นาย ภูมิภัทร เชื้อลือ 62010722
นาย ธีรวิชญ์ รุ่งรัตนวิทย์ 62010948

ภาคผนวก E

การทดลองที่ 5 การพัฒนาโปรแกรมด้วยภาษา C

การทดลองนี้คาดว่าผู้อ่านผ่านหัวข้อที่ 3.3 และมีประสบการณ์การเขียนหรือพัฒนาโปรแกรมด้วยภาษา C มาแล้ว ผู้อ่านควรมีความคุ้นเคยกับ IDE (Integrated Development Environment) จากการพัฒนาโปรแกรมและการดีบักโปรแกรมด้วยภาษา C/C++ ดังนั้น การทดลองมีวัตถุประสงค์เหล่านี้

- เพื่อให้เข้าใจการพัฒนาซอฟท์แวร์ด้วย IDE ชื่อ CodeBlocks บนระบบปฏิบัติการ Raspbian/Linux/Unix
- เพื่อให้สามารถสร้าง Makefile เพื่อพัฒนาศักยภาพการทำงานเป็นนักพัฒนาอาชีพ
- เพื่อให้เข้าใจความแตกต่างระหว่างการพัฒนาโปรแกรมภาษา C ด้วย IDE และ Makefile

E.1 การพัฒนาโดยใช้ IDE

โปรแกรมหรือแอพพลิเคชัน IDE ย่อมาจาก Integrated Development Environment ทำหน้าที่ช่วยเหลือโปรแกรมเมอร์ ทดสอบ และอาจรวมถึงควบคุมซอร์สโค้ดให้เป็นปัจจุบัน ขั้นตอนการทดลองนี้เริ่มต้นโดย

1. ตรวจสอบภายในเครื่องว่ามีโปรแกรมชื่อ CodeBlocks ติดตั้งแล้วหรือไม่ โดยพิมพ์คำสั่งเหล่านี้ลงบนโปรแกรม Terminal

   ```
   $ codeblocks
   ```

2. หากติดตั้งแล้ว ให้ผู้อ่านข้ามไปข้อที่ 4 ได้ หากไม่มีโปรแกรม ผู้อ่านจะต้องติดตั้ง CodeBlocks ผู้อ่านต้องพิมพ์คำสั่งเหล่านี้ลงบนโปรแกรม Terminal

