**Bắt đầu chạy Bộ soạn thảo vi trong**[**Unix/Linux**](https://quantrimang.com/unix-va-linux)

Có những cách sau mà bạn có thể sử dụng để bắt đầu chạy Bộ soạn thảo vi:

|  |  |
| --- | --- |
| **Lệnh** | **Miêu tả** |
| **vi filename** | Tạo một file mới nếu nó đã không tồn tại, nếu không thì mở một file đang tồn tại. |
| **vi -R filename** | Mở một file đang tồn tại trong chế độ chỉ đọc. |
| **view filename** | Mở một file đang tồn tại trong chế độ chỉ đọc. |

Dưới đây là ví dụ để tạo một file mới **testfile** nếu nó đã không tồn tại trong thư mục làm việc hiện tại:

$vi testfile

Bạn có thể quan sát kết quả giống như sau trên màn hình:

| ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ "testfile" [New File]

Bạn sẽ chú ý một dấu ngã (~) trên mỗi dòng con trỏ. Dấu ngã này đại diện cho một dòng chưa được sử dụng. Nếu một dòng không bắt đầu với dấu ngã và xuất hiện khoảng trắng, có thể là một khoảng trống, tab, dòng mới hoặc một vài ký tự không nhìn thấy.

Vì thế bây giờ bạn đã mở một file để bắt đầu nó. Trước khi thực hiện các tiến trình xa hơn, chúng ta cùng tìm hiểu một vài khái niệm quan trọng được giải thích bên dưới.

**Các chế độ hoạt động trong Unix/Linux:**

Trong khi làm việc với bộ soạn vi, bạn sẽ tình cờ theo hai chế độ sau:

1. **Chế độ lệnh (Command mode)**: Chế độ này cho bạn có thể thực hiện các nhiệm vụ hành chính như lưu file, chạy lệnh, di chuyển con trỏ, cắt và dán các dòng hoặc từ, và tìm kiếm và đổi vị trí. Trong chế độ này, bất cứ cái gì bạn gõ vào được hệ thống biên dịch (hiểu) như một lệnh.
2. **Chế độ chèn (Insert mode)**: Chế độ này cho bạn có thể chèn văn bản vào trong file. Mọi thứ mà được gõ vào trong chế độ này được hệ thống hiểu như là một đầu vào và cuối cùng nó đặt vào trong file.

Bộ soạn thảo vi thường bắt đầu với chế độ lệnh. Để nhập văn bản, bản phải trong chế độ chèn. Để vào chế độ chèn, đơn giản bạn chỉ cần gõ**i.** Để thoát khỏi chế độ chèn, nhấn phím **Esc**, mà sẽ đặt bạn quay trở lại chế độ lệnh.

**Gợi ý**: Nếu bạn không chắc chắn về chế độ mà bạn đang ở trong, nhấn phím Esc hai lần, và sau đó bạn sẽ trong chế độ lệnh. Bạn mở một file bằng cách sử dụng Bộ soạn thảo vi và bắt đầu gõ một vài ký tự, sau đó bạn vào chế độ lệnh để thấy sự khác nhau.

**Thoát khỏi vi trong Unix/Linux**

Lệnh để thoát khỏi Bộ soạn thảo vi là**:q**. Một khi đang trong chế độ lệnh, gõ dấu hai chấm và q và dấu ! theo sau. Nếu file của bạn đã được chỉnh sửa theo cách nào đó, Bộ soạn thảo sẽ cảnh báo bạn về điều này và không cho bạn thoát khỏi nó. Để bỏ qua thông báo này, lệnh để thoát khỏi vi mà không lưu các chỉnh sửa là **:q**!. Lệnh này giúp bạn thoát khỏi vi mà không lưu bất cứ sự thay đổi nào mà bạn đã tạo ra tới file.

Lệnh để lưu nội dung của Bộ soạn thảo vi là **:w**. Bạn có thể so khớp lệnh trên với lệnh quit, hoặc :wq và dấu quay lại !.

Cách dễ nhất để lưu các thay đổi và thoát khỏi vi là lệnh **ZZ**. Khi bạn đang trong chế độ lệnh, gõ ZZ và nó sẽ làm việc tương tự như lệnh **:wq**.

