組合語言與系統程式作業二報告

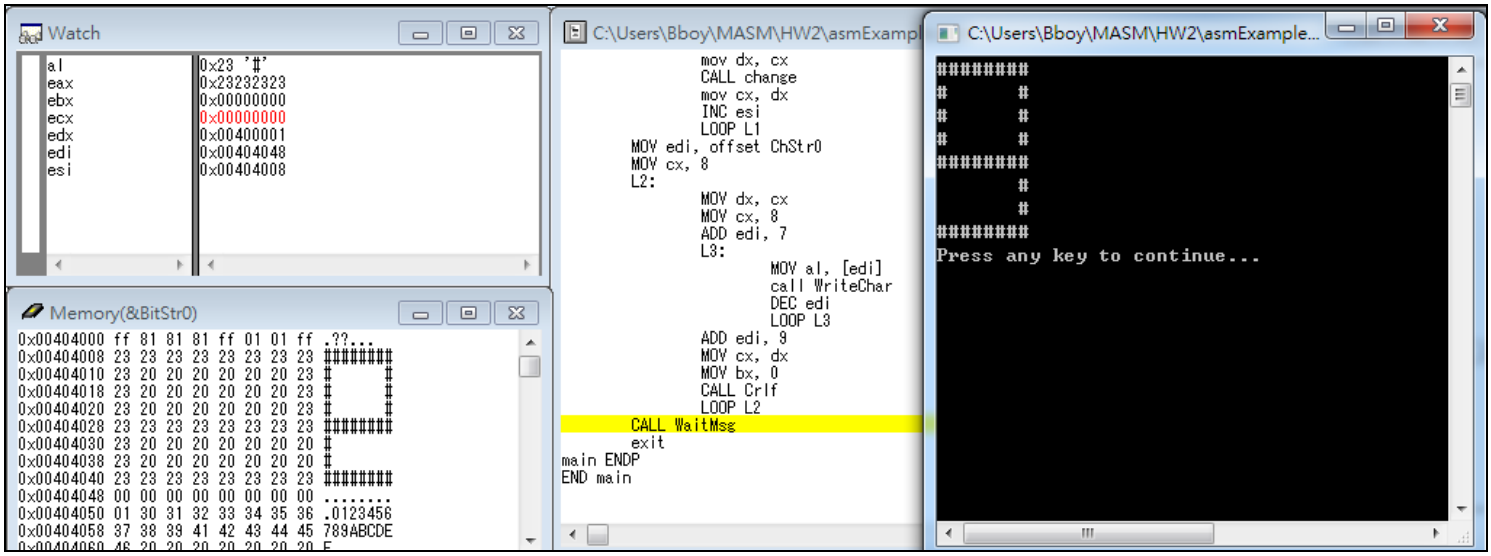
102502559 資工2B 吳承霖

程式流程及原理

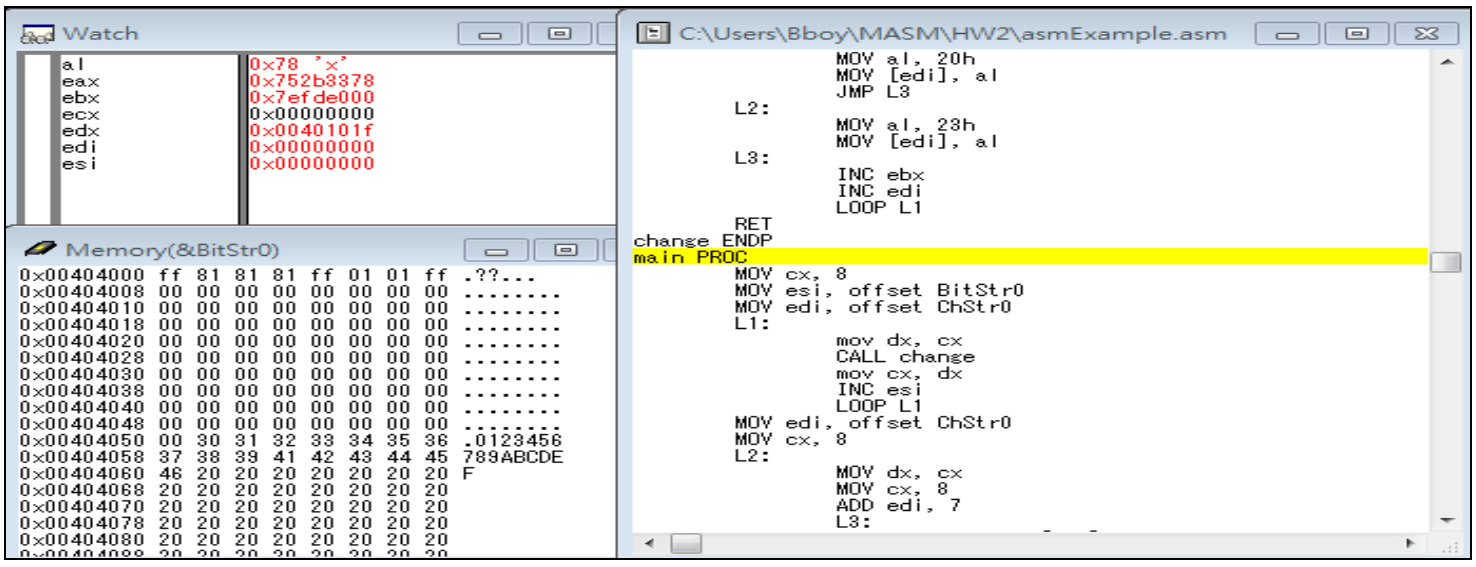
程式一開始宣告初始值(學號末一碼)進入BitStr，以1和0分別代表數字圖案的網格為＃或是空白，那是如何判斷一個BYTE內的所有Bit為0或1呢？在change這個procedure中，我使用了"BT"這個指令，右圖為此次作業核心處理的程式碼片段，以"BT eax, ebx"解釋，程式會執行"檢查eax暫存器的第ebx，如果該bit值為1，則改變CF為1"。執行完BT後我緊接著依照CF做條見判斷，如果CF為1，表示此變數的該bit為1，會跳至L2並且把＃寫進ChStr中。

```
L1:
    MOV eax, [esi]
    BT eax, ebx
    JC L2
    MOV al, 20h
    MOV [edi], al
    JMP L3
L2:
    MOV al, 23h
    MOV [edi], al
```

