

SIC XE Assembler

系統程式書面報告

資訊二丙 座號 : 82 學號 : D0588813 姓名 : 曾玉鳳
資訊二丙 座號 : 54 學號 : D0542900 姓名 : 董育汝
資訊二丙 座號 : 18 學號 : D0511203 姓名 : 陳品樺
資訊二丙 座號 : 21 學號 : D0511305 姓名 : 范瑋軒

Assembler 開發語言及平台：使用 Python

處理步驟：

Step1: 輸入檔案名稱，將檔案一行一行讀入，如果中間讀到 tab，則將 Tab error 寫入 Lisfile 檔並停止程式，其他則透過位置將字串分割出標籤、opcode、operand，並使用 Python .strip() 將空格清除，最後儲存至 list 中。

Step2: 區分出虛擬指令。我們使用 if-else 來分別出 START、RESB、RESW、BYTE、WORD、END、ORG 做額外處理，並計算位址加入最前方欄位，及判斷有無+、X 的符號，如遇 END 就結束 Step2。

①遇到「.」註解行、「BASE」即省略。

②遇到「START」設定起始位置，計算 Location 時以 10 進位運算，算完再轉至 16 進位。

③遇到「ORG」，若其後面有接標籤，則位置直接將位址跳至標籤位置，並儲存原本位置，若其後沒標籤，則跳回原本 PC 位址，若上述都不符合則顯示 ORG 指令錯誤。

④遇到 label，則另外存至 sym={} (dictionary)。

⑤遇到「WORD」轉為 16 進位，直接計算 WORD 及 BYTE 的 object code，並計算下一個指令位址(+3)。

⑥遇到「BYTE」中的字元'C'，則將其轉換成 16 進位 ASCII 碼，並將下一位址增加其字元長度。而 16 進位數字'X'，則將其數字直接存入，如不滿 1byte 則補空格。

⑦遇到「RESB」、「RESW」，增加空欄位到最後一行，並計算下一個指令位址加入欄位中。

⑧遇+格式指令，計算下一個指令位址加入欄位中(+4)。

⑨遇到有 X 的 opcode，則切割至', '前之 opcode，並放入該位址第一位加 8，轉成 16 進位，放入該位址後 3 位。

⑩剩餘指令依 op dictionary 內的指令長度計算下一個指令位址加入欄位中。

⑪若上述都不符合，則顯示 WRONG OPCODE!

Step3: 計算其餘的 object code 加入欄位，如遇 END 就結束 Step3。

①忽略 ORG、BASE、註解。

②遇到「BASE」將 B 設為此標籤位址。

③遇到+格式指令，若為「索引模式,X」，將切割至', '前之 opcode 指令+3 第三位設為 9 加上位址，若為「立即模式#」，且為#數字，將指令+1 第三位設為 1 加上位址，若為#標籤，則需另編寫 M 卡片，將位址+1 加到 M 陣列裡，若不為上述，則將指令+1 第三位設為 1 加上位址。

④遇到索引模式,X，則切割至', '前之 opcode，將指令+3 第三位設為 9 加上位址。

⑤若指令在 $op=\{ \}$ (dictionary)裡。遇到「立即模式#」，判斷為#數字，將指令+1 加上位址，若為#標籤，判斷範圍適於減 PC，則將指令+1 第三位設為 2 加上(位址-PC)，判斷範圍適於減 B，則將指令+1 第三位設為 4 加上(位址-B)，非上述 print 錯誤訊息。遇到「間接模式@」，判斷範圍適於減 PC，則將指令+2 第三位設為 2 加上(位址-PC)，判斷範圍適於減 B，則將指令+2 第三位設為 4 加上(位址-B)，非上述 print 錯誤訊息。

⑥若為 format 2，將切割至', '前之 opcode 加上暫存器編號，非上述 print 錯誤訊息。

⑦非為上述則為一般，將指令+3 第三位設為 2 加上(位址-PC)，判斷範圍適於減 B，則將指令+3 第三位設為 4 加上(位址-B)，非上述 print 錯誤訊息。

