Introduction to Vim, I

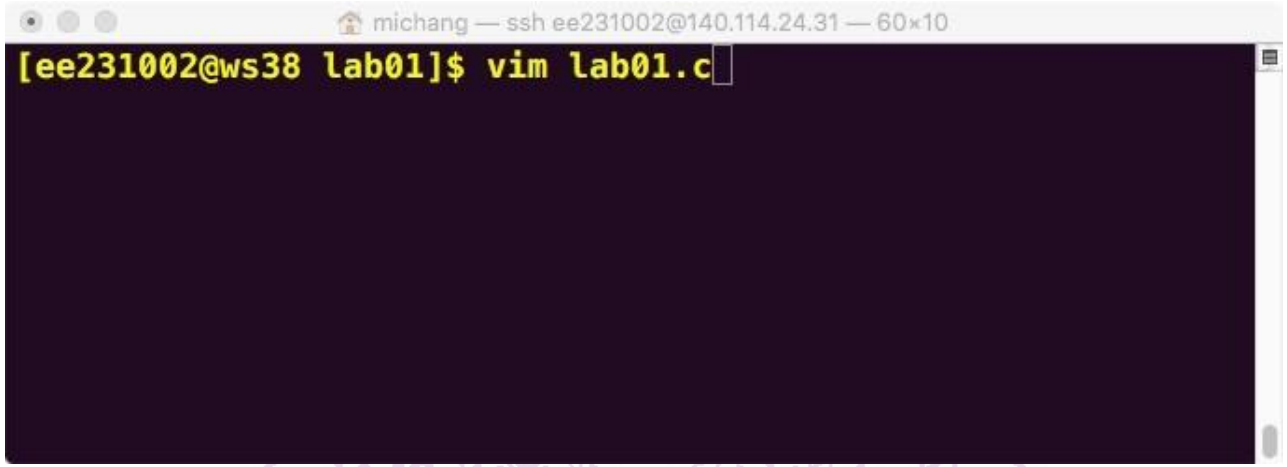
EE231002 Introduction to Programming

TA 胡恩典

Sep. 17, 2018

Starting vim

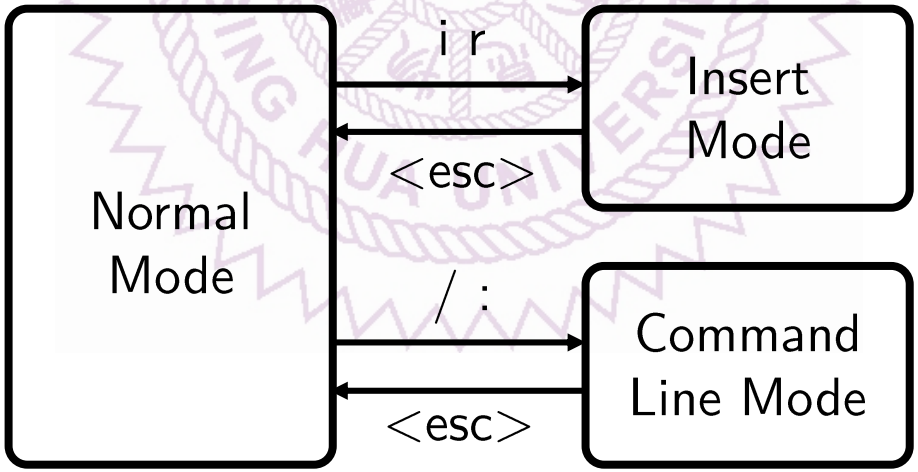
- To start `vim`: `vim file`



- For a new file

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
  - Normal mode and Command line mode will be covered next time



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

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 — 63×10

```
{
    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 — 63×10

```
 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 the color text
- It also works in other kind of files like .cpp, .py, .js ...

michang — ssh ee231002@140.114.24.112 — 63×10

```
{
    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;
}
```

michang — ssh ee231002@140.114.24.112 — 63×9

```
    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 we use to view your program
  - Be sure your program is very legible to us 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 — 63×10

