期末計畫

40747008S 陳泓銷

計畫簡介

此次計畫為實作 SIC/XE 的 assembler，使用的語言為 C，本計畫只完成到產生 object code。

實作過程與問題

一開始就決定使用 window 系統+C 語言開發，並上網查找資料，找到一個使用 java 開發的 SIC assembler，這個 source 給我讀取檔案中文字的靈感，之後又找另一個 C 語言開發的 SIC assembler，這個 source 則給我 location 計算的參考。

讀取檔案的部分是互動式輸入，需要手動輸入檔明才會開啟，增加了開啟檔案的便利性。

從開發初期就遇到讀取檔案中每一行字數的問題，因此苦思許久後決定用檔案中的換行符號、空格與 TAB 當作判斷標準，不過後來也改成讀取整行，用 fgets 讀取，再使用 strtok 進行拆分。為了方便再次讀取，我將空格的部分都用"--"代替。並將原檔案的文字加上當前 location 輸出為 loc.txt。

之後就是 location 的計算，使用 if-else 判定是 format 2 or 3 or 4 或是變數，再根據每種不同的 mnemonic 計算 location。

接下來是 symbol table 的製作，這個部分相對輕鬆，只要讀取 loc.txt 中的 label 與 address，並將其輸出為 symtab.txt。再來是讀取 symtab.txt，將 label 與 address 存入陣列中。這兩個步驟其實可以在讀取 loc.txt 的時候就可以一併完成。

接下來是 object code 的計算，這部分我歸納了很久，後來得出一個結論，先將 mnemonic 對應的 opcode(HEX)從預先儲存的 opcode 陣列中找出，再來根據不同的 format type 與變數進行分類，分類完成後再根據 operand 的不

同再次分類。

在這裡我遇到有些 operand 是 register，為了判別這些 register，我寫了一個 FUNCTION checkre(char op) 來將 register 轉為對應的數字，除此之外還遇到了 disp < -2048 的情況，因此在上面補上了找 BASE location 的步驟。

之後又遇到 disp < 0 的情況，原本上網找資料給的解法是找尋補數+1，不過應該是資料型別設定的錯誤，每次算出來的位數都會遠大於本來的，嘗試很久都沒辦法弄好，最後寫了一個 FUNCTION tcom(int disp) ，以二進位的方式計算補數+1，再將結果放回 disp。

接下來把原先的資訊加上 object code 一起輸出產生 object_code.txt，到此程式結束。

程式執行

開啟 EXE 檔，輸入來源檔案名(含副檔名)，即可產生 location file、symbol table 與 object code file.

未來展望

1. 完成 object program
2. 將重複使用的步驟寫成 function 以節省空間與方便呼叫

心得

因為對 C 語言讀取檔案資料方法、address 計算、object code 計算、C 語言字串使用的不熟悉，在了解這些部分花費大多時間，所以 object program 的部份沒有完成，不過此次計畫讓我對於這些有了更全面的認識，尤其是學會使用很多 C 語言內建的函式。

參考資料

1. https://github.com/andy6804tw/SIC
2. https://github.com/ashutoshvrm8/Assembler

3. https://hackmd.io/@wei-coding/HkiIr4pdD#Memory1
4. https://kalinlai-void.github.io/2021/04/22/System-Software-Chapter1/