

ปฏิบัติการที่ 0: อุ่นเครื่องก่อนการเขียนโปรแกรม MIPS assembly

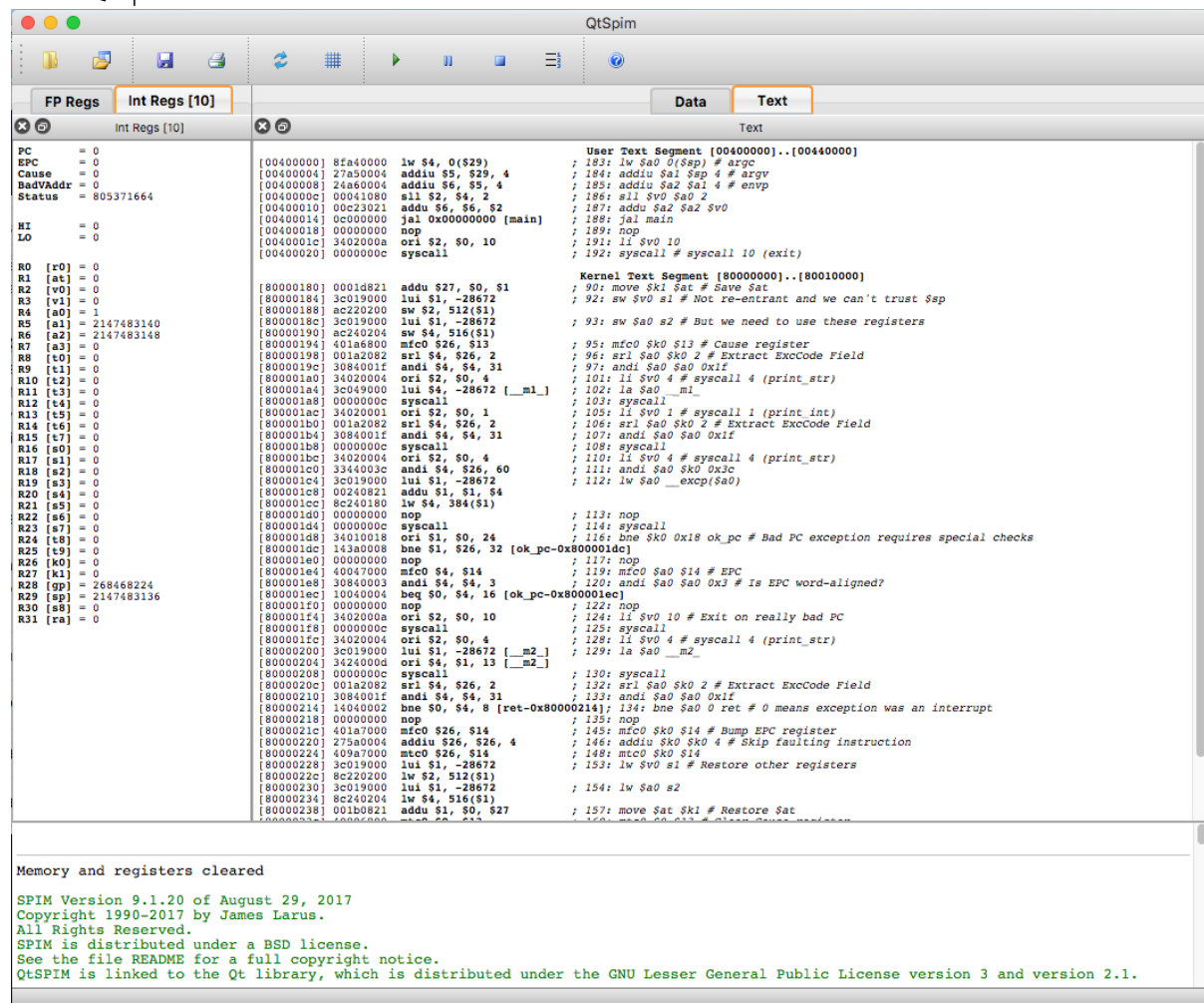
ในตอนเริ่มต้นให้นิสิตทบทวนความรู้เรื่อง MIPS assembly จากเอกสารประกอบปฏิบัติการนี้

เมื่อพร้อมจะทำปฏิบัติการแล้วให้ศึกษาและดาวน์โหลด QtSpim simulator ที่จะใช้รันโปรแกรม MIPS assembly ที่เราจะเขียนขึ้นได้จากลิงค์ต่อไปนี้

<http://pages.cs.wisc.edu/~larus/spim.html>

<https://sourceforge.net/projects/spimsimulator/files/>

เมื่อรัน QtSpim จะเห็น interface ในลักษณะตามด้านล่าง



สำรวจบริเวณ Text และ Data ที่ตัว simulator ให้สถานะตอนเริ่มต้นมาแล้ว ต่อไปบริเวณ Text และ Data จะเป็นที่เก็บคำสั่งและข้อมูลของโปรแกรม MIPS assembly ที่เราจะเขียนขึ้นตามลำดับ

ไปที่ Registers กับ Data Segment แล้วเลือกการแสดงผลเป็น Decimal

สำรวจบริเวณ Int Regs และ Data ว่า ณ ขณะนี้ค่าเริ่มต้นของ register และ memory ในส่วน User data segment มีค่าเป็นอะไรบ้าง