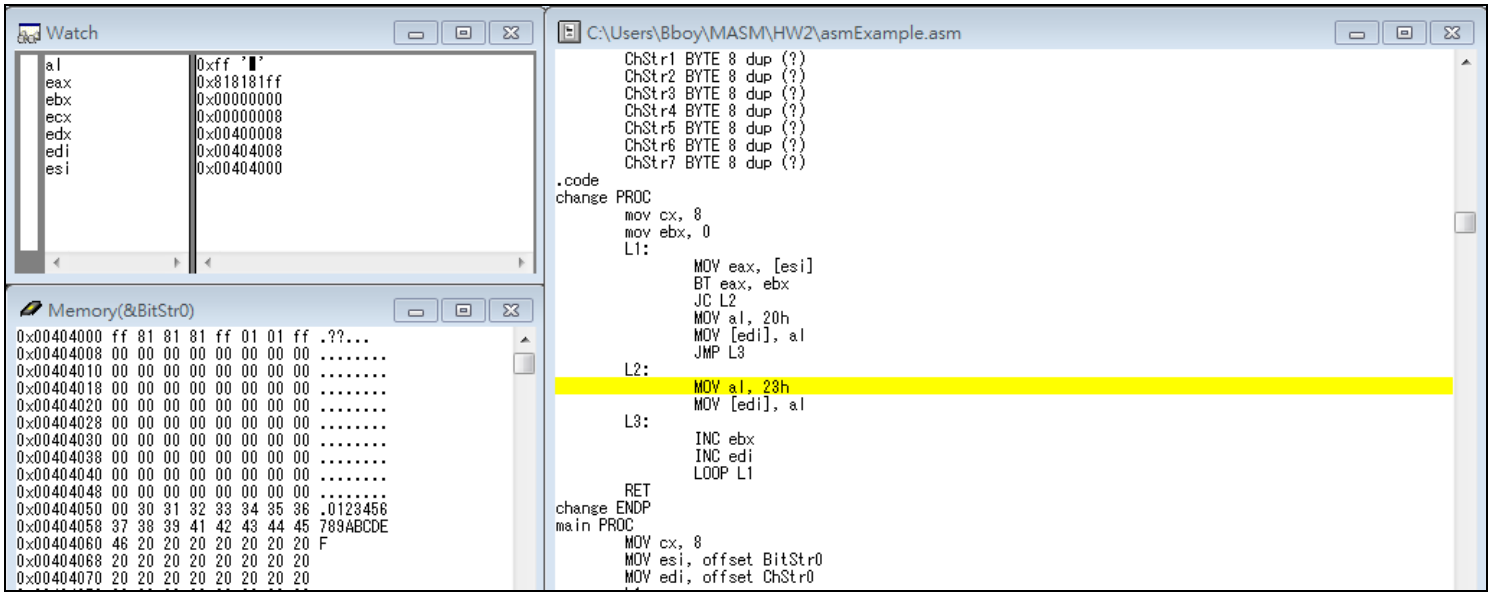
完成畫面截圖



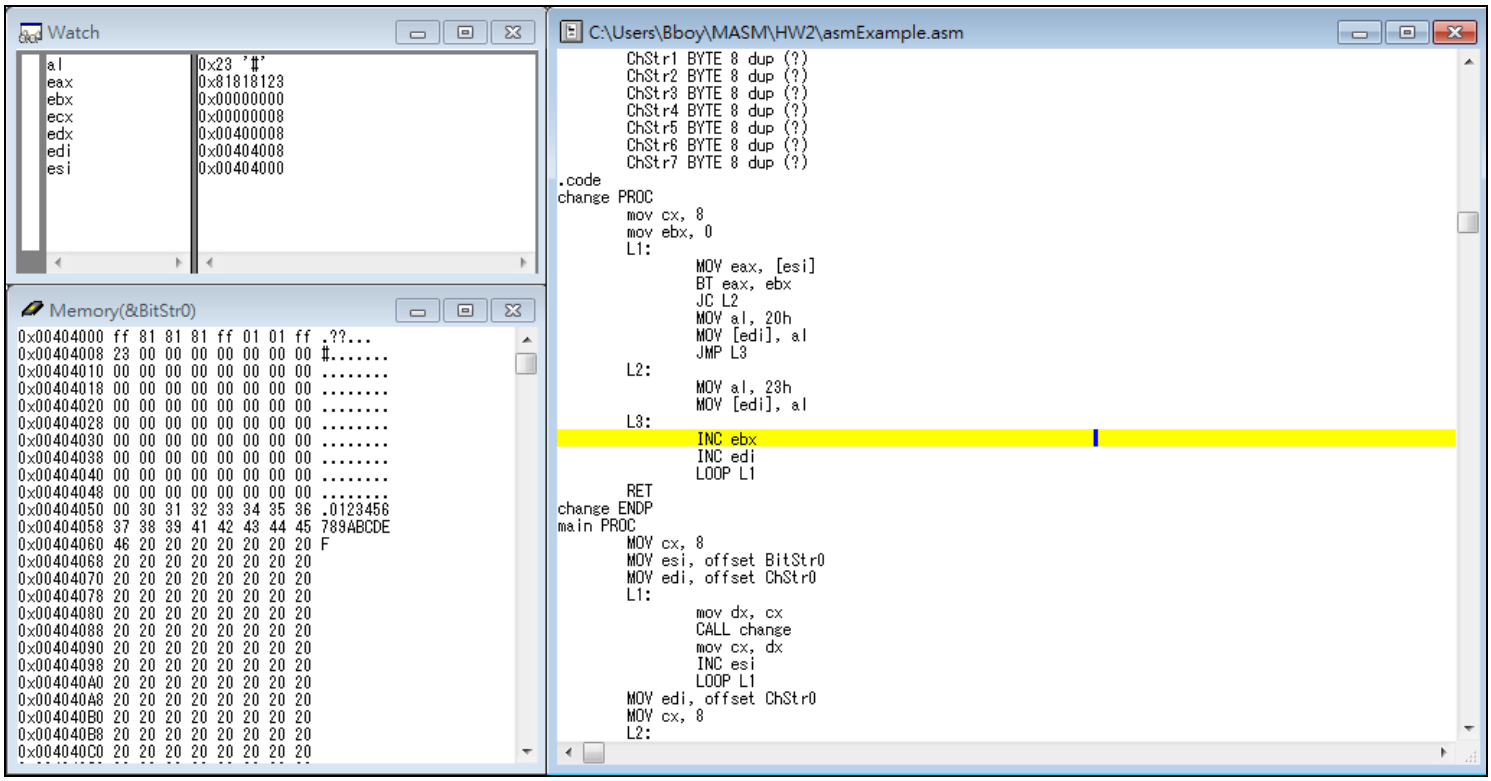