Step4：將內容依格式讀出寫成 LISTFILE.txt 檔並輸出。

Step5：將內容讀出寫成 OBJFILE.txt 檔並輸出。

①輸出 H 卡片。

②將上面編好的 object code 依序寫入 T 卡片，若長度不滿 30，則繼續。

③寫入若超過或下行 object code 為空，則換一張 T 卡片繼續編寫

如果遇到「.」、空字串(第一行)，表示是註解、RESB、RESW，省略不寫入。

④如果遇到 END，則先將前一行 T 卡片未完成字串輸出，從 M 陣列裡輸出 M 卡片並加上修改位置 05，再輸出 E 卡片。

輸入格式:

不可使用 tab

英文大、小寫皆可輸入

Label、指令及 operand 起始位置有固定要求(1-9，10-17，18-)

不一定要有 label

註解可新的一整行，不可放於後面

可處理的 addressing modes 和 assembler directives:

addressing modes : Direct addressing、Indirect addressing、

Extend format、Immediate addressing、

Index addressing、Relative addressing

assembler directives: START、END、WORD、BYTE、RESW、RESB、

BASE、ORG

Function :

hex(): 十進制轉 16 進制

ord(): 將字元轉為 ASCII

append(): 將東西新增在列表後方

insert(): 將指定對象插入列表的指定位置

zfill(): 將字串前方補 0，使之達指定字串長度

upper(): 將小寫轉大寫

split(): 通過指定分隔符對字串進行切割

strip(): 去除字頭字尾指令字元(默認為空白鍵)

math.ceil(): 取整數，無條件進位

data structures :

xe list 結構 : 很多個 list，每個 list 包含多個小 list

[[[10 進位, 16 進位], label, opcode, operand, object code], [.....],]

Ex. [[[8192, '2000'], 'HW1SIC', 'START', '2000', ''],
[[8192, '2000'], 'FIRST', 'LDA', 'AA', '00203C'],
[[8195, '2003'], '', 'ADD,X', 'BB', '18A03F'],
[[8198, '2006'], '', 'STA,X', 'CC', '0CA042'],]

sym 結構 : { 'label 名稱' : 'location', }

Ex. { 'AA': '203C',
 'BB': '203F',
 'CC': '2042',
 'DD': '2045',
 'EE': '2048',
 }

op 結構 : { '指令名稱' : ['opcode', ' ', ' '] }

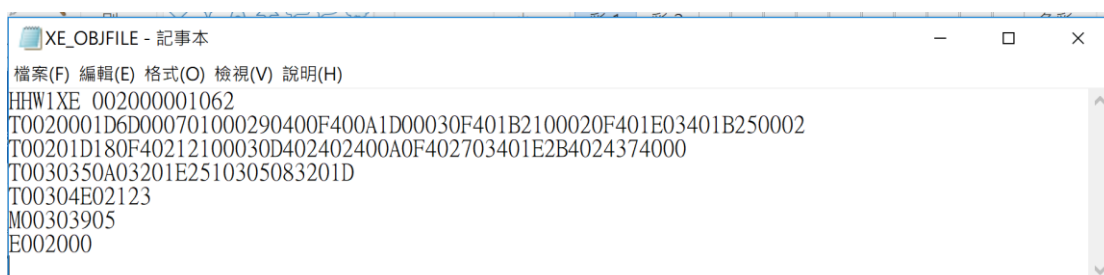
Ex. op = { 'ADD': ['18', 3], 'ADDF': ['58', 3], 'ADDR': ['90', 2], 'AND': ['40', 3], }

reg 結構 : { '暫存器' : '暫存器編號' }

Ex. reg = { 'A': '0', 'X': '1', 'L': '2', 'B': '3', 'S': '4', 'T': '5', 'F': '6' }

輸出結果 :

OBJFILE :



