початку кожного рядка екрана відображається символ "~", що позначає, що в цьому місці *буфера* немає нічого, навіть порожнього рядка. Загальний вид екрана на початку роботи буде приблизно такою: #

Символ "#" позначає курсор. На екрані термінала вміщається більше рядків, але в прикладах ми будемо для компактності зображувати тільки необхідний мінімум.

### Режими

В Vi проблема поділу команд редактору й тексту, що вводиться, вирішена за допомогою режимів: у командному режимі натискання на будь-яку клавішу – це команда редактору, у режимі вставки натискання на клавішу приводить до вставки відповідного символу в редагується текст, що. Тому при роботі з Vi користувачеві завжди потрібно звертати увагу на те, у якому режимі перебуває редактор. Розрізняють три режими Vi:

- командний (натискання будь-якої клавіші вважається командою й негайно виконується),
- вставки (натискання клавіші-друкованого символу приводить до вставки цього символу в текст)
- командного рядка (для введення довгих команд, відображуваних на екрані; введення завершується Enter).

Редактор Vi завжди починає роботу в **командному** режимі. У цьому режимі є два способи віддавати команди редактору. По-перше, натискання практично на будь-яку клавішу редактор сприймає як команду. В Vim, навіть у режимі Vi-Сумісності, командне значення визначене для всіх латинських букв (у верхньому й нижньому *регістрах*), цифр, розділових знаків і більшості інших друкованих символів. При натисканні на ці клавіші вводяться команди, що, ніде не відображаються – вони просто **виконуються**.

По-друге, в Vi є свій командний рядок: щоб її викликати, потрібно ввести в командному режимі ":". У результаті на початку останнього рядка екрана з'явиться

двокрапка – це запрошення командного рядка. Тут уводяться більше складні команди Vi, які містять у собі цілі слова (наприклад, імена файлів), причому текст набирається команди, що, звичайно, відображається. Команди передаються Vi клавішею Enter. У

сучасних версіях Vim з командним рядком можна працювати так само, як і в shell: редагувати її, добудовувати команди клавішею Tab, користуватися історією команд.

Під час роботи з редактором часто виникає необхідність виконати команду оболонки без виходу з редактора. Vim надає таку можливість:

**!:команда – виконати команду “команда” !ls**

**– виконати команду ls**

Головна команда командного рядка Vim – виклик підсистеми допомоги "help Enter". Двокрапка переводить Vim у режим командного рядка, "help" – властиво команда, Enter передає команду. help можна викликати з аргументом: назвою команди або настроювання Vim. Vim дуже добре документований, тому по команді ":help об'єкт" можна одержати інформацію про будь-яку властивість Vim, наприклад, команда ":help i" виведе відомості про значення клавіші "i" у командному режимі Vi.

Команда ":set ім'я\_настроювання" дозволяє набудовувати Vim прямо в процесі роботи з ним. Наприклад, віддавши команду ":set wrap" користувач тим самим включає настроювання "wrap", що змушує редактор переносити занадто довгі рядки, що не вміщаються завширшки термінала. Виключити це настроювання можна командою ":set nowrap", так що кінці довгих рядків зникнуть за правим краєм екрана.

Нарешті, щоб увести текст, потрібно перейти з командного режиму в режим вставки, нажавши клавішу "i" (від "insert" – "вставка"). У цей момент в останньому рядку з'явиться повідомлення про те, що редактор перебуває в режимі вставки: " -- INSERT--" або " --ВСТАВКА --", залежно від установленної мови системних повідомлень.

У режимі вставки можна вводити текст, завершуючи рядок натисканням Enter.

Однак потрібно пам'ятати, що в деяких (ортодоксальних) версіях Vi у режимі введення не працюють ніякі команди переміщення по тексту – тут можна тільки набирати. Якщо ви помітили, що помилилися в наборі – не потрібно відразу переміщати курсор і намагатися виправити помилку: набагато зручніше буде внести всі виправлення потім, у командному режимі, де доступна безліч спеціальних команд швидкого переміщення й заміни тексту. Щоб перейти з режиму вставки назад у командний режим, потрібно натиснути ESC.