   ```
   $ sudo apt-get install codeblocks
   ```

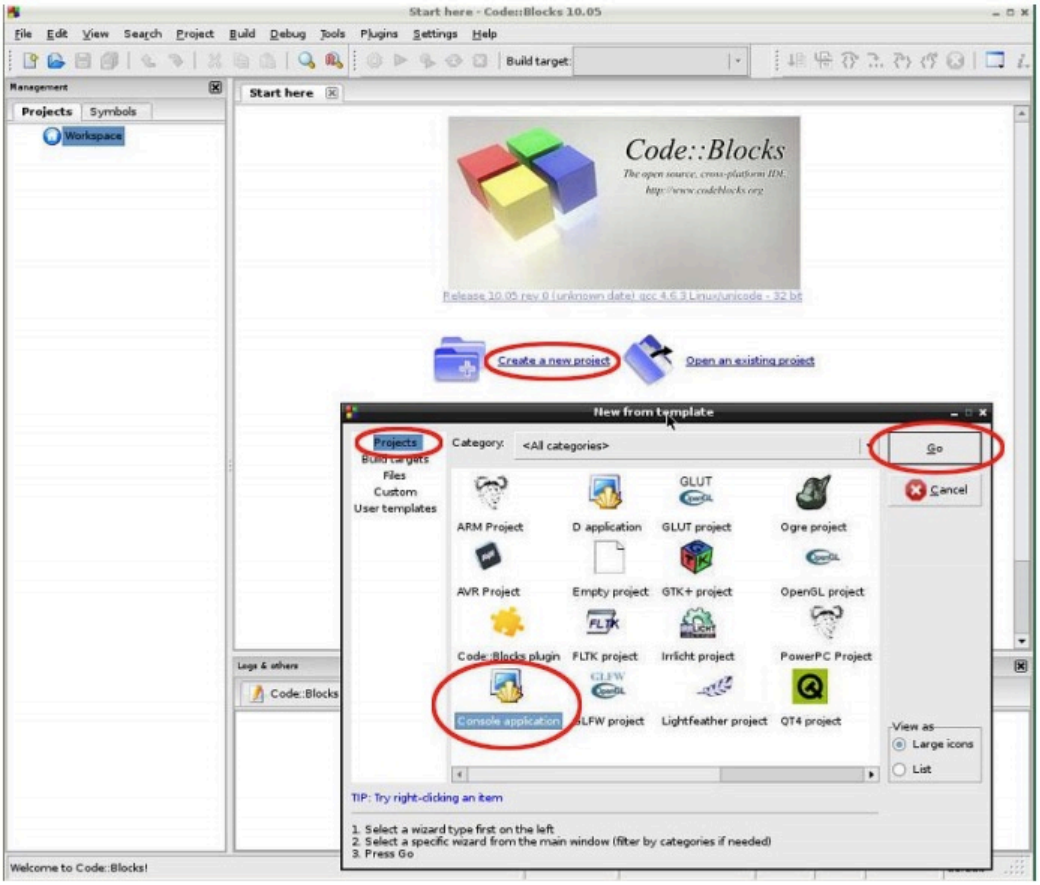
   คำสั่ง sudo นำหน้าคำสั่งใดๆ นี้จะเป็นการเรียกใช้งานคำสั่งนั้นด้วยสิทธิ์ระดับ superuser การติดตั้งจะดาวน์โหลดโปรแกรมผ่านทางเครือข่ายอินเทอร์เน็ต และจำเป็นต้องใช้สิทธิ์ระดับสูงสุดนี้

3. เมื่อติดตั้งเสร็จสิ้น พิมพ์คำสั่งนี้เพื่อเริ่มต้นใช้งาน CodeBlocks

   ```
   $ codeblocks
   ```

4. การใช้งาน CodeBlocks ครั้งแรกจะเป็นการติดตั้งค่า compiler plug-ins เป็น GCC หรือ GNU C Compiler.

5. หน้าต่างหลักจะปรากฏขึ้น หลังจากนั้น ผู้อ่านควรกด ”Create a new project” เพื่อสร้างโปรเจ็คท์ใหม่ในหน้าต่าง ”New from template”



รูปที่ E.1: หน้าต่างเลือกชนิดโปรเจ็คท์ที่จะพัฒนาเป็นชนิด ”Console application”

6. เลือก ”New Projects” ในช่องด้านซ้าย แล้วเลือก ”Console application” ในช่องด้านขวาเพื่อสร้างโปรแกรมในรูปแบบเท็กซ์โหมด (Text Mode) กดปุ่ม ”Go” ตามรูปที่ E.1

