

Introduction to Vim, I

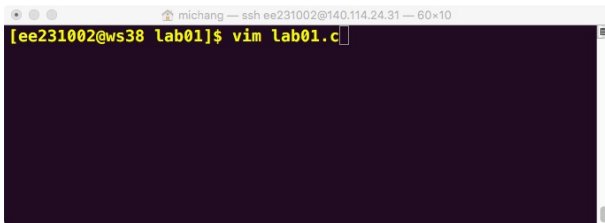
Introduction to Programming

EE231002

Sep. 18, 2017

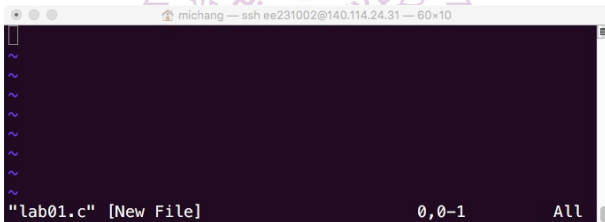
Starting vim

- To start vim: `vim file`



A terminal window with a dark background. The title bar shows "michang — ssh ee231002@140.114.24.31 — 60x10". The prompt is "[ee231002@ws38 lab01]\$". The command "vim lab01.c" has been entered, and the cursor is at the end of the line.

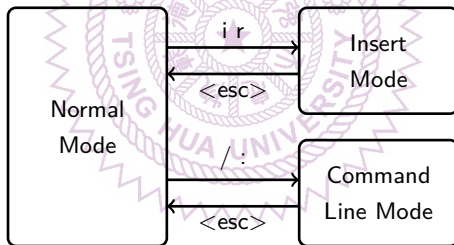
- For a new file



A terminal window with a dark background. The title bar shows "michang — ssh ee231002@140.114.24.31 — 60x10". The prompt is "[ee231002@ws38 lab01]\$". The command "vim" has been entered, and the cursor is at the end of the line. The terminal shows several blue tilde characters (~) representing empty lines. At the bottom, it says "\"lab01.c\" [New File] 0,0-1 All".

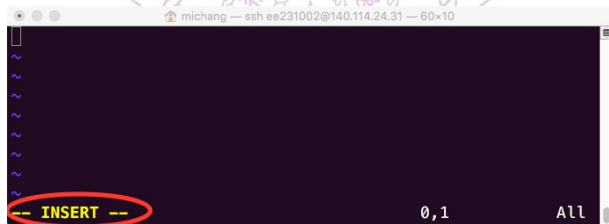
Three Modes in vim

- There are three modes in `vim`
 - Normal mode: copy, delete, paste
 - Insert mode: insert text
 - Command line mode: save file, exit, search and replace



Inserting Text

- When `vim` starts, it enters normal mode
- Press `i` to enter insert mode
 - Note the `-- INSERT --` on the lower-left corner
 - You can type in C program at this time



Insert Mode

- In insert mode, you can type in texts
- To move cursor
 - `↑`, `↓`, `←`, `→` keys move cursor in four directions
 - `PgUp` and `PgDn` keys scroll one page of text
 - `Home` key moves cursor to the beginning of the line
 - `End` key moves cursor to the end of the line
 - `Tab` key moves cursor to fixed columns (4x or 8x)
 - In our labs please use `Tab` key for indentation and each `Tab` key moves 4 spaces
- Press `Esc` key to return to normal mode

Quitting vim

- In normal mode, the following commands save file or quit `vim` program
 - `:w`: save typed inputs to the file
 - `:q`: quit `vim` program (no saving file)
 - `:q!`: forced quitting from `vim` program
 - Changes are not updated to the file
 - `:wq`: save file and then quit `vim` program
 - `ZZ`: same as `:wq` but is a normal mode command
- Note that that the above except `ZZ` are executed in command line mode



```
michang — ssh ee231002@140.114.24.112 — 60x10
{
    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;
}
:wq
```

Show Line Numbers in vim

- `vim` does not show line numbers by default
 - Line numbers are very useful in debugging compiler errors
 - To show line number type in `:set nu` in normal mode

