รหัสนิสิต..... ชื่อ สกุล ..... กลุ่ม .....

ปฏิบัติการที่ 1 แนะนำเครื่องมือในปฏิบัติการ

บทเรียนย่อย

- GNU C++ Compiler : g++
- Utility : make and Makefile
- การใช้ Debug ด้วย GBD

วัตถุประสงค์การเรียนรู้

- อธิบายถึงเครื่องมือที่ใช้ในการปฏิบัติการ
- อธิบายถึงการคอมไพล์ในภาษา C++ และการเรียกใช้งานโปรแกรม
- เรียนรู้การใช้งาน Make
- เรียนรู้เครื่องมือที่ใช้ตรวจสอบข้อผิดพลาดของโปรแกรม

ตอนที่ 1 ทำความรู้จักกับภาษา C++

1. ทำการสร้าง source code ของภาษาโปรแกรม C++ โดยทำการ login ไปยังเครื่อง dekdee.buu.ac.th แล้วสร้างไฟล์ hello.cpp แล้วพิมพ์ source code ดังต่อไปนี้

```
#include <iostream>
using namespace std;
int main()
{
  cout << "Hello World" << endl;
  cout << "My This First Program" << endl;
  return 0;
}
```

2. ตัวแปลภาษา (Compiler) จะทำหน้าที่ตรวจสอบ Source Code ที่เขียนว่าถูกต้องตามโครงสร้างของภาษาหรือไม่ แล้วทำการแปลงจาก Source code เป็นภาษาคอมพิวเตอร์
3. วิธีการ Compile โปรแกรม โดยการพิมพ์คำสั่ง

   # g++ hello.cpp
4. ตรวจสอบผลลัพธ์ที่เกิดขึ้น

   # ls
5. วิธีการ Execute โปรแกรม หรือเรียกใช้งานโปรแกรม

   # ./a.out

Note:
หลังเครื่องหมาย # เป็นคำสั่งที่สั่งให้เครื่องคอมพิวเตอร์ประมวลผลคำสั่งที่ผู้ใช้งานพิมพ์ไป

ตอนที่ 2 GNU C++ Compiler

Option -c, -S, or -E to say where gcc is to stop. Note that some combinations(for example, -x cpp-output -E) instruct gcc to do nothing at all.

-E Stop after the preprocessing stage; do not run the compiler proper.The output is in the form of preprocessed source code, which is sent to the standard output.Input files which don't require preprocessing are ignored.

-S Stop after the stage of compilation proper; do not assemble. Theoutput is in the form of an assembler code file for each non-assembler input file specified.By default, the assembler file name for a source file is made byreplacing the suffix .c, .i, etc., with .s.Input files that don't require compilation are ignored.

- c Compile or assemble the source files, but do not link. The linkingstage simply is not done. The ultimate output is in the form of anobject file for each source file.By default, the object file name for a source file is made byreplacing the suffix .c, .i, .s, etc., with .o.Unrecognized input files, not requiring compilation or assembly,are ignored.

1. รู้จักออฟชันของการคอมไพล์ ในภาษาซีพลัสพลัส
   ให้นิสิตสรุปว่าในแต่ละออพชันของการคอมไพล์มีลักษณะการทำงานเป็นอย่างไร

   .....
   .....
   .....
   .....
   .....
   .....
   .....
   .....
   .....
   .....
   .....

2. ทำการสร้างไฟล์ lab01.cpp

```
#include <iostream>
using namespace std;
int main()
{
    int num;
    num = 6;
    cout << "My first C++ program." << endl;
    cout << "The sum of 2 and 3 = " << 5 << endl;
    cout << "7 + 8 = " << 7 + 8 << endl;
    cout << "Num = " << num << endl;
    return 0;
}
```

3. ทดลองคอมไพล์ด้วยออฟชันต่างๆ

ในการ Compile & Link โปรแกรมภาษา C++ นั้นทำได้โดยการใช้คำสั่ง g++
ให้นิสิตทำการทดลองต่อไปนี้เพื่อตรวจและรายงานผลลัพธ์ที่เกิดขึ้น

4. ทดลองใช้คำสั่ง gcc lab01.cpp

ผลลัพธ์ที่ได้คือ

.....

.....

.....

5. ทดลองใช้คำสั่ง g++ lab01.cpp

ผลลัพธ์ที่ได้คือ

.....

.....

.....

6. ทดลองใช้คำสั่ง g++ -E lab01.cpp

ผลลัพธ์ที่ได้คือ

.....

.....

.....

.....

7. ทดลองใช้คำสั่ง g++ -S lab01.cpp

ผลลัพธ์ที่ได้คือ

.....

.....

.....

.....

8. ทดลองใช้คำสั่ง g++ -c lab01.cpp ตามด้วยคำสั่ง g++ lab01.o

ผลลัพธ์ที่ได้คือ

.....

.....

.....

9. ทดลองใช้คำสั่ง g++ lab01.cpp -o lab01

ผลลัพธ์ที่ได้คือ

.....

.....