記憶體＆暫存器狀態圖



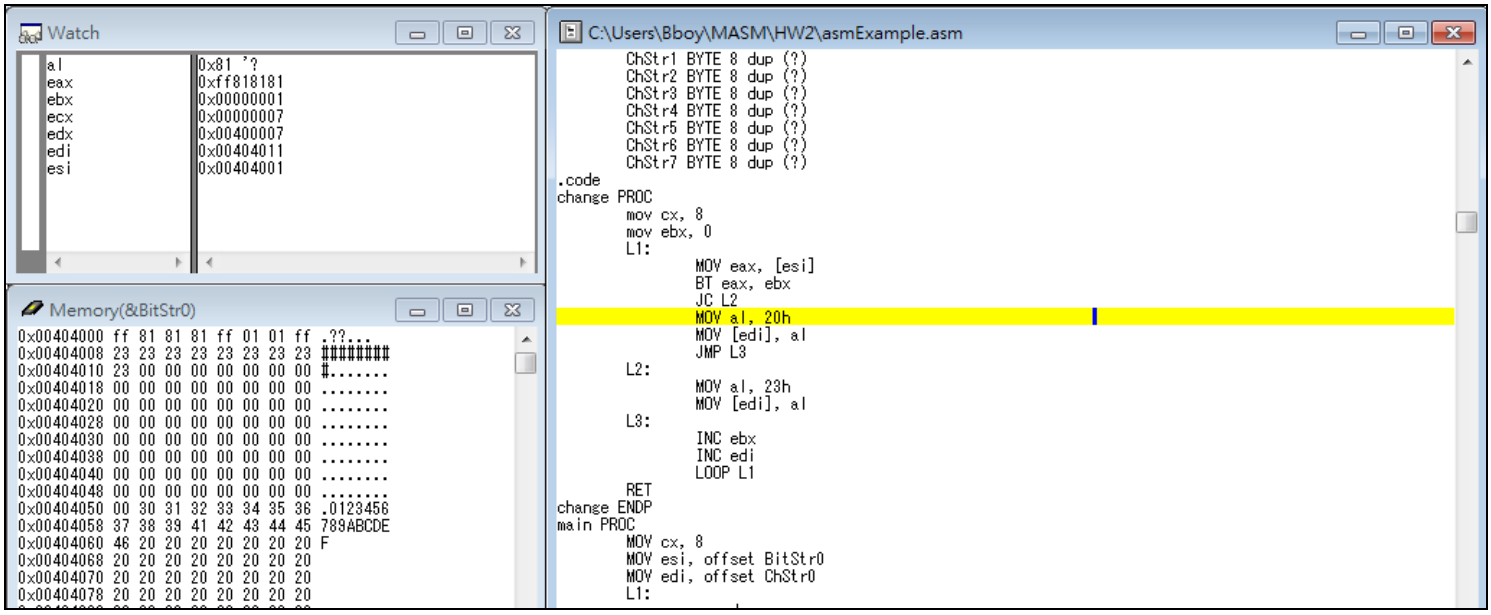
程式開始的狀態，我們可以看到0x00404000的記憶體位置內已經初始化成學號末一碼



