

班級： 資工三 學號:B1043009 姓名:賴韋捷

1. 撰寫 MIPS 程式 (共 2 題，100 分，滿分 100 分)

請於 2023/11/28 前上傳至 M 數位園區作業區繳交

請安裝 QtSpim (<http://spimsimulator.sourceforge.net/>) 模擬器，並請詳細參考課本第二章及附錄 A 的介紹，於 QtSpim 模擬器環境下，撰寫一完整的 MIPS 核心指令集版本的程式。(需貼完整程式碼，截圖呈現結果並文字說明。)

- (1) 實作第二章 2.7 小節範例 if-then-else (中英文版 90 頁)，請自行完成變數設定，觀察暫存器及記憶體狀態並說明程式之運作。(50 分)

用記事本來撰寫程式，最後存檔使用.s 作為附檔名

再到 QtSpim 開啟檔案就可以了

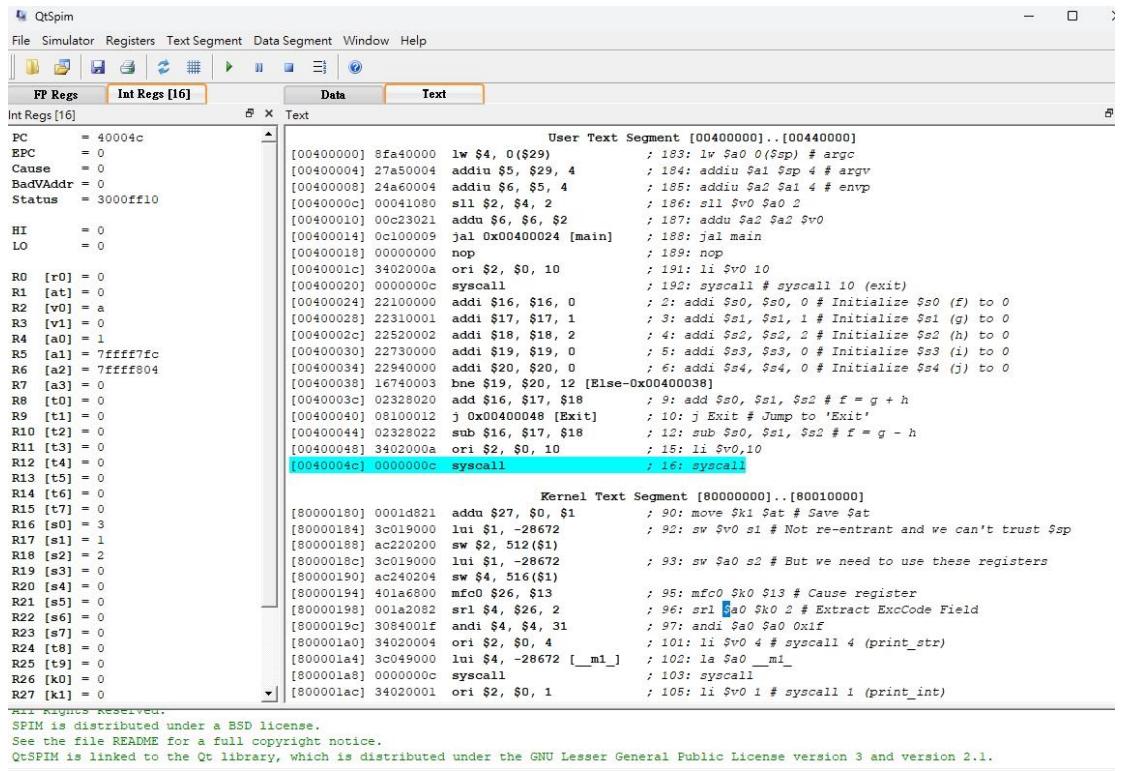
```
main:  
    addi $s0, $s0, 0      # Initialize $s0 (f) to 0  
    addi $s1, $s1, 1      # Initialize $s1 (g) to 1  
    addi $s2, $s2, 2      # Initialize $s2 (h) to 2  
    addi $s3, $s3, 0      # Initialize $s3 (i) to 0  
    addi $s4, $s4, 0      # Initialize $s4 (j) to 0  
  
    bne $s3, $s4, Else   # Branch to 'Else' if i != j  
    add $s0, $s1, $s2     # f = g + h  
    j Exit                # Jump to 'Exit'  
  
Else: sub $s0, $s1, $s2    # f = g - h  
  
Exit:  
    li $v0,10 #MIPS中，用 exit 的常用編號是 10  
    syscall
```

先初始化參數 f~j 對應 s0~s4

If else if 用 bne else 來寫

最後退出的時候 需要寫 li \$v0 ,10 因為這樣才能退出

讓程式成功執行



完成

- (2) 實作第二章 2.7 小節範例 while 迴圈 (中英文版 92 頁)，請自行完成變數設定，觀察暫存器及記憶體狀態並說明程式之運作。(50 分)