.....

ตอนที่ 3 Utility: make and Makefile

การพัฒนาซอฟต์แวร์สามารถใช้ make utility ในการทำ automatic build โปรแกรมได้ ให้นิสิตทดลองสร้าง Makefile โดยพิมพ์คำสั่ง vi ดังนี้

#vi Makefile

```
# Sample Makefile
lab01 : lab01.o
# Use <tab> not <space_bar> for indent
      @echo "Link and create an execute file"
      g++ lab01.o -o lab01
lab01.o : lab01.cpp
      @echo "Compile the program"
      g++ -c lab01.cpp
clean :
      @echo "Clean non source code file"
      rm -rf *.o lab01
```

ใช้คำสั่ง make และตรวจสอบผลที่ได้

make

ผลลัพธ์ที่ได้คือ

.....

.....

make clean

ผลลัพธ์ที่ได้คือ

.....

.....

ทดลองคอมไพล์ และรันโปรแกรม ด้วยคำสั่ง ./lab01 และตรวจสอบผลที่ได้

make

./lab01

ผลลัพธ์ที่ได้คือ

.....

.....

รันคำสั่ง make clean และตรวจสอบผลที่ได้

make clean

ผลลัพธ์ที่ได้คือ

.....

.....

ให้นิสิตสำเนา Makefile เดิมไว้ที่ Makefile.a และให้ปรับแก้แฟ้ม Makefile ใหม่

cp Makefile Makefile.a

vi Makefile

รหัสนิสิต..... ชื่อ สกุล ..... กลุ่ม .....

โดยให้ปรับแก้ข้อมูลซอร์สโค้ดเดิม

```
lab01 : lab01.cpp
     @echo "Compile & Link and create an execute file"
     g++ lab01.cpp -o lab01
clean :
     @echo "Clean non source code file"
     rm -rf *.o lab01
```

เมื่อบันทึกเรียบร้อย สั่ง make และตรวจสอบผลที่ได้

ผลลัพธ์ที่ได้คือ

.....

.....

ทดลองสร้างไฟล์ lab02.cpp

```
using namespace std;
#include <iostream>    // This is a key C++ library
#include <cmath>       // The standard C library math.h

int main ()
{
   double a;
   a = 1.2;
   a = sin (a);
   cout << a << endl;
   return 0;
}
```

ให้นิสิตสำเนา Makefile เดิมไว้ที่ Makefile.b และให้ปรับแก้แฟ้ม Makefile ใหม่อีกให้มีข้อมูลดังนี้

```
# cp Makefile Makefile.b
# vi Makefile
```

โดยให้ปรับแก้ข้อมูลซอร์สโค้ดเดิม

```
all : lab01 lab02
lab01 : lab01.cpp
     g++ lab01.cpp -o lab01
lab02 : lab02.cpp
     g++ lab02.cpp -o lab02
clean :
     @echo "Clean non source code file"
     rm -rf *.o lab01 lab02
```

เมื่อบันทึกเรียบร้อย สั่ง make และตรวจสอบผลที่ได้

ผลลัพธ์ที่ได้คือ

.....

.....

ตอนที่ 4 การ Debug ด้วย GDB

การ Debug เป็นการตรวจสอบการทำงานของโปรแกรมอย่างละเอียด
เพื่อที่จะค้นหาและแก้ไขข้อผิดพลาดของโปรแกรม เครื่องมิที่ใช้ในการ Debug คือ gdb

รูปแบบการใช้: gdb file_name

gdb มีคำสั่งที่น่าสนใจดังนี้

run	สั่งให้โปรแกรมเริ่มทำงาน
ctrl+C	ยกเลิกการทำงาน
where	แสดงบรรทัดที่กำลัง run
step	run ทีละบรรทัด โดย run ใน function ย่อยด้วย (step into)
next	run ทีละบรรทัด (step over)
list <function name>	แสดง code ใน function
print <variable name>	แสดงค่าของตัวแปรนั้น
help	ดูว่ามีคำสั่งอะไรบ้าง
help <command>	ดูว่ามีรายละเอียดของคำสั่งนั้น
break <function>	หยุดการทำงานที่ function นั้น
break <line>	หยุดการทำงานที่บรรทัดนั้น
watch <variable>	หยุดการทำงานเมื่อค่าตัวแปรนั้นเปลี่ยนแปลง
info break	แสดง ลำดับของ break point ทั้งหมด
delete <seq number>	ลบ break point ที่หมายเลขนั้นออก
cont	ทำงานต่อ หลังจากที่หยุดจาก break point
quit	ออกจากโปรแกรม gdb

Note:
การใช้ Help เพื่อช่วยในการเขียนโปรแกรม ทำได้โดยการใช้สั่ง man เช่น หากต้องการทราบรายละเอียดของการcompile ก็พิมพ์ว่า man g++

การ Debug โดยใช้ GDB

สร้างไฟล์ C++ ชื่อ num.cpp

```
#include<iostream>
using namespace std;
int main()
{
  int a,b,c;
  a = 2;
  b = a + 1;
  c = b + b;
  cout << "value of c =  "<< c << "\n";
  return 0;
}
```