Bạn có thể xác định tên các file khác nhau để lưu bằng cách xác định tên file sau :w. Ví dụ, nếu bạn muốn lưu file bạn đang làm việc thành một file khác có tên là filename2, bạn sẽ gõ :wfilename2 và dấu !. Bạn tự mình thử nhé.

**Di chuyển trong một file trong Unix/Linux**

Để di chuyển xung quanh trong một file mà không ảnh hưởng đến văn bản, bạn phải trong chế độ lệnh (nhấn phím Esc hai lần). Tại đây có một vài lệnh mà bạn có thể sử dụng để di chuyển xung quanh một ký tự tại cùng một thời gian.

|  |  |
| --- | --- |
| **Lệnh** | **Miêu tả** |
| **k** | Di chuyển con trỏ lên trên một dòng. |
| **j** | Di chuyển con trỏ xuống một dòng. |
| **h** | Di chuyển con trỏ sang trái một ký tự. |
| **l** | Di chuyển con trỏ sang phải một ký tự. |

Có hai điểm quan trọng mà bạn cần ghi nhớ:

Bộ soạn thảo vi là phân biệt chữ hoa và thường, vì thế bạn cần chú ý khi viết chữ hoa trong khi sử dụng lệnh.

Hầu hết các lệnh trong vi có thể được mở đầu bằng số lần bạn muốn hành động xảy ra. Ví dụ, **2j** là di chuyển con trỏ xuống 2 dòng.

Có nhiều cách khác để di chuyển trong một file trong vi. Nhớ rằng bạn phải trong chế độ lệnh (nhấn Esc hai lần). Dưới đây là một bảng liệt kê các lệnh khác mà bạn có thể sử dụng để di chuyển xung quanh file:

|  |  |
| --- | --- |
| **Lệnh** | **Miêu tả** |
| **0 or |** | Đặt vị trí con trỏ tại đầu dòng. |
| **$** | Đặt vị trí con trở ở cuối dòng. |
| **w** | Đặt vị trí con trỏ ở từ tiếp theo. |
| **b** | Đặt vị trí con trỏ ở từ trước. |
| **(** | Đặt vị trí con trỏ ở đầu câu văn hiện tại. |
| **)** | Đặt vị trí con trỏ ở đầu câu văn kế tiếp. |
| **E** | Di chuyển tới phần cuối của từ được giới hạn khoảng trắng. |
| **{** | Di chuyển tới đoạn văn sau. |
| **}** | Di chuyển về đoạn văn trước. |
| **[[** | Di chuyển tới khu vực sau. |
| **]]** | Di chuyển về khu vực trước. |
| **n|** | Di chuyển tới cột n trong dòng hiện tại. |
| **1G** | Di chuyển tới dòng đầu tiên của file. |
| **G** | Di chuyển tới dòng cuối cùng của file. |
| **nG** | Di chuyển tới dòng thứ n của file. |
| **:n** | Di chuyển tới dòng thứ n của file. |
| **fc** | Di chuyển thuận tiến tới c |
| **Fc** | Di chuyển ngược lại tới c |
| **H** | Di chuyển tới đầu của màn hình. |
| **nH** | Di chuyển tới dòng thứ n tính từ đầu của màn hình. |
| **M** | Di chuyển tới giữa màn hình. |
| **L** | Di chuyển tới cuối cùng màn hình. |
| **nL** | Di chuyển tới dòng thứ n tính từ cuối cùng của màn hình. |
| **:x** | Dấu hai chấm theo sau bởi một số sẽ đặt vị trí của con trỏ trên dòng số **x** |

**Các lệnh điều khiển trong Unix/Linux**

Các lệnh dưới đây hữu dụng mà bạn có thể sử dụng song song với phím control:

|  |  |
| --- | --- |
| **Lệnh** | **Miêu tả** |
| CTRL+d | Di chuyển tiến tới 1/2 màn hình. |
| CTRL+d | Di chuyển tiến tới 1/2 màn hình. |
| CTRL+f | Di chuyển tiến tới một màn hình đầy đủ. |
| CTRL+u | Di chuyển ngược về sau 1/2 màn hình. |
| CTRL+b | Di chuyển ngược về sau một màn hình. |
| CTRL+e | Di chuyển màn hình lên một dòng. |
| CTRL+y | Di chyển màn hình xuống một dòng. |
| CTRL+u | Di chuyển màn hình lên 1/2 trang. |
| CTRL+d | Di chuyển màn hình xuống 1/2 trang. |
| CTRL+b | Di chuyển màn hình lên một trang. |
| CTRL+f | Di chuyển màn hình xuống một trang. |
| CTRL+I | Kéo màn hình ngược lại. |