```
 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 — 63×10

```
 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

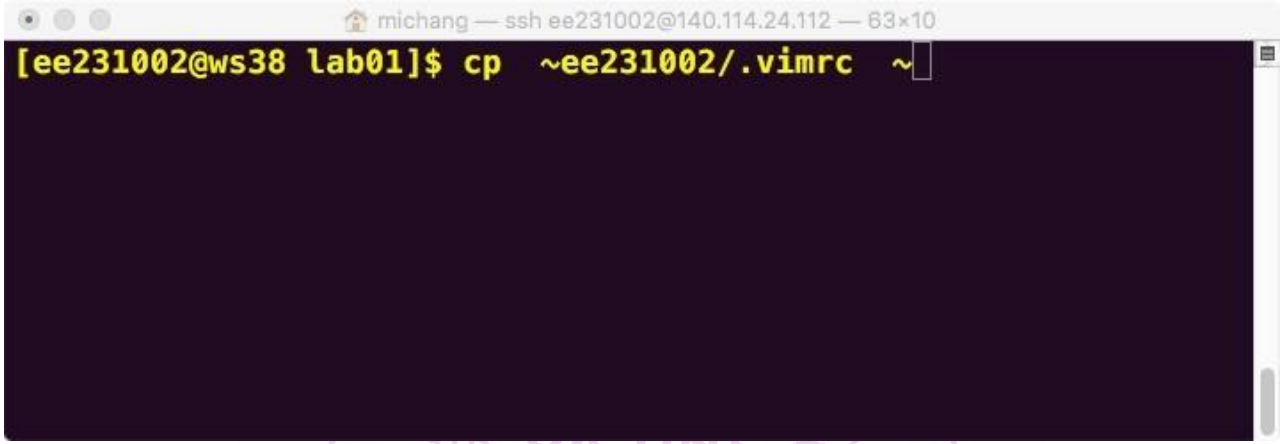
- In insert mode, after typing a line of text, the cursor moves to the first column – not aligned with the indented text
- This can be changed by `:set ai`, auto-indent, command
- `:set noai` sets no auto-indent

```
michang — ssh ee231002@140.114.24.112 — 63×10
 8 {
 9     int degreeC, degreeF;   // store temperatures
10
11     printf("Enter temperature in Celsius: "); // prompt
12
~
~
~
~
-- INSERT --                                  12,1          Bot
```

```
michang — ssh ee231002@140.114.24.112 — 63×10
 8 {
 9     int degreeC, degreeF;   // store temperatures
10
11     printf("Enter temperature in Celsius: "); // prompt
12
~
~
~
~
-- INSERT --                                  12,2-5        Bot
```

.vimrc

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



- This file sets
  - Auto-indent mode
  - Each `Tab` inserts 4 spaces
  - Remember that your points will be deducted if you don't indent appropriately

Set your preference in .vimrc

- Following example shows how to set your preference
- Before

```
set tabstop=4
set sw=4
```

.vimrc

```
#include <stdio.h>

int main(void)
{
    return 0;
}
```

lab01.c

- Launch `.vimrc` by command `vim ~/.vimrc`
- Insert `set nu` and you will see the difference after saving by `:wq`

```
set tabstop=4
set sw=4
set nu
```

.vimrc

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5    return 0;
6 }
```

lab01.c

vim Tutorial

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

michang — ssh ee231002@140.114.24.112 — 80×20

```
[ee231002@ws38 lab01]$ vimtutor
```

vim Tutorial, II

- Most frequently used commands are demonstrated

```
michang — ssh ee231002@140.114.24.112 — 80×24
===================================================================
=   W e l c o m e   t o   t h e   V I M   T u t o r   -   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.
```

Compiler and GCC

- GCC: GNU Compiler Collection
  - Original: GNU C Compiler for processing C only
  - It can process some other programming languages like C++ nowadays
- Compiler: transforms code in one programming language to target language
  - Just consider it as a translator
- C Compiler performs following operation when compiling
  - Lexical Analysis: Mark the tokens like keywords, constants, identifiers...
  - Parsing: Go through the tokens and process the grammar
  - Code Generation: For C Compiler, it generates the assembly code
  - Linking: Link the related object file and generate an execution file
  - Load: Load the execution file into the memory
- In this course, you only need to know:
  - `gcc labxx.c` generates an execution file `a.out`
  - `./a.out` executes your program
- For more information about Compiling and Compiler, you can take: CS1356 I2P(II), EE3450/CS4100 Computer Archi. CS3404 Compiler Design

Example Program in Required Format

Format.c

```c
/* EE231002 Lab00 Format
 * 104061220 En-Tien, Hu
 * Date: Sep. 17, 2018
 */

#include <stdio.h>

int main(void)
{
    int i;                                  //for iteration
    int id;                                 //to store the input

    printf("Insert your student ID: ");     //print the statement
    scanf("%d", &id);                       //read the input number

    for (i = 0; i < 10; i++) {              //10-time iteration
        if (i % 2 == 0)                     //if i is multiple of 2
            printf("%d\n", id);             //print the id
        else                                //if not
            printf("%d\n", id + 100);       //print id+100
    }

    return 0;                               //program terminates
}
```

