

การทดลองที่ 3 การใช้ Assembler เบื้องต้น

1. การใช้ Editor

ใน Raspberry Pi มี Editor ที่ใช้งานหลายโปรแกรม ในที่นี้แนะนำ 2 โปรแกรม คือ Nano และ Vim

คำสั่ง ให้เรียกโปรแกรม nano หรือ vim แล้วทดลองใช้งาน ตามคำแนะนำในเอกสารแนบ

หมายเหตุ หากจะใช้ vim ต้องติดตั้งโดยใช้คำสั่ง `sudo apt install vim`

2. First Program

ให้ใช้ Editor สร้างโปรแกรมต่อไปนี้ (lab3a.s) (Indent ให้ใช้ Tab)

```
.global _start
_start:
    MOV    R0, #65
    MOV    R7, #1
    SWI     0
```

_start เป็น Label ซึ่งทุกโปรแกรมต้องมี โดยทำหน้าที่บอกว่าโปรแกรมจะเริ่มทำงานที่จุดใด (คล้ายกับ main ใน c) โดย Label ต้องปิดท้ายด้วย : เพื่อบอกว่าเป็น Label ซึ่งหมายถึงตำแหน่ง

คำสั่ง SWI (SoftWare Interrupt) เป็นคำสั่งพิเศษ สำหรับเรียกใช้บริการของระบบปฏิบัติการ PI OS โดยการใช้งานต้องระบุ รหัสฟังก์ชันไว้ที่ r7 ซึ่งฟังก์ชัน #1 คือ ให้ออกจากโปรแกรม

ให้ใช้คำสั่ง

<code>as -o lab3a.o lab3a.s</code>	เพื่อสร้าง object code
<code>ld -o lab3a lab3a.o</code>	เพื่อ Link

รันโปรแกรมโดยใช้คำสั่ง

./lab3a

ใช้คำสั่ง echo \$?

ระบบแสดงอะไร

ให้แก้ไขโปรแกรม โดยแก้ไขจาก #65 เป็นเลขอื่น แล้วรันโปรแกรมใหม่อีกครั้ง และใช้ echo \$?

ระบบแสดงอะไร

อธิบายการทำงานของโปรแกรมนี้อยู่

3. Second Program

ให้ใช้ Editor สร้างโปรแกรมต่อไปนี้ (lab3b.s)

```
/* Add two 32-bit numbers together */
```

```
/* Perform R0=R1+R2 */
```

```
.global _start
```

```
_start:
```

```
MOV    R1, #50    @ Get 50 into R1
```

```
MOV    R2, #60    @ Get 60 into R2
```

```
ADDS   R0, R1, R2 @ Add the two, result in R0
```

```
MOV    R7, #1     @ exit through syscall
```

```
SWI    0
```

ให้ทดลองรัน และดูผล

ระบบแสดงอะไร

อธิบายการทำงานของโปรแกรมนี้อยู่

4. การแสดงผลออกจอภาพ

ในการแสดงผลออกจอภาพ จะเรียกใช้ System Call ของ Linux โดยผ่านคำสั่ง SWI โดยใช้ System Call หมายเลข 4 โดยมีรูปแบบการใช้งานดังนี้

r0 = หมายเลขของอุปกรณ์ที่จะส่งออก (1 = Monitor)

r1 = ตำแหน่งของ String ที่จะแสดงผล

r2 = จำนวนตัวอักษรที่แสดงผล

ให้ใช้ Editor สร้างโปรแกรมต่อไปนี้

```
/* How to use Syscall 4 to write a string */

.global _start
_start:
    MOV    R7, #4          @ Syscall number
    MOV    R0, #1          @ Stdout is monitor
    MOV    R2, #19         @ string is 19 chars long
    LDR    R1, =string      @ string located at string:
    SWI 0

_exit:
    MOV R7, #1             @ exit syscall
    SWI 0

.data
string: .ascii "Hello World String\n"
```

สำหรับ .data หมายถึง เป็นส่วนของข้อมูล และ .ascii หมายถึงเป็นรหัส ascii

ให้ทดลองรัน และดูผล ระบบแสดงอะไร _____

อธิบายการทำงานของโปรแกรมนี้

กรณีของการกำหนดข้อมูลสามารถใช้ข้อมูลแบบอื่นๆ ดังนี้

.byte หมายถึงข้อมูล หน่วยละ 8 บิต

.hword หมายถึงข้อมูล หน่วยละ 16 บิต

.word หมายถึงข้อมูล หน่วยละ 32 บิต

.float หมายถึงข้อมูล ทศนิยม

.double หมายถึง ข้อมูลทศนิยม แบบ double precision

.space #bytes [,fill] หมายถึงจองที่ในหน่วยความจำตามจำนวนที่ระบุ (fill เป็น option)

.ascii หมายถึง string

.asciiz หมายถึง string ที่ปิดด้วย 0

5. การรับข้อมูลจากคีย์บอร์ด

การรับข้อมูลจากคีย์บอร์ดจะใช้ System Call หมายเลข 3 โดยมีรูปแบบการใช้งานดังนี้

r0 = หมายเลขของอุปกรณ์ที่จะรับเข้า (0 = Keyboard)

r1 = ตำแหน่งของ String ของข้อมูลที่ได้รับเข้ามา

r2 = จำนวนตัวอักษรที่ต้องการรับ

ให้ใช้ Editor สร้างโปรแกรมต่อไปนี้ /* How to use Syscall 3 to read from keyboard */

```
.global _start

_start:

_read:                                @ read syscall

    MOV    R7, #3                    @ Syscall number
    MOV    R0, #0                    @ Stdin is keyboard
    MOV    R2, #5                    @ read first 5 characters
    LDR    R1,=string                @ string placed at string:
    SWI 0

_write:                                @ write syscall

    MOV    R7, #4                    @ Syscall number
    MOV    R0, #1                    @ Stdout is monitor
    MOV    R2, #19                   @ string is 19 chars long
    LDR    R1, =string               @ string located at string:
    SWI 0

_exit:                                @ exit syscall

    MOV    R7, #1
    SWI 0

.data
string: .ascii "Hello World String\n"
```

ให้ทดลองรัน และดูผล ระบบแสดงอะไร

อธิบายการทำงานของโปรแกรมนี้อยู่

6. แสดงตัวอักษรของตัวเลขที่รับเข้ามา

รับตัวเลข 0 – 9 จำนวน 1 ตัว แล้วแสดงตัวอักษรของตัวเลขที่รับเข้ามาออกทางจอภาพ (Zero, One, Two, Three, Four, Five, Six, Seven, Eight หรือ Nine)

7. รับตัวอักษรทาง Keyboard จำนวน 5 ตัวอักษร แล้วแสดงผล ASCII + 5 ของตัวอักษรเหล่านั้นแบบแยกบรรทัด