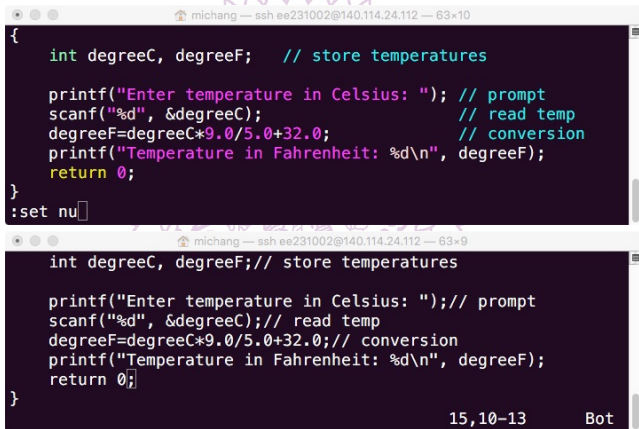
```
michang — ssh ee231002@140.114.24.112 — 63x10
{
  int degreeC, degreeF; // store temperatures

  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;
}
:set nu
```

```
michang — ssh ee231002@140.114.24.112 — 63x10
8 {
9   int degreeC, degreeF; // store temperatures
10
11  printf("Enter temperature in Celsius: "); // prompt
12  scanf("%d", &degreeC); // read temp
13  degreeF=degreeC*9.0/5.0+32.0; // conversion
14  printf("Temperature in Fahrenheit: %d\n", degreeF);
15  return 0;
16 }
15,10-13 Bot
```

Color Text

- `vim` takes advantage of the color terminal to make the file more legible
- The text color can be turned off by using `:syntax off` command
- `:syntax on` turns on color text



The image shows two screenshots of a terminal window. The top screenshot shows the same code as the bottom one, but with syntax highlighting. The bottom screenshot shows the code without syntax highlighting, demonstrating the effect of the `:syntax off` command. The terminal window title is "michang — ssh ee231002@140.114.24.112 — 63x10".

```
{
  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;
}
:set nu
```

```
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;
}
```

15,10-13 Bot

- This is the mode that I use to view your program
 - Be sure your program is very legible to me in this mode

Color Text, II

- Depending on terminal background, the text color may need to be adjusted

- `:set bg=dark`



```
michang — ssh ee231002@140.114.24.112 — 63x10
8 {
9     int degreeC, degreeF;    // store temperatures
10
11     printf("Enter temperature in Celsius: "); // prompt
12     scanf("%d", &degreeC);           // read temp
13     degreeF=degreeC*9.0/5.0+32.0;    // conversion
14     printf("Temperature in Fahrenheit: %d\n", degreeF);
15     return 0;
16 }
```

15,10-13 Bot

- `:set bg=light`



```
michang — ssh ee231002@140.114.24.112 — 63x10
8 {
9     int degreeC, degreeF;    // store temperatures
10
11     printf("Enter temperature in Celsius: "); // prompt
12     scanf("%d", &degreeC);           // read temp
13     degreeF=degreeC*9.0/5.0+32.0;    // conversion
14     printf("Temperature in Fahrenheit: %d\n", degreeF);
15     return 0;
16 }
```

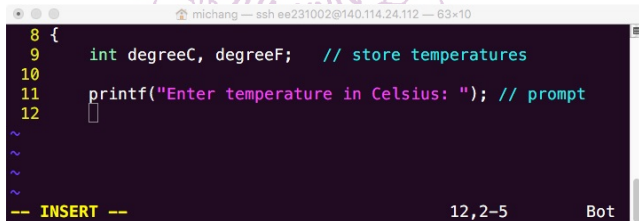
15,10-13 Bot

Auto-indent

- In insert mode, after typing a line of text the cursor moves to the first column – not aligned with the indented text
- This can be changed by `:set ai`, auto-indent, command
- `:set noai` sets no auto-indent