**Chỉnh sửa file trong Unix/Linux**

Để chỉnh sửa file, bạn cần trong chế độ chèn. Có nhiều cách để vào chế độ chèn từ chế độ lệnh.

|  |  |
| --- | --- |
| **Lệnh** | **Miêu tả** |
| **i** | Chèn văn bản trước vị trí con trỏ hiện tại. |
| **I** | Chèn văn bản ở phần đầu dòng hiện tại. |
| **a** | Chèn văn bản sau vị trí con trỏ hiện tại. |
| **A** | Chèn văn bản tại phần cuối của dòng hiện tại. |
| **o** | Tạo một dòng mới để nhập văn bản dưới vị trí con trỏ hiện tại. |
| **O** | Tạo một dòng mới để nhập văn bản trên vị trí con trỏ hiện tại. |

**Xóa các ký tự trong Unix/Linux**

Dưới đây liệt kê các lệnh quan trọng mà có thể được sử dụng để xóa các ký tự và các dòng trong một file đã mở:

|  |  |
| --- | --- |
| **Lệnh** | **Miêu tả** |
| **x** | Xóa một ký tự dưới vị trí con trỏ hiện tại. |
| **X** | Xóa một ký tự trước vị trí con trỏ hiện tại. |
| **dw** | Xóa từ vị trí con trỏ hiện tại tới từ kế tiếp. |
| **d^** | Xóa từ vị trí con trỏ hiện tại tới phần bắt đầu của dòng. |
| **d$** | Xóa từ vị trí con trỏ hiện tại tới phần cuối của dòng. |
| **D** | Xóa từ vị trí con trỏ hiện tại tới phần cuối của dòng hiện tại. |
| **dd** | Xóa dòng mà con trỏ hiện tại đang ở trên. |

Như đã đề cập ở trên, hầu hết các lệnh trong vi có thể được đặt ở trước bởi số lần bạn muốn hành động xảy ra. Ví dụ, **2x** sẽ xóa 2 ký tự dưới vị trí con trỏ hiện tại và 2dd sẽ xóa 2 dòng mà con trỏ đang ở trên đó hiện tại.

Chúng tôi đề nghị bạn thực hành các lệnh trên nhiều lần trước khi thực hiện các tiến trình phức tạp hơn.

**Các lệnh thay đổi trong Unix/Linux**

Bạn cũng có khả năng để thay đổi các ký tự, các từ,, hoặc dòng trong vi mà không cần xóa chúng. Dưới đây là các lệnh liên quan:

|  |  |
| --- | --- |
| **Lệnh** | **Miêu tả** |
| **cc** | Gỡ bỏ nội dung của dòng, làm cho bạn rời khỏi chế độ chèn. |
| **cw** | Thay đổi từ mà con trỏ đang ở tại đó, từ vị trí con trỏ tới vị trí chữ w thường cuối cùng của từ. |
| **r** | Đổi vị trí của ký tự dưới vị trí con trỏ. Vi trở lại chế độ lệnh sau khi sự đổi vị trí này được thực hiện xong. |
| **R** | Ghi đè nhiều ký tự bắt đầu với ký tự hiện tại ở dưới con trỏ. Bạn phải sử dụng phím **Esc** để dừng việc ghi đè này. |
| **s** | Đổi vị trí của ký tự hiện tại với ký tự mà bạn gõ vào. Sau đó, bạn bị rời khỏi chế độ chèn. |
| **S** | Xóa dòng mà con trỏ đang ở trên và thay đổi bằng một đoạn văn bản mới. Sau khi văn bản mới được nhập, vi vẫn giữ nguyên chế độ chèn. |