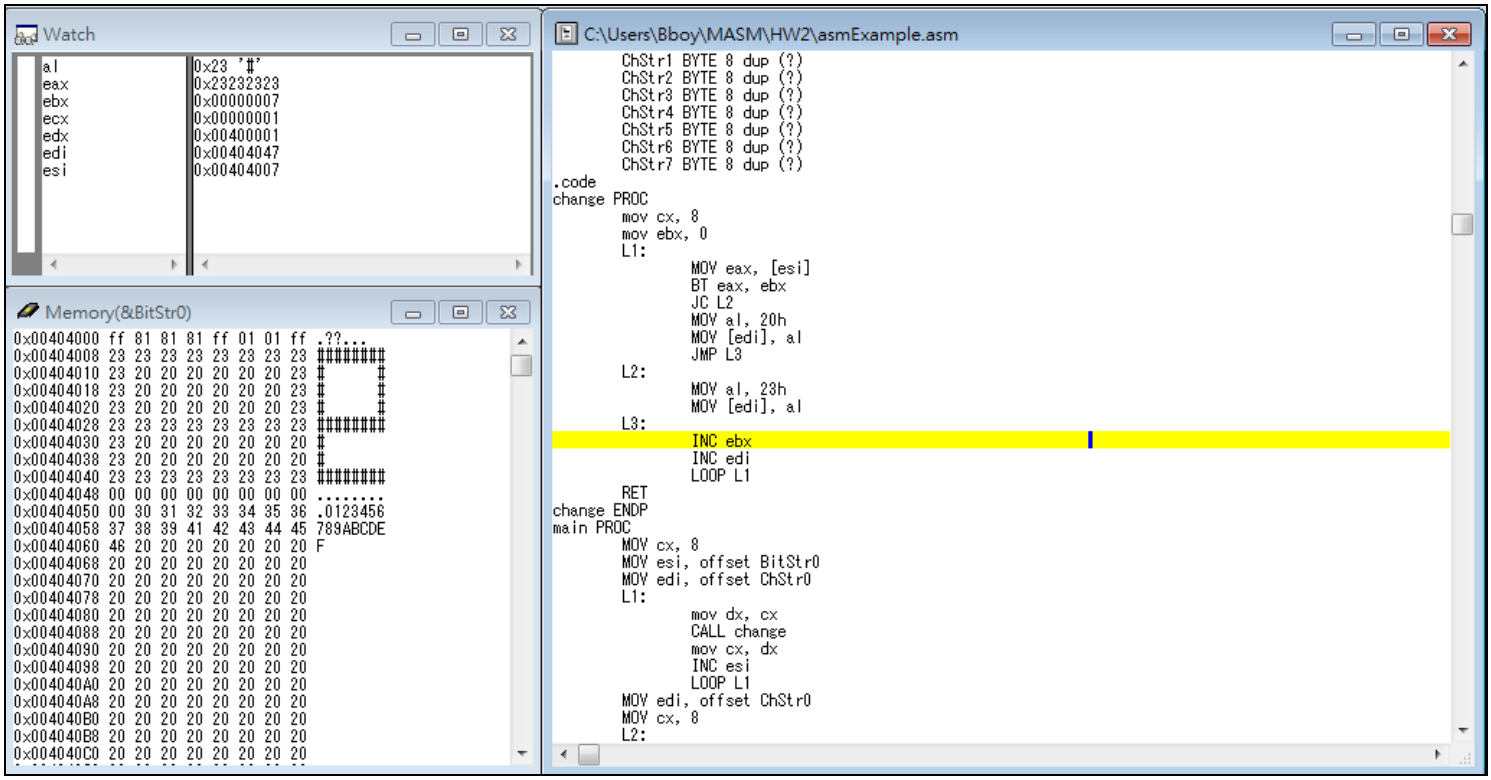
在L1內，JC指令會判斷CF是否為1，若是1的話代表eax暫存器內當前判斷的bit為1，程式會進入L2執行



