

Introduction to Vim, II

Introduction to Programming

EE231002

Sep. 23, 2019

Moving Cursor

- `vim` use keyboard as the input device for better efficiency
- In normal mode, one can use `h`, `j`, `k`, and `l` to move cursor
 - `h` = ←, `j` = ↓, `k` = ↑, `l` = →
- One can also use the following normal mode command to go to line number `n` directly
 - `:n`: go to line `n`, `n` is an integer
 - This is especially useful for debugging

Copy and Paste

- In normal mode, one can copy one line or multiple lines into the buffer
 - `yy`: copy the current line into the buffer
 - `nyy`: copy `n` lines, starting from the current line, into the buffer (`n` is an integer)
- The content of the buffer can be pasted following the current line by
 - `p`: paste the content of the buffer following the current line.

- In normal mode, the following are delete commands
 - `dd`: delete the current line and place it into the buffer
 - `n dd`: delete `n` lines, starting from the current line, and place it into the buffer
 - `D`: delete to the end of the line (from the cursor position) and place the content into the buffer
 - `dw`: delete the current word and place it into the buffer
- The content of the buffer can be pasted at the current position using
 - `p`: paste the content of the buffer at the current position.

Undo, Redo and Repeat Command

- In normal mode, delete (and other commands) can be **undo** by
 - `u`: undo command
- After undo, the original operation can be redone by
 - `Ctl-r`: redo command
- In normal mode, the same command can be repeated by
 - `.`: repeat command

Search for a Word

- In normal mode, one can search for a `word` by
 - `/word`: searches for `tkbword`
 - The cursor will be placed at the first `word` after the cursor.
 - To search for the next `word` use command
 - `n`: searches for the next `word`
 - To search for the previous `word` use command
 - `N`: searches for the previous `word`
- Note that `/word` searches the first `word` after the cursor, to search for the previous `work` use the following command
 - `?word`: search for the previous `word` before the cursor position
- Note that after searching command, all `word` will be highlighted
 - To turn off highlight, use the following command
 - `:noh`: turn off highlights

Search Example

```
~/u2019/PL/labs/vim2 — vi lab01.c — bash — 60x12

int main(void)
{
    int DegreeC, degreeF;    // store temperatures

    printf("Enter temperature in Celsius: "); // prompt
    scanf("%d", &DegreeC);           // read temp
    degreeF = DegreeC * 9.0 / 5.0 + 32.0; // conversion
    printf("Temperature in Fahrenheit: %d\n", degreeF);
    return 0;
}
```

13,1-4 Bot

- Search for `degreeC`

```
~/u2019/PL/labs/vim2 — vi lab01.c — bash — 60x12

int main(void)
{
    int degreeC, degreeF;    // store temperatures

    printf("Enter temperature in Celsius: "); // prompt
    scanf("%d", &degreeC);           // read temp
    degreeF = degreeC * 9.0 / 5.0 + 32.0; // conversion
    printf("Temperature in Fahrenheit: %d\n", degreeF);
    return 0;
}
/degreeC
```

Substitute a Word

- In normal mode, one can substitute `word1` by `word2` by
 - `:0,$s/word1/word2/g`
 - This command replaces all occurrences of `word1` by `word2`
 - `:m,ns/word1/word2/cgi`
 - This command replaces `word1` by `word2` from line `m` to `n`
 - `c`: confirmation
 - `g`: all occurrence, not the first one of each line
 - `i`: case insensitive

Substitute Example

```
~/u2019/PL/labs/vim2 — vi lab01.c — bash — 60x12

int main(void)
{
    int DegreeC, degreeF;    // store temperatures

    printf("Enter temperature in Celsius: "); // prompt
    scanf("%d", &DegreeC);                // read temp
    degreeF = DegreeC * 9.0 / 5.0 + 32.0;    // conversion
    printf("Temperature in Fahrenheit: %d\n", degreeF);
    return 0;
}
```

13,1-4 Bot

- Substitute `degreeC` by `DgC`

```
~/u2019/PL/labs/vim2 — vi lab01.c — bash — 60x12

int main(void)
{
    int DgC, degreeF;    // store temperatures

    printf("Enter temperature in Celsius: "); // prompt
    scanf("%d", &DgC);                // read temp
    degreeF = DgC * 9.0 / 5.0 + 32.0;    // conversion
    printf("Temperature in Fahrenheit: %d\n", degreeF);
    return 0;
}
:0,$s/degreeC/DgC/g
```

Read and Write

- In normal mode, the file can be saved to a different name

- `:w filename`: save to `filename` file

```
~/u2019/PL/labs/vim2 — vi lab01.c — bash — 60x7
printf("Enter temperature in Celsius: "); // prompt
scanf("%d", &DgC);                       // read temp
degreeF = DgC * 9.0 / 5.0 + 32.0;         // conversion
printf("Temperature in Fahrenheit: %d\n", degreeF);
return 0;
}
:w test.c
```

- One can also read in other file's content and place after current line

- `:r filename`: read `filename`

```
~/u2019/PL/labs/vim2 — vi lab01.c — bash — 60x7
// EE231002 lab02 Day of Week
// Mi-Chang Chang
// Sep. 23, 2019

#include <stdio.h>

:r ../lab01/lab01.c
```

Swap File Error Example

- vim error?

```
michang — ssh ee231002@140.114.24.112 — 80x23
E325: ATTENTION
Found a swap file by the name ".lab01.c.swp"
    owned by: ee231002   dated: Fri Sep  9 07:19:36 2016
    file name: ~ee231002/C_program/lab01/lab01.c
    modified: no
    user name: ee231002   host name: ws38
    process ID: 1907 (still running)
While opening file "lab01.c"

(1) Another program may be editing the same file.
    If this is the case, be careful not to end up with two
    different instances of the same file when making changes.
    Quit, or continue with caution.

(2) An edit session for this file crashed.
    If this is the case, use ":recover" or "vim -r lab01.c"
    to recover the changes (see ":help recovery").
    If you did this already, delete the swap file ".lab01.c.swp"
    to avoid this message.

Swap file ".lab01.c.swp" already exists!
[O]pen Read-Only, (E)dit anyway, (R)ecover, (Q)uit, (A)bort:[]
```

Swap File

- When editing a file `file`, a temporary file `.file.swp`, is created
 - `.file.swp` is deleted after successful quitting from vim program
 - If vim is abnormally terminated, `.file.swp` remains
 - The next time `vim file` will have an error
 - This `.file.swp` should be removed using linux command
 - `rm .file.swp`



A terminal window showing the following commands and output:

```
michang — ssh ee231002@140.114.24.112 — 80x12
[ee231002@ws38 lab01]$ vim lab01.c
[ee231002@ws38 lab01]$ rm .lab01.c.swp
[ee231002@ws38 lab01]$ vim lab01.c
```