**Các lệnh sao chép và dán trong Unix/Linux**

Bạn có thể sao chép các dòng hoặc các từ từ một vị trí và sau đó bạn dán chúng tại vị trí khác bằng cách sử dụng các lệnh sau:

|  |  |
| --- | --- |
| **Lệnh** | **Miêu tả** |
| **yy** | Sao chép dòng hiện tại. |
| **yw** | Sao chép từ hiện tại từ ký tự chữ thường w con trỏ ở trên tới phần cuối của từ. |
| **p** | Dán bản sao sau vị trí con trỏ. |
| **P** | Dán bản sao trước vị trí con trỏ. |

**Các lệnh nâng cao trong Unix/Linux**

Dưới đây liệt kê một số lệnh nâng cao mà làm đơn giản hóa việc chỉnh sửa thường ngày và mang lại nhiều hiệu quả hơn trong việc sử dụng vi:

|  |  |
| --- | --- |
| **Lệnh** | **Miêu tả** |
| **J** | Nhập dòng hiện tại với dòng kế tiếp. |
| **<<** | Di chuyển dòng hiện tại sang trái một độ rộng shift. |
| **>>** | Di chuyển dòng hiện tại sang phải một độ rộng shift. |
| **~** | Chuyển kiểu của ký tự dưới vị trí con trỏ (VD: chữ hoa thành chữ thường ). |
| **^G** | Nhấn Ctrl+G cùng một lúc để chỉ trạng thái và tên file hiện tại. |
| **U** | Hồi phục dòng hiện tại trở về trạng thái trước khi con trỏ vào dòng đó. |
| **u** | Undo các thay đổi vừa thực hiện với file. Nhập u lần nữa sẽ redo sự thay đổi. |
| **J** | Nhập dòng hiện tại với dòng kế tiếp. |
| **:f** | Hiển thị vị trí hiện tại của file trong % và tên file, tổng số file. |
| **:f filename** | Đặt lại tên hiện tại của file. |
| **:w filename** | Ghi tên file |
| **:e filename** | Mở một file khác với tên của nó. |
| **:cd dirname** | Thay đổi thư mục làm việc hiện tại tới thư mục dirname. |
| **:e #** | Sử dụng để chuyển đổi giữa hai file đã mở. |
| **:n** | Trong trường hợp bạn mở nhiều file sử dụng vi, bạn sử dụng :n để tới file kế tiếp trong seri các file đó. |
| **:p** | Trong trường hợp bạn mở nhiều file sử dụng vi, bạn sử dụng :p để tới file phía trước trong seri file đó. |
| **:N** | Trong trường hợp bạn mở nhiều file sử dụng vi, bạn sử dụng :N để tới file phía trước trong seri file đó. |
| **:r file** | Đọc file và chèn nó sau dòng hiện tại. |
| **:nr file** | Đọc file và chèn nó sau dòng thứ n. |

**Tìm kiếm từ và ký tự trong Unix/Linux**

Bộ soạn thảo vi có hai loại tìm kiếm: chuỗi và ký tự. Để tìm kiếm chuỗi, lệnh / và ? được sử dụng. Khi bạn bắt đầu những lệnh này, lệnh vừa được nhập vào sẽ được chỉ ở cuối cùng của dòng, tại đó bạn nhập một chuỗi cụ thể mà bạn muốn tìm để tìm kiếm nó.

Hai lệnh này chỉ khác nhau ở phương diện nơi mà nó diễn ra:

Lệnh / tìm kiếm theo chiều thuần về phía trước trong một file.

Lệnh ? tìm kiếm theo chiều ngược lại về phía sau trong một file.

Các lệnh n và N tương ứng lặp lại lệnh tìm trước theo cùng cách tìm hoặc theo hướng ngược lại. Một vài ký tự có ý nghĩa quan trọng trong khi sử dụng lệnh tìm kiếm và được đặt trước bởi một dấu \ để trở thành một phần của biểu thức cần tìm.

|  |  |
| --- | --- |
| **Ký tự** | **Miêu tả** |
| **^** | Tìm tại đầu của dòng. (Sử dụng tại phần bắt đầu của một biểu thức tìm kiếm.) |
| **.** | So khớp với một ký tự đơn. |
| **\*** | So khớp với 0 hoặc nhiều ký tự trước. |
| **$** | Cuối của dòng. (sử dụng tại phần cuối của biểu thức tìm kiếm.) |
| **[** | Bắt đầu một thiết lập của biểu thức so khớp hoặc không so khớp. |
| **<** | Đặt biểu thức tìm kiếm tách khỏi dấu chéo ngược \ để tìm phần cuối hoặc phần đầu của từ. |
| **>** | Đặt biểu thức tìm kiếm tách khỏi dấu chéo ngược \ để tìm phần cuối hoặc phần đầu của từ. |