Appendix

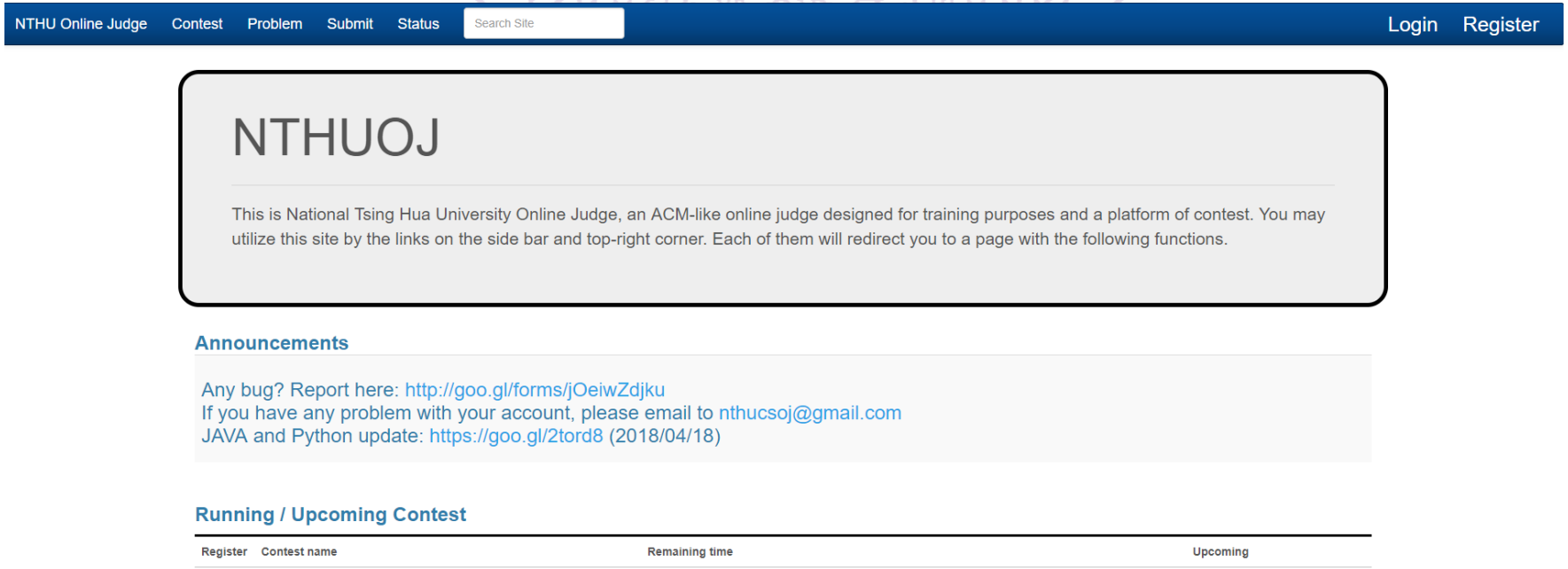
EE231002 Introduction to Programming

TA 胡恩典 (logical203010@gmail.com)

(Optional) Send an email to me for following information

For those who want more practice

- Google "NTHU Online Judge"
- This is used in CS Department, but everyone can register an account
- Click "Contest", you can try some of which beginning with I2P or I2P(I)
- Actually, you don't need any extra practice in our course
  - Therefore, don't spend too much time on this
  - All you need to do is getting familiar with problems in our lab



My Email Policy

- Again, this is my email address
  - logical203010@gmail.com
- You can expect that your mail will be replied in 2 hours
  - Only if you send the mail between 10:00 AM and 01:00 AM
  - You may send another mail if I don't reply in 2 hours
  - If I can't save your problem in 2 hours, I will still reply and ask you to wait
- Writing mails in English is encouraged
  - But mails in Mandarin or Korean are also allowed
  - I will reply the mail in the language you use

Links to SNS Accounts and Policy

- Following SNS Accounts are provided for convenience
  - We don't encourage you to ask question through SNS
  - Practicing formal letters is important

Messenger

Kakao Talk

WeChat

LINE

About Foreign Language Test

- I have taken several Foreign Language Tests these months
  - TOEIC on Dec. 19, 2017
  - GRE General Test on Aug. 18, 2018
  - TOEFL iBT on Sep. 15, 2018
  - TOPIK I, II on Oct. 21, 2018 (Preparing)
- Feel free to come to me if you need some advices

Courses I have taken in EECS

- Feel free to come to me if you need information about following courses
  - All required courses in EE
  - CS1356 I2P(II)
  - CS3404 Compiler Design
  - CS4100 Computer Architecture
  - EE2230 Logic Design Laboratory
  - EE2405 Embedded System Laboratory
  - EE3980 Algorithms
  - EE6455 Adv. Computer Architecture
  - EECS2040 Data Structure
  - CS3423 Operating Systems
  - EECS4010 Digital System Design
  - EE2401 Microprocessor Systems
  - EECS3020 Intro. to Computer Network
  - EE4292 IC Design Laboratory

Encouragement from TA

三年前的現在我對程式設計一無所知
所以在過去的這三年我告訴自己要比別人更努力
既然今天我能夠爬上助教的位置
表示大家都有能力把這堂課學好
當你快撐不下去的時候請記得主動尋求幫助　　—TA