一樣使用記事本打程式

```

.data
    i: .word 0
    k: .word 5
    save: .word 5, 5, 5, 15, 5

.text
main:
    lw $s3, i      # i = $s3
    lw $s5, k      # k= $s5
    lw $s6, save # save = $s6

Loop:
    sll $t1, $s3, 2  # i * 4
    add $t1, $t1, $s6 # save[i]
    lw $t0, 0($t1)   # save[i] = $t0

    bne $t0, $s5, Exit # if save[i] != k

    addi $s3, $s3, 1  # i++
    j Loop          # Jump back to Loop

Exit:
    li $v0, 10
    syscall |

```

因為程式要寫 while(save[i]==k)
i+=1;

所以我們一開始設定 k 是 5 , i 是 0 , save 的陣列[5,5,5,15,5]

s3 是 i

s5 是 k

s6 是 save 陣列

準備好前置作業後

進入到 while 迴圈

一樣寫 Loop

設定 i 記憶體

Save[i]

If else 判斷

如果 save[i]!=k

i++

所以我的陣列會讓 i=3

The screenshot shows the QtSpim simulator interface. The top menu bar includes File, Simulator, Registers, Text Segment, Data Segment, Window, and Help. The main window has tabs for FP Regs, Int Regs [16], Data, and Text. The Text tab displays assembly code with comments. The Int Regs [16] tab shows register values. A status bar at the bottom provides copyright information.

Assembly code (Text tab):

```
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2 $v0
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # syscall 10 (exit)
[00400028] 3c011001 lui $1, 4097 ; 7: lw $s3,i
[0040002c] 8c330000 lw $19, 0($1) ; 8: lw $s5,k
[00400030] 3c011001 lui $1, 4097 ; 9: lw $s6,save
[00400034] 8c350004 lw $21, 4($1) ; 10: lw $s7,main
[00400038] 8c3e0008 lw $22, 8($1) ; 11: lw $s8,main
[0040003c] 00134980 sll $9, $19, 2 ; 12: sll $t1,$s3,2
[00400040] 01364820 add $9, $9, $22 ; 13: add $t1,$t1,$s6
[00400044] 8d280000 lw $8, 0($9) ; 14: lw $t0,0($t1)
[00400048] 15150003 bne $8, $21, 12 [Exit-0x00400048]
[0040004c] 22730001 addi $19, $19, 1 ; 15: addi $s3,$s3,1
[00400050] 0810000f j 0x0040003c [Loop] ; 17: j Loop
[00400054] 3402000a ori $2, $0, 10 ; 19: li $v0,10 #MIPS中用 exit 的常用編號是 10
[00400058] 0000000c syscall ; 20: syscall
```

Kernel Text Segment [800000000]..180010000

```
[80000180] 0001d921 addn $27, $0, $1 ; 90: move $k1 $at # Save Sat
[80000184] 3c019000 lui $1, -28672 ; 92: sv $v0 $1 # Not re-entrant and we can't trust $sp
[80000188] ac220020 sw $2, 512($1) ; 93: sv $a0 $2 # But we need to use these registers
[8000018c] 3c019000 lui $1, -28672
[80000190] ac240204 sw $4, 516($1)
[80000194] 401a6800 mfcr $26, $13 ; 95: mfcr $k0 $13 # Cause register
[80000198] 001a2082 sr1 $4, $26, 2 ; 96: sr1 $a0 $k0 2 # Extract ExcCode Field
[8000019c] 3084001f andi $4, $4, 31 ; 97: andi $a0 $a0 $1xf
[800001a0] 34020004 ori $2, $0, 4 ; 101: li $v0 4 # syscall 4 (print_str)
[800001a4] 3c049000 lui $4, -28672 [_m1_] ; 102: la $a0 _m1_
[800001a8] 0000000c syscall ; 103: syscall
[800001ac] 34020001 ori $2, $0, 1 ; 105: li $v0 1 # syscall 1 (print_int)
```

Registers (Int Regs [16] tab):

Reg	Value
PC	0
EPC	0
Cause	0
BadVAddr	0
Status	3000ff10
HI	0
LO	0
R0 [r0]	0
R1 [at]	0
R2 [v0]	0
R3 [v1]	0
R4 [a0]	1
R5 [a1]	7ffff7fc
R6 [a2]	7ffff804
R7 [a3]	0
R8 [t0]	0
R9 [t1]	0
R10 [t2]	0
R11 [t3]	0
R12 [t4]	0
R13 [t5]	0
R14 [t6]	0
R15 [t7]	0
R16 [s0]	0
R17 [s1]	0
R18 [s2]	0
R19 [s3]	0
R20 [s4]	0
R21 [s5]	0
R22 [s6]	0
R23 [s7]	0
R24 [t8]	0
R25 [t9]	0
R26 [k0]	0
R27 [k1]	0

Copyright notice:

QtSpim is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.