З командного режиму можна перейти в режим введення декількома командами. Різниця між ними полягає в тім, у якій крапці почнеться введення символів. Наприклад, по командах "O" і "o" ("open") можна вводити текст із нового рядка (до або після поточної), по команді "I" – з початку рядка, команди "a" і "A" ("append") відають додаванням символів (після курсору або в кінець рядка) і т.п.

## Робота з файлами

Редагуючи текст в Vi, користувач працює не безпосередньо з файлом, а з *буфером*. Якщо відкривається вже існуючий файл, Vi копіює його вміст у *буфер* і відображає *буфер* на екрані. Всі зміни, які робить користувач, відбуваються саме у вмісті *буфера* – відкритий файл поки залишається незмінним. Якщо ж Vi викликаний без параметра, то створюється порожній *буфер*, що поки не зв'язаний ні з яким файлом<sup>1</sup>.

Запуск редактора виробляється командою vim:

**vim <ім'я файлу>**

Якщо ви запустили vim без вказівки ім'я файлу, то Ви можете або почати редагування, або відкрити файл командою

**:edit повне\_ім'я\_файлу<Enter>.**

Зберегти поточний файл можна за допомогою команди

**:w <ім'я\_файлу><Enter>.**

Параметр “ім'я\_файлу” в команді збереження необов'язковий. Якщо він не зазначений, що файл буде збережений з тим же ім'ям, з яким і був відкритий. Якщо ж ви редагуєте новий файл, то ім'я вказати обов'язково.

Вийти з vim можна по команді

**:q<Enter>.**

Для того, щоб вийти з редактора, зберігши всі зроблені зміни, використовуйте комбінацію команд

**:qw <ім'я файлу><Enter>**

а для скасування збереження всіх зроблених змін

**:q!<Enter>.**

## Переміщення по тексту

При редагуванні в тексті завжди є *крапка*, у якій користувач "перебуває" у цей момент: символи, що вводяться із клавіатури, з'являться саме тут, віддалятися символи будуть теж звідси, до цієї крапки будуть застосовуватися команди редактора й т.п. Звичайно ця *крапка* позначається курсором, а в останній (інформаційної) рядку екрана Vim указує номер поточного рядка й колонки (номер символу в рядку), у яких перебуває курсор.

Для того щоб виконувати редагування тексту, по ньому необхідно переміщатися, тобто переміщати курсор. Самий очевидний спосіб це робити - скористатися клавішами зі стрілками. Натискання на одну із цих клавіш звичайно змушує курсор переміщатися на один символ уліво/вправо або на один рядок нагору/долілиць. Важко придумати більше неефективний і повільний спосіб переміщення, якщо потрібно потрапити на інший кінець об'ємного тексту, і навіть просте переміщення до початку або кінця рядка може зайняти кілька секунд.

Через те, що редактор забезпечує сумісність із різними типами терміналів, крім стандартних клавіш керування, він підтримує клавіші буквеної клавіатури для керування курсором:

перехід на рядок уверх - k  
 перехід на символ ліворуч — h  
 перехід на символ праворуч - l  
 перехід на рядок вниз — j

Такий режим переміщення працює тільки в режимі перегляду й режимі виділення. Для того, щоб довідатися, у якому місці файлу Ви зараз перебуваєте, натисніть

**<Ctrl+g>**. Унизу екрана з'явиться рядок стану, у якій буде зазначене ім'я файлу, що редагується, і поточна позиція в ньому.

Для переходу до кінця файлу, Вам необхідно натиснути комбінацію **<Shift+G>**. Якщо ж потрібно перейти до якогось певного рядка у файлі, натисніть **<№строки Shift+G>**, наприклад для переходу до 35 рядка: **<3><5><Shift+G>**.