ทดลอง Debug โปรแกรมนี้โดยใช้ gdb แต่ก่อนที่จะ Debug นั้นต้องมีการคอมไพล์โปรแกรมเสียก่อนและต้องมีการเพิ่มอ็อปชั่น -g ลงไปในคำสั่ง g++ ด้วย มิฉะนั้นจะไม่สามารถ Debug ได้

g++ -Wall -g num.cpp -o num

จากนั้นจึงเริ่ม Debug โปรแกรมโดยพิมพ์ว่า gdb num โดยเราสามารถกำหนดได้ว่าให้โปรแกรมทำงานไปถึงบรรทัดไหนโดยใช้คำสั่ง break ซึ่งเรียกว่าการกำหนด break point และเมื่อสั่ง run โปรแกรมก็จะหยุดตรงบรรทัดที่เรากำหนด

```
[peerasak@dekdee ~]$ gdb num
GNU gdb (GDB) Red Hat Enterprise Linux (7.2-64.el6_5.2)
Copyright (C) 2010 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
<http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/staff/peerasak/num...done.
(gdb) break 3
Breakpoint 1 at 0x40078c: file num.cpp, line 3.
(gdb) run
Starting program: /home/staff/peerasak/num

Breakpoint 1, main () at num.cpp:6
6         a = 2;
Missing separate debuginfos, use: debuginfo-install glibc-2.12-
1.132.el6_5.2.x86_64 libgcc-4.4.7-4.el6.x86_64 libstdc++-4.4.7-4.el6.x86_64
```

เมื่อโปรแกรมหยุด เราก็สามารถรันโปรแกรมทีละบรรทัดได้โดยใช้คำสั่ง step จากนั้นให้สังเกตผลที่เกิดขึ้นในแต่ละบรรทัด

```
Breakpoint 1, main () at num.cpp:6
6         a = 2;
Missing separate debuginfos, use: debuginfo-install glibc-2.12-
1.132.el6_5.2.x86_64 libgcc-4.4.7-4.el6.x86_64 libstdc++-4.4.7-4.el6.x86_64
(gdb) step
7         b = a + 1;
(gdb) step
8         c = b + b;
(gdb) step
9         cout << "value of c =  "<<c << "\n";
(gdb) step
value of c =  6
10        return 0;
(gdb) step
11      }
(gdb) step
0x0000003243c1ed1d in __libc_start_main () from /lib64/libc.so.6
(gdb) step
Single stepping until exit from function __libc_start_main,
which has no line number information.

Program exited normally.
(gdb)
```

แสดง source code ด้วยคำสั่ง list

```
(gdb) list
1       #include<iostream>
2       using namespace std;
3       int main()
4       {
5         int a,b,c;
6         a = 2;
7         b = a + 1;
8         c = b + b;
9         cout << "value of c =  "<< c << "\n";
10        return 0;
```

สามารถตรวจสอบค่าตัวแปรได้โดยใช้คำสั่ง print ซึ่งจะเห็นว่าเมื่อรันโปรแกรมผ่านบรรทัดที่ 6 ไปแล้ว ค่าของตัวแปร b จะมีค่าเท่ากับ 3

```
(gdb) break 3
Breakpoint 1 at 0x40078c: file num.cpp, line 3.
(gdb) run
Starting program: /home/staff/peerasak/num

Breakpoint 1, main () at num.cpp:6
6         a = 2;
Missing separate debuginfos, use: debuginfo-install glibc-2.12-
1.132.el6_5.2.x86
_64 libgcc-4.4.7-4.el6.x86_64 libstdc++-4.4.7-4.el6.x86_64
(gdb) step
7         b = a + 1;
(gdb) step
8         c = b + b;
```

```
(gdb) print b
$1 = 3
(gdb) step
9         cout << "value of c =  "<<c << "\n";
(gdb)
```

คำถาม

โปรแกรมต่อไปนี้มีข้อผิดพลาดทางไวยากรณ์ (Syntax) ของภาษา c++หรือไม่ ถ้ามีแก้ไขให้ถูกต้อง

```
#include<iostream>
using namespace std;
main
(
       a=5
       cout<< Hello World
)
```

Source Code ที่ถูกต้องคือ

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

งานหลังการทดลอง

ให้นิสิตทดลองเขียนโปรแกรมเป็นรูปต่างๆ หรือ ข้อความต่างๆ ตามใจชอบมาคนละ 1 แบบ (โดยใช้คำสั่ง cout) ตัวอย่างเช่น

```
****      ****     **********     ****         ****              *****
****      ****     ****           ****         ****          ****    ****
**********        **********     ****         ****          ***       ***
****      ****     ****           ****         ****          ****    ****
****      ****     **********     **********   **********       *****
```

รหัสนิสิต..... ชื่อ สกุล ..... กลุ่ม .....

ให้นิสิตคัดลอก Souce code ที่นิสิตเขียน