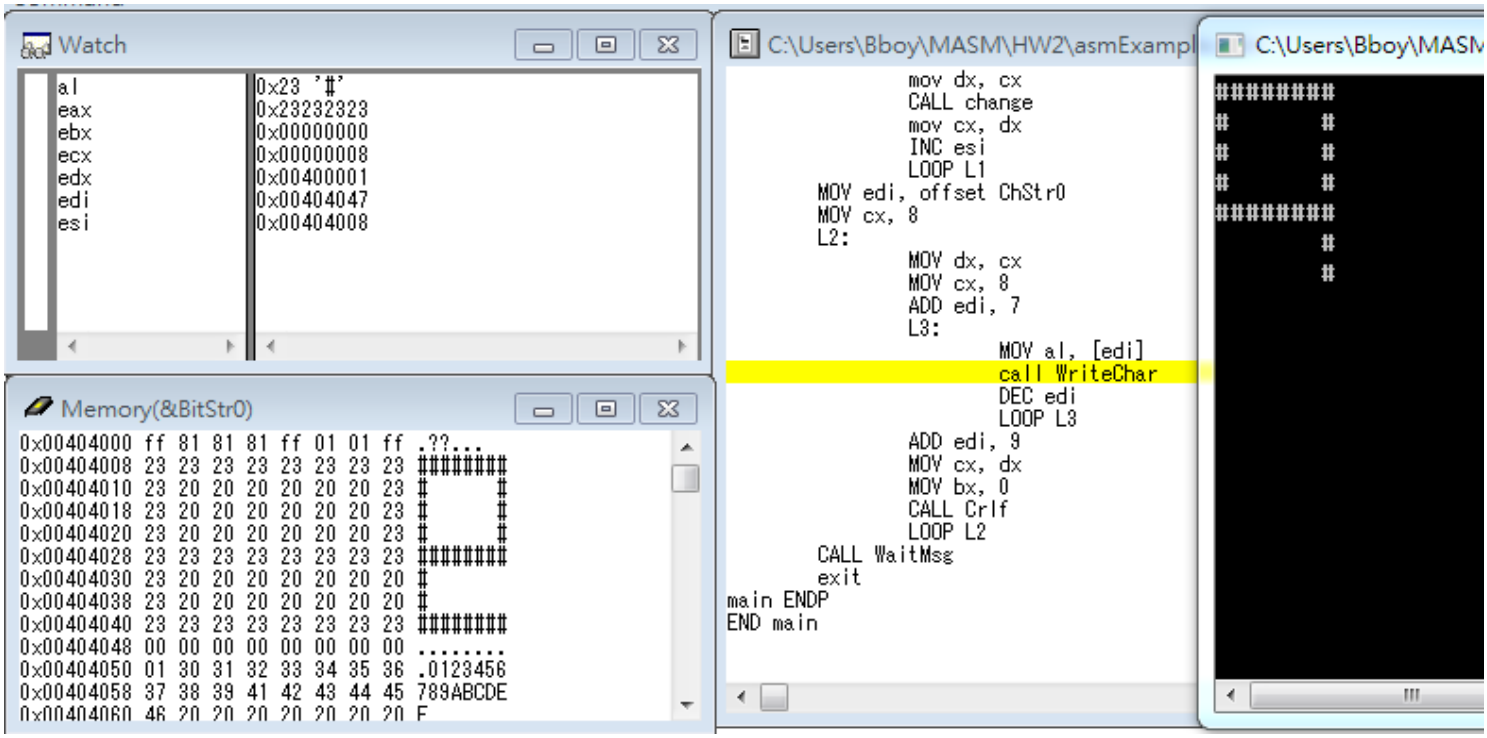
現在是L2內的程式碼執行完1伺候的狀態圖，我們可以看到記憶體位址0x00404008內存了23h的值，以ASCII表示即為#



此圖為BT第一次判斷到bit內是0的情況，在此情況下，不會跳到L2執行，而是繼續執行L1的所有指令。



此狀態圖為存入空白或是#字號進入每個ChStr中，但我們發現若按照記憶體位置由小到大依序印出來，會是左右顛倒的圖形。



所以我在WriteChar的迴圈中，以反序的方式印出來，如此一來便是正確的執行結果了。

心得與感想

這次作業跟前幾次上機或是作業相較起來，是比較複雜且規模較大的程式，這次需要的應用也包含了之前教過的procedure和巢狀迴圈，終於有一個實際的例子讓我練習了，覺得寫組語這麼底層的語言可以了解高階語言時尚是怎麼運作的是一件很好玩的事情。不過我發現，越複雜的程式，除了會有更複雜的邏輯，也會遇到暫存器不夠的問題，如何有效且能重複地利用暫存器不會干擾到其他行程，是我必須去注意的事項。