Hoạt động tìm kiếm ký tự tìm kiếm trong một dòng để tìm một ký tự được nhập sau lệnh. Lệnh f và F chỉ tìm kiếm cho một ký tự trên dòng hiện tại. Lệnh f tìm kiếm thuận về phía trước và F tìm kiếm ngược về phía sau và con trỏ di chuyển tới vị trí của ký tự được tìm thấy.

Lệnh t và T chỉ tìm kiếm một ký tự trên dòng hiện tại, nhưng với t, con trỏ di chuyển tới vị trí trước ký tự, và T tìm kiếm dòng về phía ngược lại tại vị trí sau ký tự.

**Các lệnh set trong Unix/Linux**

Bạn có thể tạo thay đổi màn hình hiển thị của vi bằng cách sử dụng lệnh**:set**. Để sử dụng những lệnh này, bạn phải vào chế độ lệnh sau đó gõ **:set** theo sau bởi một trong các chức năng sau:

|  |  |
| --- | --- |
| **Lệnh** | **Miêu tả** |
| **:set ic** | Bỏ qua kiểu chữ (chữ hoa, thường) khi tìm kiếm. |
| **:set ai** | Thiết lập chế độ thụt vào đầu dòng tự động. |
| **:set noai** | Để không đặt chế độ thụt vào đầu dòng tự động. |
| **:set nu** | Hiển thị các dòng với số dòng trên cạnh trái. |
| **:set sw** | Thiết lập độ rộng của điểm dừng tab của một phần mềm. Ví dụ bạn có thể thiết lập một độ rộng shift là 4 với lệnh này **:set sw=4** |
| **:set ws** | Nếu *wrapscan* được thiết lập, nếu từ không được tìm thấy tại phần cuối cùng của file, nó sẽ cố gắng để tìm kiếm tại phần bắt đầu. |
| **:set wm** | Nếu chức năng này có một giá trị lớn hơn 0, Bộ soạn thảo sẽ tự động bao lề ký tự (word wrap). Ví dụ, để thiết lập lề bao tới hai ký tự, bạn sẽ gõ thế này **:set wm=2** |
| **:set ro** | Thay đổi kiểu file thành "chỉ đọc (read only)" |
| **:set term** | In kiểu terminal. |
| **:set bf** | Bỏ qua việc điều khiển ký tự từ đầu vào. |

**Chạy các lệnh trong Unix/Linux**

Bộ soạn thảo vi có khả năng chạy các lệnh từ trong Bộ soạn thảo. Để chạy một lệnh, bạn chỉ cần tiến vào chế độ lệnh và nhập lệnh :!.

Ví dụ, nếu bạn muốn kiểm tra một file có tồn tại không trước khi bạn cố gắng để lưu file với tên đã cho, bạn có thể nhập :! ls và bạn sẽ thấy kết quả của lệnh ls trên màn hình.

Khi bạn nhấn bất kỳ phím nào, bạn được quay trở lại khu vực vi của bạn.

**Thay thế văn bản trong Unix/Linux**

Lệnh thay thế (**:s/**) cho bạn có thể nhanh chóng đổi các từ hoặc nhóm từ trong file của bạn. Dưới đây là cú pháp đơn giản:

:s/search/replace/g

Ký tự g đại diện cho toàn cầu (globally). Kết quả của lệnh này là tất cả các sự kiện diễn ra trên dòng mà con trỏ tại đó đã được thay đổi.

**Các điểm quan trọng trong Unix/Linux:**

Dưới đây là các điểm quan trọng giúp bạn thành công khi sử dụng vi:

Bạn phải trong chế độ lệnh để sử dụng các lệnh. (Nhấn phím Esc hai lần vào bất cứ lúc nào để đảm bảo bạn trong chế độ lệnh).

Bạn phải cẩn thận để sử dụng các kiểu chữ (hoa và thường) cho tất cả các lệnh.

Bạn phải trong chế độ chèn để nhập văn bản.