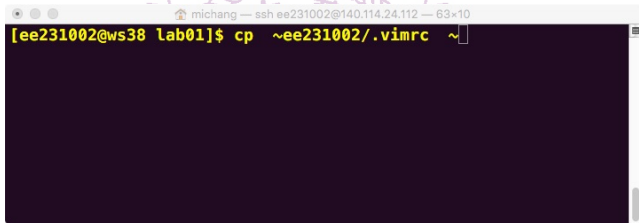


```
8 {
9     int degreeC, degreeF;    // store temperatures
10
11     printf("Enter temperature in Celsius: "); // prompt
12 |
~
~
~
-- INSERT --                               12,1      Bot
```



```
8 {
9     int degreeC, degreeF;    // store temperatures
10
11     printf("Enter temperature in Celsius: "); // prompt
12 |
~
~
~
-- INSERT --                               12,2-5    Bot
```

- `vim` program executes the commands in `.vimrc` every time it is invoked.
- Please copy `~ee2310/.vimrc` to your home directory



```
michang — ssh ee231002@140.114.24.112 — 63x10  
[ee231002@ws38 lab01]$ cp ~ee231002/.vimrc ~
```

- This file sets
 - Auto-indent mode
 - Each `Tab` inserts 4 spaces

vim Tutorial

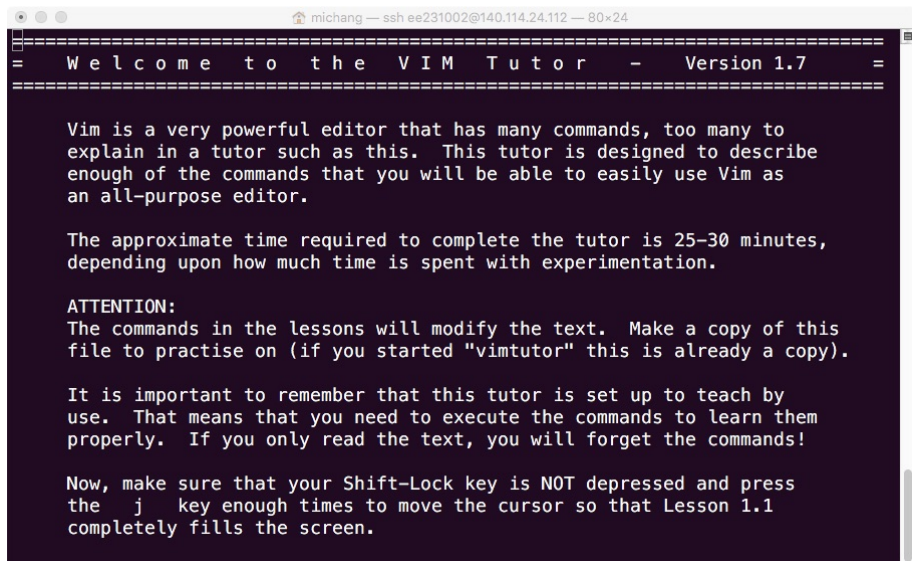
- `vim` program provides a tutorial for users to learn the easy commands
- At a linux terminal type in `vimtutor` as following to enter the tutorial



The screenshot shows a terminal window with a dark background. The title bar at the top reads "michang — ssh ee231002@140.114.24.112 — 80x20". The prompt is "[ee231002@ws38 lab01]\$". The command "vimtutor" has been typed and is highlighted in yellow. A cursor is visible at the end of the command.

vim Tutorial, II

- Most frequently used commands are demonstrated



```
michang — ssh ee231002@140.114.24.112 — 80x24
=====
=  Welcome to the VIM Tutor - Version 1.7  =
=====

Vim is a very powerful editor that has many commands, too many to
explain in a tutor such as this.  This tutor is designed to describe
enough of the commands that you will be able to easily use Vim as
an all-purpose editor.

The approximate time required to complete the tutor is 25-30 minutes,
depending upon how much time is spent with experimentation.

ATTENTION:
The commands in the lessons will modify the text.  Make a copy of this
file to practise on (if you started "vimtutor" this is already a copy).

It is important to remember that this tutor is set up to teach by
use.  That means that you need to execute the commands to learn them
properly.  If you only read the text, you will forget the commands!

Now, make sure that your Shift-Lock key is NOT depressed and press
the  j  key enough times to move the cursor so that Lesson 1.1
completely fills the screen.
```