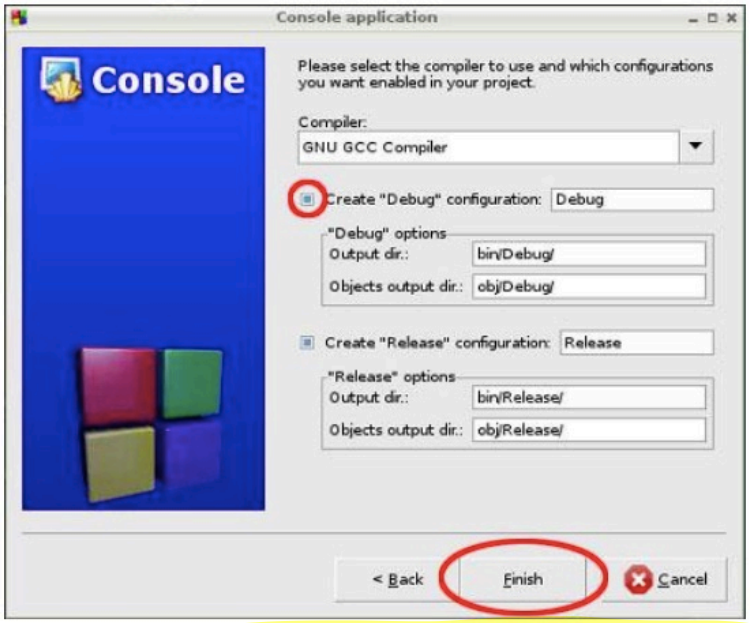
7. กดปุ่ม Next> เพื่อดำเนินการต่อ

8. หน้าต่าง ”Console application” จะปรากฏขึ้น กดเลือกภาษา ”C” เพื่อพัฒนาโปรแกรมแล้วกดปุ่ม ”Next>” ตามรูปที่ E.2)

รูปที่ E.2: หน้าต่างเลือกภาษา C หรือ C++ สำหรับโปรเจ็คท์ที่จะพัฒนา

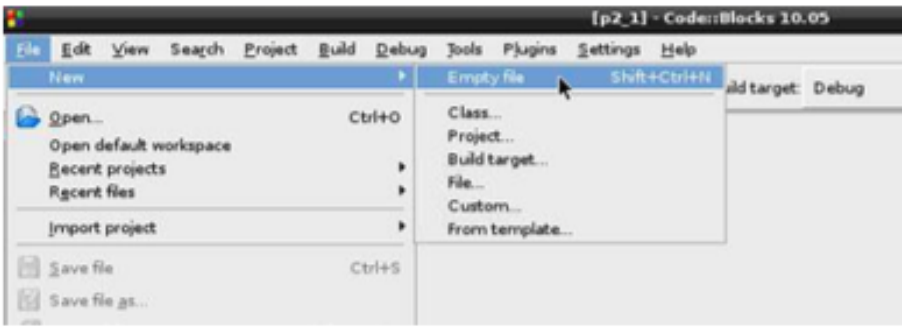
9. กรอกชื่อโปรเจ็คท์ใหม่ชื่อ Lab5 ในช่อง Project title: และกรอกชื่อไดเรคทอรี /home/pi/c/ ในช่อง Folder to create project in: โปรดสังเกตข้อความในช่อง Project filename: ว่าตรงกับ Lab5.cbp ใช่หรือไม่

10. กดปุ่ม ”Next>” เพื่อดำเนินการต่อและสุดท้ายจะเป็นขั้นตอนการเลือกคอนฟิกกูเรชัน (Configuration) สำหรับคอมไพเลอร์ในรูปที่ E.3 โดย Debug เหมาะสำหรับการเริ่มต้นและแก้ไขข้อผิดพลาด แล้วจึงกดปุ่ม ”Finish” เมื่อเสร็จสิ้น



รูปที่ E.3: การเลือกคอนฟิกกูเรชัน (Configuration) Debug สำหรับคอมไพเลอร์ GNU GCC ในโปรเจ็คท์ Lab5

11. เพิ่มไฟล์เปล่าด้วยเมนูต่อไปนี้ File->New->Empty file ตามรูปที่ E.4



รูปที่ E.4: การเพิ่มไฟล์เปล่าให้กับโปรเจ็คท์ Lab5 ที่สร้างขึ้น

12. ป้อนโปรแกรมนี้ลงในหน้าต่าง main.c

```
#include <stdio.h>
int main(void)
{
  int a;
  printf("Please input an integer: ");
  scanf("%d", &a);
  printf("You entered the number: %d\n", a);
  return 0;
}
```

13. คอมไพล์และ Build โปรแกรม จนไม่มีข้อผิดพลาด โดยสังเกตจากหน้าต่างด้านล่างสุด

14. รันโปรแกรมเพื่อทดสอบการทำงาน

E.2 การดีบัก (Debugging) โดยใช้ IDE

การดีบักโปรแกรม คือ การตรวจสอบการทำงานของโปรแกรมอย่างละเอียด CodeBlocks รองรับการดีบักผ่านเมนู Debug ผู้อ่านสามารถเริ่มต้นโดย

