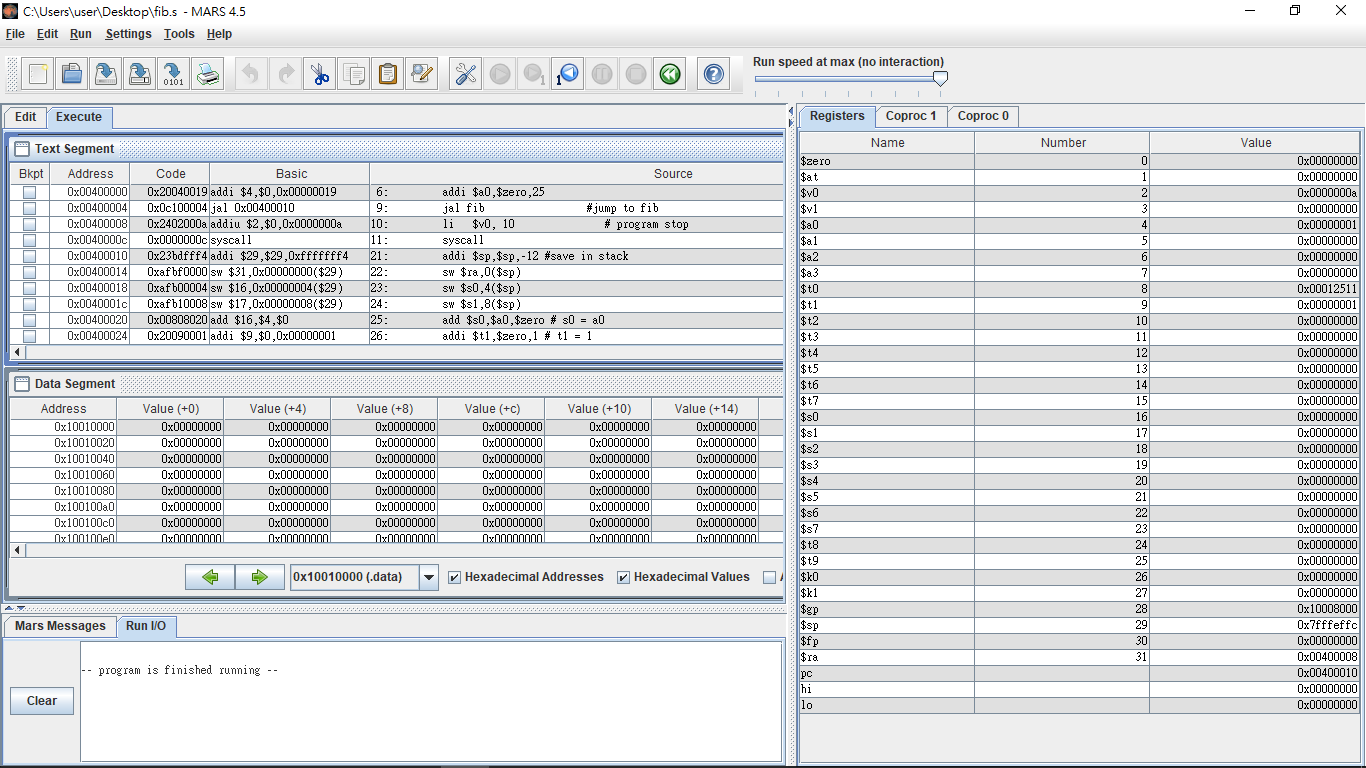
**Computer Organization 2018**

**HOMEWORK I**

系級: 資訊108 學號: C14041141 姓名: 王韋凱

**實驗結果圖:**



**程式運作流程:**

先將25 assign 給$a0當作呼叫fib 函式的參數也就是fib(n)的n

在fib內先將 save 要用到的 $ra(return address 因為recursive)

$s0(用來備份$a0的值) $s1(用來儲存fib(n-1)) 到 stack

然後判斷該函式＄a0(n) 的值，為0則跳到Return0，為1跳到Return1

Return 0:將 0 assign 給$v0(return) 然後restore $ra $s0 $s1 再return

Return 0:將 1 assign給$v0(return) 然後restore $ra $s0 $s1 再return

若都不是(n>=2)則先把 $a0 減一再呼叫fib => $v0 = fib(n-1) 然後

將$v0 assign 給$s1 ($s1=fib(n-1) ) 再把$a0 減二再呼叫(由於有s0備份所以直接將s0-2的值assign給a0即可 前面的a0-1也是)fib=>$v0 = fib(n-2)

再把v0+s1的結果assign回$v0($v0=fib(n-1)+fib(n-2) ) restore 再把return value($v0) assign給 $t0 最後return

**心得**

此作業若用C語言來寫的話大概5到6行就結束了

但是用MIPS寫的話函式參數的運作就需要仔細思考

而經由這次作業我學到了呼叫函式在MIPS中須考慮到許多register在stack的運作 包括save 和 restore 和 如何用MIPS作條件判斷(if)

而我覺得最困難的部分就是 要將原本短短的幾行函式(在C) 變成許多指令來組合實現 例如函式在 leaf 跟 nonleaf的類別所要save的也不同 而每個return的參數若不只一個的話還要另外先備份以及條件判斷要拆成2,3個步驟