LISFILE :

```

XE_LISFILE - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
2000      HW1XE  START  2000
2000 6D0007  FIRST  LDS   #7
2003 010002      LDA   #2
          BASE   IF
2006 9040      ADDR  S,A
2008 0F400A      STA  CC
200B 1D0003      SUB  #3
200E 0F401B      STA  EE
2011 210002      MUL  #2
2014 0F401E      STA  FF

2017 03401B      LDA  EE
201A 250002      DIV  #2
201D 0F4021      STA  GG
2020 210003      MUL  #3
2023 0D4024      STA  #HH
2026 02400A      LDA  @CC
2029 0F4027      STA  II
202C 03401E      LDA  FF
202F 2B4024      COMP HH
2032 374000      JGT  IF
2035      RESB   4096

3035 03201E  IF    LDA  GG
3038 25103050 +DIV  #EE
303C 83201D  STF   II

.
303F      CC      RESW  5
304E 123    EOF    BYTE  X'123'
          ORG     CC
303F      RESB   1
          ORG

3050      EE      RESW  1
3053      FF      RESW  1
3056      GG      RESW  1
3059      HH      RESW  1
305C      II      RESW  2
3062      END     FIRST

```

錯誤訊息範例 :

Tab Error :

-測試程式碼 :

HW1XE	START	2000			
FIRST	LDS	#7			
	LDA	#2			
	BASE	IF			
	ADDR	S,A			
	STA	CC	IF	LDA	GG
	SUB	#3		+DIV	#EE
	STA	EE		STF	II
	MUL	#2			
	STA	FF			
	AAA	4096	CC	RESW	5
	LDA	EE	EOF	BYTE	X'123'
	DIV	#2		ORG	CC
	STA	GG		RESB	1
	MUL	#3		ORG	
	STA	#HH	EE	RESW	1
	LDA	@CC	FF	RESW	1
	STA	II	GG	RESW	1
	LDA	FF	HH	RESW	1
	COMP	HH	II	RESW	2
	JGT	IF		END	FIRST
	RESB	4096			

-測試結果：

```
****Tab Error !  
An exception has occurred, use %tb to see the full traceback.
```

SystemExit: Tab Error !

```
C:\Users\User\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2918: UserWarning: To exit: use  
'exit', 'quit', or Ctrl-D.  
    warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```

Wrong Opcode：

-測試結果：

```
****Wrong OPCODE !  
An exception has occurred, use %tb to see the full traceback.
```

SystemExit: Wrong OPCODE !

```
C:\Users\User\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2918: UserWarning: To exit: use 'exit',  
'quit', or Ctrl-D.  
    warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```

Relative error：

-測試程式碼：

1	HWLXE	START	2000
2	FIRST	LDS	#7
3		LDA	#2
4	DD	STA	#1
5		BASE	DD
6		RESB	4096
7		RESW	4096
8		RESB	4096
9		ADDR	S,A
10		STA	CC
11		SUB	#3
12		STA	EE
13		MUL	#2
14		STA	FF
15	.		
16		LDA	EE
17		DIV	#2
18		STA	GG
19		MUL	#3
20		STA	#HH
21		LDA	@CC
22		STA	II
23		LDA	FF
24		COMP	HH
25		JGT	IF
26		RESB	4096
27	.		
28	IF	LDA	GG
29		+DIV	#EE
30		STF	II
31	.		
32	CC	RESW	5
33	EOF	BYTE	X'123'
34		ORG	CC
35		RESB	1
36		ORG	
37	EE	RESW	1
38	FF	RESW	1
39	GG	RESW	1
40	HH	RESW	1
41	II	RESW	2
42		END	FIRST

-測試結果：

```
****Relative Error!  
An exception has occurred, use %tb to see the full traceback.
```

SystemExit: Relative Error !

```
C:\Users\User\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2918:  
UserWarning: To exit: use 'exit', 'quit', or Ctrl-D.  
    warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```