1. กด Debug บนเมนูแถวบนสุด เลือก Active Debuggers ซึ่งค่าปัจจุบัน (Target's Default) คือ GDB/CDB Debugger

2. ตั้งเบรกพอยท์ (Break Point) ตรงประโยคที่ต้องการศึกษา โดยเลื่อนเคอร์เซอร์ (Cursor) ไปยังบรรทัดนั้น แล้วกดปุ่ม F5 โปรดสังเกตต้นประโยคจะมีวงกลมสีแดงปรากฏขึ้น และเมื่อกด F5 อีกครั้งวงกลมสีแดงจะหายไป เรียกว่า การท็อกเกิล (Toggle) เบรคพอยท์ กด F5 อีกครั้งเพื่อสร้างวงกลมสีแดงตรงบรรทัดที่สนใจเพียงจุดเดียวเท่านั้น

3. กด F8 บนคีย์บอร์ดเพื่อรันโปรแกรมอีกรอบ โปรแกรมจะรันไปจนหยุดตรงประโยคที่มีวงกลมสีแดงนั้น โปรดสังเกตสัญลักษณ์สามเหลี่ยมสีเหลืองซ้อนทับกันอยู่ หลังจากนั้น กด F8 เพื่อดำเนินการต่อ

4. เลื่อนเคอร์เซอร์ไปยังประโยคที่มีวงกลมสีแดง กดปุ่ม F5 บนคีย์บอร์ดเพื่อปลดวงกลมสีแดงออก หรือยกเลิกเบรคพอยท์

5. เริ่มต้นใหม่เพื่อศึกษาการทำงานของ F4 (Run to cursor) โดยเลื่อนเคอร์เซอร์ไปวางบนประโยค ที่สนใจ กดปุ่ม F4 และสังเกตว่าสามเหลี่ยมสีเหลืองจะปรากฏหน้าประโยค เพื่อระบุว่าเครื่องรันมาถึงประโยคนี้แล้ว

6. กด F8 เพื่อรันต่อไป จนสิ้นสุดการทำงานของโปรแกรม

E.3 การพัฒนาโดยใช้ประโยคคำสั่งทีละขั้นตอน

ผู้อ่านควรเข้าใจคำสั่งพื้นฐานในการแปลโปรแกรมภาษาแอสเซมบลีที่สร้างขึ้นใน CodeBlocks ก่อนหน้านี้ ตามขั้นตอนต่อไปนี้

1. เปิดโปรแกรม Terminal หน้าต่างใหม่ แล้วย้ายไดเรคทอรีไปยัง **/home/pi/c/Lab5** โดยใช้คำสั่ง **cd**

2. ทำการคอมไพล์ (Compile) ไฟล์ซอร์สโค้ดให้เป็นไฟล์อ็อบเจ็คท์ (.o) โดยเรียกใช้คอมไพเลอร์ชื่อ gcc ดังนี้

```
$ gcc  -c main.c
```

ไฟล์ผลลัพธ์ ชื่อ main.o จะปรากฏขึ้น ผู้อ่านต้องตรวจสอบโดยใช้คำสั่ง ls -la เพื่อตรวจสอบวันที่และขนาดของไฟล์ เปรียบเทียบการใช้งานกับรูปที่ 3.17

3. ทำการลิงค์ (Link) โดยใช้ gcc ทำหน้าที่เป็นลิงค์เกอร์ (Linker) และแปลงไฟล์อ็อบเจ็คท์เป็นไฟล์โปรแกรม (Executable file) โดย

```
$ gcc main.o -o Lab5
```

ไฟล์ผลลัพธ์ ชื่อ Lab5 จะปรากฏขึ้น ผู้อ่านต้องตรวจสอบโดยใช้คำสั่ง ls -la เพื่อตรวจสอบวันที่และขนาดของไฟล์เพื่อเปรียบเทียบกับ man.o