Помітимо, що в процесі редагування тексту звичайно виникає необхідність переміщатися не в довільну *кранку*, а в деякі ключові: початок і кінець рядка, слова, пропозиції, абзацу, вираження, укладеного в дужки, цілого тексту. Особливо це помітно, якщо потрібно редагувати структурований текст: програму (наприклад, сценарій), *конфігураційний файл* і т.п. В Vi для кожного такого переміщення передбачені спеціальні команди, що звичайно складаються в натисканні однієї клавіші в командному режимі. Використовуючи їх, можна не тільки нажавши одну кнопку переміститися на будь-яку відстань у тексті, але й рухатися по структурних елементах, переходячи до попереднього/наступного слова, пропозиції, абзацу, дужці й т.д. Отже, пересунути курсор на початок поточного рядка можна командою "0", на перший непобельный символ у рядку – "^", у кінець рядка – "\$".

Абзацами Vi вважає фрагменти тексту, розділені порожнім рядком, до початку попередні/наступного абзацу можна потрапити командами "{" і "}" відповідно.

Дуже розповсюджене завдання – необхідність потрапити в самий кінець файлу: для цього служить команда "G" ("Go"), у самий початок – "gg".

Пересунути курсор уперед на початок наступного слова можна командою "w" (від "word", слово), на початок попереднього – "b" (від "backward", назад).

До початку попередні/наступних пропозицій можна переміститися командами "(" і ")" відповідно. Потрібно враховувати, що границі слів і пропозицій Vi знаходить по формальних ознаках (керуючись спеціально певними *регулярними вираженнями*), тому рішення Vi може іноді не збігатися з поданнями користувача про границі слів і пропозицій. Однак користувач завжди владний змінити відповідні *регулярні вираження*, подробиці – у документації по Vim.

В Vim ніколи не слід вручну повторювати ту саму команду: якщо потрібно перейти на три слова вперед, не слід тричі натискати "w" – для повторення команди

використовується *множник*. *Множник* – це будь-яке число, набране перед командою Vi: команда буде повторене відповідне число раз. Наприклад, "3w" – означає "тричі перемістити курсор на слово вперед", інакше кажучи, перемістити курсор на три слова вперед. Зверніть увагу, що *множники* можуть

застосовуватися не тільки з командами переміщення, але й з будь-якими командами Vi. Аналогічно можна перемістити курсор на 10 абзаців уперед командою "10}".

Не відразу очевидно, що пошук шаблону в тексті (рядка або регулярного вираження) – це теж команда переміщення. Як і будь-яке переміщення, пошук здійснюється в командному режимі: насамперед, потрібно натиснути "/" : в останньому рядку з'явиться символ "/". Далі варто ввести шаблон для пошуку - він буде відображатися в цьому рядку, його можна редагувати. За замовчуванням пошук регічезалежний (тобто маленькі й більші букви вважаються різними). Змінити це можна за допомогою команди: **:set ic<Enter>**. Підсвітити кольором всі шукані збіги можна командою **:set hls<Enter>**. Продовжити пошук до наступного входження нижче по тексту можна командою **n** а нагору по тексту **Shift+N**.

Звичайно Vi настроєний таким чином, що шаблон для пошуку інтерпретується як *регулярне вираження*, де ряд символів має спеціальне значення, це настроювання можна відключити ("**:set nomagic**"). Після того, як уведений шаблон, варто натиснути Enter – курсор переміститься до найближчого (далі по тексту) збігу із шаблоном. Пошук у зворотному напрямку (до попереднього збігу) варто починати з команди "?".

Зовсім просто перейти до наступного вживання в тексті того слова, на якому коштує курсор: для цього потрібно просто натиснути "\*" у командному режимі. Аналогічна команда

пошуку слова у зворотному напрямку – "#". Можна спеціально відзначити в тексті *кранку* й потім у будь-який момент вертатися до неї, як до закладки. Одну закладку визначає сам Vi – "`", місце в тексті, де була зроблена остання зміна. Докладно про ці й інші команди переміщення можна прочитати в посібнику з Vim по команді "**:help usr\_03.txt**".

