SIC Assembler實做報告

組員:

24 D0511192 錢思翰

47 D0542367 蔡凱勛

58 D0542694 方品中

63 D0542889 鄧宇淇

* 開發語言及平台   
  我們使用PYTHON作為開發語言，並透過其IDLE做DEBUG及確認結果。原因是相較於C及JAVA等其它我們學過的語言，PYTHON在處理字串及檔案讀寫有提供方便的FUNCTION可以呼叫，在程式邏輯編寫上也更加簡潔，降低錯誤機率。
* 格式的輸入

需要組譯的檔案名稱須為”SRCFILE”，內容格式則與一般SIC組合語言程式相同即可。由於我們使用PYTHON的split()方法切割字串，組合語言之指令排版不須特別嚴謹，只要在指令單詞間插入空格且正確換行(每行僅1指令)便可以正常讀取。

* Addressing Modes

此組譯器為SIC基本版專用，因此我們程式能夠正確編寫之Addresssing Modes為一般模式(TA = ADDRESS)以及運用到X暫存器之模式(TA = ADDRESS + (X))。

* Assembler Directives  
  可以正常處理的組譯器假指令包含start, end, resw, resb, word, byte。即SIC基本版假指令皆可執行。
* Data Structures

說明程式內所用變數之功能:

PASS 1:

Opcode ，存放指令之opcode，型態為字典。

Out ，pass 1輸出之中間檔。

LOCCTR ，計算個指令記憶體位址的站存變數，當記憶體位置都計算完後便無用處，因為位置會與指令內容憶起被另外記錄。

Arr ，將SRCFILE之內容以split()切割除來的指令list。會逐一編寫位址及Object code。

SYMTAB ，pass 1 紀錄標籤位址。

PASS 2:

arr，相當於LISFILE的格式

obcode ，依各種指令編寫object code時用到的暫存。

temp(in lisfile)，用來站存字串，完成後存入file並歸回""

Ctemp(in lisfile)，當讀到BYTE且變數為C' '時，將規定之存入Ctemp

OBJFILE H內

Hcard為string型態，最終將存入file內

ProgramName=arr[0][1]，將整個程式的程式名稱存入，待會會加入Hcard中

ProgramStart=arr[0][0].zfill(6)，將開始位置存入，透過.zfill(6)，將其定為6位

ProgramEnd，程式結束位置

ProgramLen等於結束-開始，為程式總長度

OBJFILE T內

T為string型態，最後會將成功的T卡片寫入FILE內

check，若是有失敗的T卡片，會將程式結束，並提醒之後的E卡片不需要做

TTEMP為string，此提供占存的功用，可以用來計算以及增加項目

Ctemp(in OBJFILE)，當讀到BYTE且變數為C' '時，將規定之存入Ctemp

Tlen，T卡片的長度，用來避免卡片長度大於60

* 程式流程

(pass 1)

我們在處理讀入的SRCFILE的時候，先以split()方法擷取出指令，並檢查是否有定義start，若有，將location counter設定初始值，否則訂為0。之後開始計算每個指令之地址且紀錄標籤地址。之後輸出至中間檔。

(pass 2)

開啟中間檔讀取地址及指令，並依各指令計算object code(查表及計算TA後，轉換為16進制儲存)。

當所有的程式碼計算完成之後，換成使用List arr，內有5個變數，分別是位置、程式敘述、程式碼、變數、OBCODE。

最後在寫入LISFILE及OBJFILE時，是將一句一句string做好之後，利用file.write()寫入對應之file中。