4. รัน (Run) โปรแกรม Lab5 โดยพิมพ์

```
$ ./Lab5
```

5. เปรียบเทียบผลลัพธ์ที่ปรากฏขึ้นว่าตรงกับผลการรันใน IDE หรือไม่ อย่างไร

E.4 โครงสร้างของ Makefile

นอกเหนือจากการพัฒนาโปรแกรมด้วย IDE แล้ว การพัฒนาด้วย Makefile จะช่วยให้นักพัฒนามือสมัครเล่นและมืออาชีพดำเนินการได้ถูกต้องและรวดเร็ว ไฟล์ชื่อ Makefile เป็นไฟล์อักษรหรือเท็กซ์ไฟล์ (text file) ง่ายๆ ที่อธิบายความสัมพันธ์ระหว่าง ไฟล์ซอร์สโค้ดต่างๆ ไฟล์อ็อบเจ็คท์ และไฟล์โปรแกรม แต่ละบรรทัดจะมีโครงสร้างดังนี้

```
target : prerequisites ...
<tab>recipe
<tab>        ...
<tab>...
```

- target หมายถึง ชื่อไฟล์ที่จะถูกสร้างขึ้น โดยอาศัยไฟล์ต่างๆ จากส่วนที่เรียกว่า prerequisites นอกจากชื่อไฟล์แล้ว คำสั่ง 'clean' สามารถใช้เป็น target ได้ จึงนิยมใช้สำหรับลบไฟล์ต่างๆ ที่ไม่ต้องการ
- recipe หมายถึง คำสั่งหรือการกระทำที่จะใช้รายชื่อไฟล์ใน prerequisites นั้นมาสร้างไฟล์ target ได้สำเร็จ โดยแต่ละบรรทัดจะต้องเริ่มต้นด้วยปุ่ม tab เสมอ

E.5 การพัฒนาโดยใช้ Makefile

ตัวอย่างนี้เป็นการสร้าง Makefile เพื่อใช้คอมไพล์และลิงค์โปรแกรมเดิมที่เรามีอยู่ ผู้อ่านจะได้เข้าใจกลไกการทำงานที่ง่ายที่สุดก่อน หลังจากนั้นผู้อ่านสามารถศึกษาเพิ่มเติมด้วยตนเองได้จากเว็บไซต์หรือตัวอย่างโปรแกรม Open Source ที่ซับซ้อนขึ้นเรื่อยๆ ต่อไป

1. ในโปรแกรม Terminal ย้ายไดเร็คทอรีปัจจุบันไปที่ /home/pi/c/Lab5
2. เรียกใช้โปรแกรม nano ในหน้าต่าง Terminal

```
$ nano
```

กรอกข้อความเหล่านี้ในไฟล์เปล่าโดยใช้ nano

```
Lab5: main.o
        gcc main.o -o Lab5
main.o: main.c
        gcc -c main.c
clean:
        rm *.o
```

3. เมื่อกรอกเสร็จแล้ว ให้ทำการบันทึก หรือ save โดยตั้งชื่อไฟล์ว่า Makefile หรือ makefile อย่างใดอย่างหนึ่งโดยไม่มีนามสกุล หลังจากนั้น และบันทึกในไดเรคทอรี /home/pi/c/Lab5 แล้วปิดโปรแกรม nano

4. พิมพ์คำสั่งนี้ใน Terminal

```
$ make clean
```

เพื่อเรียกใช้คำสั่ง rm *.o ผ่านทาง Makefile เพื่อลบ (Remove) ไฟล์ที่มีนามสกุล .o ทั้งหมด