### Зміна тексту

У командному режимі не можна вводити символи в текст із клавіатури, але змінити текст при цьому можна, наприклад, видаляючи символи. Для видалення символів у режимі введення команд, Вам необхідно натиснути **<Del>** або **<x>**, а для видалення відразу цілого рядка – **<dd>**. Якщо знадобилося видалити відразу кілька рядків, то перед введенням команди **<dd>**, необхідно ввести число рядків, що видаляються, (наприклад: **<5dd>** видалить 5 рядків, включаючи поточну долілиць по тексту). Для видалення всіх символів до кінця слова необхідно набрати команду **<dw>**, а до кінця рядка – **<d\$>**. Якщо ж необхідно видалити рядок до її початку – використовуйте комбінацію **<d^>**.

Для початку редагування (вставки тексту) необхідно натиснути **<Insert>** або **<i>**. Перевести vim у режим заміни символів у режимі редагування також може клавіша **<Insert>**, а у звичайному режимі - **<Shift+R>**. Вийти з режиму редагування можна за допомогою натискання **<Esc>**

Якщо в процесі редагування Ви зробили помилку, і Вам необхідно виконати відкат виконаних дій, то:

1. перейти в режим введення команд;
2. натиснути **<u>**.

Відкат виконаної операції також діє на операції видалення символів, рядків, слів і частин рядків. Для відкату відкату (повернення виконаних команд) використовується комбінація **<Ctrl+r>**.

При виконанні операцій видалення, вилучений текст міститься в буфер. Цей буфер можна вставити потім у будь-яке місце в редагується тексті, що. Робиться це натисканням клавіші **<r>**. Текст, що вставляється, буде поміщений після поточного рядка.

З командного рядка **Vi** можна виконати операцію пошуку й заміни: для простого рядка або для *регулярного вираження*. **Vim** дозволяє робити заміну:

**:s/було/стало/g** – зробити заміну в поточному рядку

**:1,20s/було/стало/g** – зробити заміну в рядках з 1 по 20-ю

**:%s/було/стало/g** – зробити заміну у всьому файлі

**:30,100s/було/стало/gc** – зробити заміну в рядках з 30 по 100 і

запитувати при кожній заміні дозволу зробити її.

Для зміни частини слова використовується команда **<cw>**. Вставши над словом, яких необхідно виправити, натисніть **<cw>**. Після цього всі знаки слова, починаючи з позиції курсору будуть вилучені й **vim** перейде в режим редагування.

Команда **<c>** використовується з тими ж параметрами, що й команда видалення, за винятком початку рядка:

- **<cw>** – редагувати до кінця слова
- **<c\$>** - редагувати до кінця рядка

Можна застосовувати цифри, наприклад:

- **<2cw>** - редагувати до кінця другого слова, включаючи поточне
- **<5c\$>** - редагувати до кінця п'ятого рядка, включаючи поточну.

### Настроювання **vi** і **Vim**

Вид і поведінка **Vi** і **Vim** можна істотно змінити за допомогою налаштувань, пристосовавши редактор саме до своїх смаків і звичок. Прямо під час роботи редактори можна міняти налаштування з командного рядка **Vi** за допомогою команди **":set ім'я\_налаштування"**. Крім того, можна зробити налаштування постійної, вписавши всі потрібні значення в *конфігураційний файл* **.vimrc** (або **.exrc** – для **Vi**) у домашньому каталозі користувача. При кожному запуску **Vim/Vi** читає цей файл і виконує всі команди, що втримуються в ньому. Обсяг лекції не дозволяє докладно зупинитися на налаштуванні **Vi**, читач може зайнятися вивченням цього питання сам: всі необхідні відомості є в руководствах. Щоб оцінити можливості налаштування, можна виконати в **Vim** (але не в **Vi**!) команду **":options"**, по якій буде виведений список всіх доступних опцій з коротким описом їхнього змісту.