ดาวน์โหลดไฟล์ sample1.asm ที่แนบมากับปฏิบัติการนี้ ใช้ text editor เช่น ATOM หรือ VS Code เปิดขึ้นมา ศึกษาโปรแกรมที่ให้มาในไฟล์นี้ สิ่งที่มาด้านหลัง # ทั้งหมดคือคอมเมนต์ นิสิตบอกได้ไหมว่าโปรแกรมในไฟล์ sample1.asm ทำอะไร

โหลด sample1.asm เข้าไปที่ QtSpim โดยไปที่ File->Reinitialize and Load File สังเกตการเปลี่ยนแปลงที่เกิดขึ้นกับค่าใน register ต่างๆและค่า memory ในส่วน User data segment นอกจากนั้นให้สังเกตการเปลี่ยนแปลงที่เกิดขึ้นกับบริเวณ Text ด้วย

จากนั้นไปที่แท็บ Simulator แล้วเลือก Run/Continue สังเกตการเปลี่ยนแปลงที่เกิดขึ้นกับค่าใน register และ memory ในส่วน User data segment

เมื่อเข้าใจการทำงานของโปรแกรมและตัว QtSpim แล้ว เริ่มทำปฏิบัติการเขียนโปรแกรม MIPS assembly ตามข้อกำหนดที่ให้มาตามด้านล่างนี้ ห้ามใช้คำสั่ง branch/jump โดยเด็ดขาด

1. Answer.pdf

ตอบคำถามต่อไปนี้ลงในไฟล์ Answer.pdf

- เมื่อเริ่มต้น QtSpim สถานะของ Text Data และ Int Regs เป็นอย่างไร ให้อธิบายเป็นคำพูดแต่อาจมีการแคปหน้าจอประกอบคำอธิบาย
- เมื่อโหลดโปรแกรม sample1.asm เข้าไปที่ QtSpim สถานะของ Text Data และ Int Regs เป็นอย่างไร ให้อธิบายเป็นคำพูดแต่อาจมีการแคปหน้าจอประกอบคำอธิบาย
- หลังจากรันโปรแกรม sample1.asm สถานะของ Text Data และ Int Regs เป็นอย่างไร ให้อธิบายเป็นคำพูดแต่อาจมีการแคปหน้าจอประกอบคำอธิบาย
- เปลี่ยนการแสดงผลฟอร์แมตที่ Registers และ Data Segment เป็น Hex แล้วกลับไปสังเกตค่าที่ register และ memory ในส่วน User data segment ทำไมค่าที่เป็นจำนวนลบดูเหมือนจะ “มากกว่า” ค่าที่เป็นบวก

2. VectorSum.asm

นิยาม array 3 ตัวแต่ละตัวเป็น array of int ที่บรรจุค่า int ได้ 4 ตัว ให้ค่าเริ่มต้นของ array ตัวแรกเป็น

.word 100, 200, 300, 400

ค่าเริ่มต้นของ array ตัวที่สองเป็น

.word 10, 20, 30, 40

หาผลบวกของสมาชิกของ array ตัวที่หนึ่งและตัวที่สองที่ index ตำแหน่งเดียวกันและนำผลบวกนี้ไปเก็บไว้ที่ array ตัวที่สามที่ตำแหน่ง index เดียวกัน (นั่นคือการบวก vector สองตัวเข้าด้วยกันเพื่อให้ได้ vector ผลลัพธ์ตัวที่สามที่มีค่า [110 220 330 440] นั่นเอง)

3. DotProduct.asm

นิยาม array 2 ตัวแต่ละตัวเป็น array of int ที่บรรจุค่า int ได้ 4 ตัว ให้ค่าเริ่มต้นของ array ตัวแรกเป็น

```
.word 24, 13, 9, -16
```

ค่าเริ่มต้นของ array ตัวที่สองเป็น

```
.word 8, 7, -11, 3
```

หา dot product ของ array 2 ตัวนี้แล้วนำผลลัพธ์ที่ได้เก็บไว้ที่ register \$2

แหล่งข้อมูลเพื่อค้นคว้าเพิ่มเติม

- https://en.wikipedia.org/wiki/Dot_product

- เรื่องเกี่ยวกับการคูณใน MIPS assembly

```
mult $16, $17      # Multiply the numbers stored in these registers
                   # ($16 and $17). This yields a 64 bit number, which
                   # is stored in two 32 bits parts: "hi" and "lo"
```

```
mfhi $8            # loads the upper 32 bits from the product register
```

```
mflo $9           # loads the lower 32 bits from the product register
```

การส่งงาน:

นำไฟล์ที่เขียนเสร็จแล้ว ใส่ไว้ที่โฟลเดอร์:

studentID1_firstname1_studentID2_firstname2_lab0

โดย studentID และ firstname คือเลขประจำตัวและชื่อแรกของสมาชิกที่ทำปฏิบัติการร่วมกัน จากนั้น zip โฟลเดอร์นี้แล้วส่ง zip ไฟล์มาที่ Google Classroom ของวิชาก่อนกำหนดส่ง