5. พิมพ์คำสั่งนี้ใน Terminal

```
$ make Lab5
```

เพื่อเรียกใช้คำสั่ง gcc -c main.c และ gcc -g main.c -o Lab5 เพื่อสร้างไฟล์คำสั่ง Lab5 ที่จะทำงานตามซอร์สโค้ด main.c ที่กรอกไป โดยไฟล์ Lab5 ที่เกิดขึ้นใหม่จะมีโครงสร้างรูปแบบ ELF

6. พิมพ์คำสั่งนี้ใน Terminal

```
$ ls -la
```

เพื่ออ่านค่าเวลาที่ไฟล์ Lab5 ที่เพิ่งถูกสร้าง โปรดสังเกตสีของชื่อไฟล์ต่างๆ ว่ามีสีอะไรบ้าง และบ่งบอกอะไรตามสีนั้นๆ

7. พิมพ์คำสั่งนี้ใน Terminal

```
$ ./Lab5
```

เป็นการเรียกใช้คำสั่ง Lab5 ให้ซีพียูปฏิบัติตาม

E.6 กิจกรรมท้ายการทดลอง

1. จงเปรียบเทียบโฟลว์การพัฒนาโปรแกรมในภาคผนวกนี้กับรูปที่ 3.16
2. http://www.sunshine2k.de/coding/javascript/onlineelfviewer/onlineelfviewer.html โหลดไฟล์ Lab5 ที่ได้จากการคอมไพล์และลิงค์ และเปรียบเทียบกับโครงสร้างของไฟล์ ELF ในรูปที่ 3.10
3. ใน Terminal จงย้ายไดเร็คทอรีปัจจุบันไปที่ /home/pi/c/Lab5

```
$ ls -la
```

เพื่ออ่านรหัสสีของชื่อไฟล์ต่างๆ

4. จงพัฒนาโปรแกมภาษา C โดยประกาศตัวแปรและตั้งค่าเริ่มต้น unsigned int i=1 และให้วนลูปเพิ่มค่า i=i+1 จน i มีค่าเป็นศูนย์แล้วแสดงผลค่าของ i มาทางหน้าจอ ตามตัวอย่างต่อไปนี้

```
#include <stdio.h>
int main()
{
    unsigned int i=1;
    while (i>0) {
        i=i+1;
        if (i==0) {
            printf("i was %10u before\n",i-1);
            printf("i  is %10u now\n",i);
        }
    }
    return 0;
}
```

ไฟล์ Lab5_4.c

5. จงพัฒนาโปรแกมภาษา C โดยประกาศตัวแปรและตั้งค่าเริ่มต้น int i=0 และให้วนลูปเพิ่มค่า i=i+1 จน i มีค่าเป็นลบแล้วแสดงผลค่าของ i ออกมาทางหน้าจอ โดยใช้โปรแกรมก่อนหน้าเป็นต้นแบบ

ไฟล์ Lab5_5.c

6. จงพัฒนาโปรแกมภาษา C โดยประกาศตัวแปรและตั้งค่าเริ่มต้น int i=-1 และให้วนลูปลดค่า i=i-1 จน i มีค่าเป็นบวกแล้วแสดงผลค่าของ i ออกมาทางหน้าจอ โดยใช้โปรแกรมก่อนหน้าเป็นต้นแบบ

ไฟล์ Lab5_6.c

7. จงพัฒนาโปรแกมภาษา C ให้สามารถอ่านไฟล์ Makefile เพื่อแสดงตัวอักษรในไฟล์ทีละตัวและค่ารหัส ASCII ฐานสิบหกของตัวอักษรนั้นบนหน้าจอ แล้วปิดไฟล์เมื่อเสร็จสิ้น

8. จงพัฒนาโปรแกมภาษา C เพื่อสั่งพิมพ์เลขอนุกรม Fibonacci โดยรับค่าเลขเป้าหมาย n ซึ่งเกิดจาก n = (n-1) + (n-2) และรายละเอียดเพิ่มเติมได้จาก wikipedia ตัวอย่างต่อไปนี้ n=5 และพิมพ์ผลลัพธ์ดังนี้ 1 1 2 3 5

ไฟล์ fibbonacci.c