



『系統程式』—課程作業(3)

評 分

File Name: E236600-H-AY111A-03.doc

一、注意事項：

- (1) 本作業繳交方式：以上傳電子檔方式繳交，並應配合下列之說明。
  - (A) 請使用本作業電子檔之作答頁依題序完成本作業，於首頁填入個人識別資料後，以[學號]-HW03.doc 之格式命名存檔並繳交。
  - (B) 將完成本作業相關檔案(含本作業檔案)收集於一資料夾內(資料夾之命名方式：[學號]-HW3)，並附加資料夾內容說明檔；將該資料夾壓縮後，上傳至成功大學數位學習平台(Moodle)之指定位置。壓縮檔名之格式為：[學號]-HW02.(rar|zip)，壓縮格式可為 rar 或 zip。
- (2) 切勿更改本作業原始版面設定，並依規定保存所有相關資料與程式等檔案。
- (3) 獨立完成作業，如有抄襲或剽竊情事，除酌情議處並扣減學期成績。
- (4) 評分基準：各題配分×83%，優增劣減。遲交酌減得分，扣減方式另行公佈。

二、基本資料：

繳交期限：2022/12/30

繳交日期：2022/ 12 / 30

班級：電機113

學號：F64096114

姓名：郭家佑

三、作業內容：(Main Theme – Program Relocation and SIC Program Linking)

- (1) Suppose a computer has the word-addressable memory capacity of 256K 60-bit words and the instructions are either 15-bit format-1 or 30-bit format-2. Therefore, one memory word can be composed of 4 format-1 instructions, 2 format-2 instructions, 2 format-1 instruction and 1 format-2 instructions, or 60-bit binary data. Based on the information, please design the format of text records with relocation information of the object program. (20 points)
- (2) 試設計與實作一適用於 SIC/XE 之 Relocation Linking Loader，使用一陣列(Array)模擬 16 KB 記憶體儲存空間，可將例如課程用書 pp. 132 Figure 3.5 與 pp. 137~138 之 Object Program 載入隨機分配之記憶體儲存空間，並將載入結果產生文字(Text)格式之記憶體映像檔(Memory Image File)以供使用 SIC Simulator 匯入並執行。 (80 points)

■ 輸入資料：Object code program for SIC/XE (example. pp. 132 & test code)

■ 輸出資料：ASCII Text File of Memory Image as described below.

Line	Text Content and Format	Note
1	A[char(6)][Hex(6)][Hex(6)][Hex(6)]	Header line: I/Name/Saddr/Size/Taddr
2 ~ m	xxxxxxx ... xxxxxxxx ... xxxxxxxx xxxx ... xxxx ... xxxx ... xxxx ... xxx xxxxxxx ... xxxxxxxx ... xxxxxxxx	Memory bytes in ASCII Hex Digits ① 32 Bytes/Line (64 chars/Line) ② 2 Hex Digits/Byte
(m+1)	xxxxxxx	Last line of memory bytes (<= 64 char)

\* Notes: (A) File name: DEVF2 when imported by SIC simulator

(B) For relocatable object program files, the M record described on page 132 in course textbook is followed.

Course Textbook: "System Software – An Introduction to Systems Programming", by Leland L. Beck 3rd Ed.





請使用本頁起之空間完成作業。(注意事項：請標明題號並依題序完成作業。)

(1) Text record

**Design:**

[0]: T (symbol for T record)

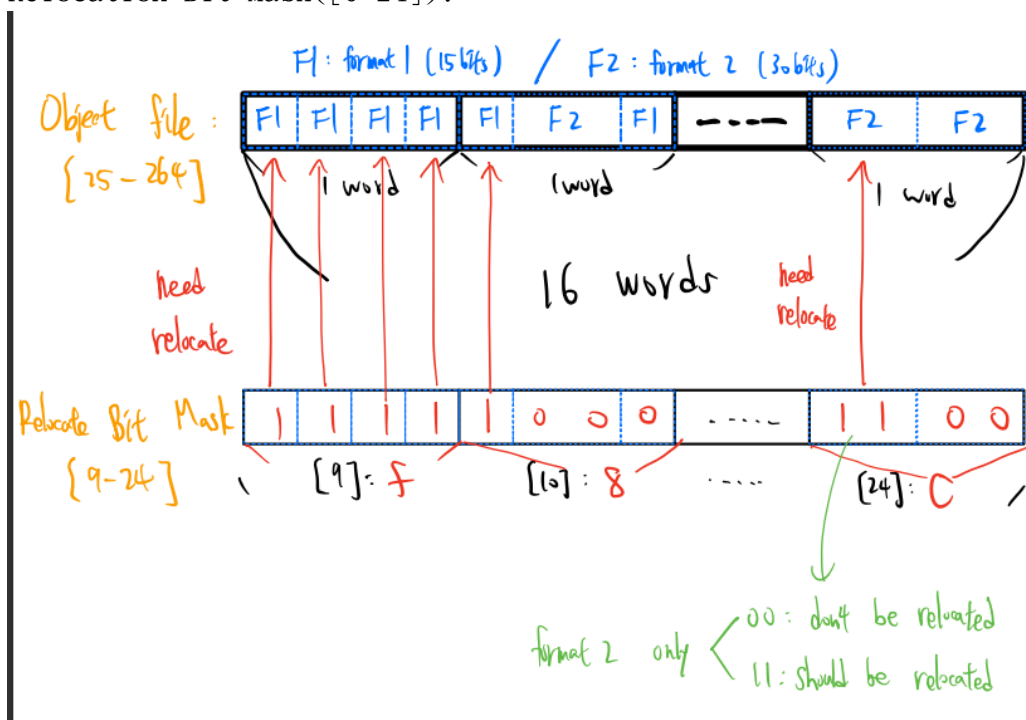
[1~7]: start address of object file (Hex)

[8]: words number of record (Hex), which 1 char hex number can contain max 16 words

[9~24]: Bits Mask of relocation. Because max number of T record object code is 16 words, so it needs at least 16 char of Hex(4 bits can express each format(15bits) of word) to express.

[25~264]: Object code of program. (hex). Because max number of T record object code is 16 words, so it needs  $16 \times 60 / 4 = 240$  char of Hex number to express.

Relocation Bit Mask([9~24]):



**Format:**

Field	Content / Description	Format
Col. 1	T	Char
Col. 2~7	Starting address of object file in this text record	Hex
Col. 8	Number of words in this text record (+1 → 1~16)	Hex
Col. 9~24	Relocation Bit Mask	Hex
col. 25~264	Object code	Hex





Ex.

T^001000^3^F8C0000000000000^4820C900F03628\_03F2FEC0320120\_16A4B1000003E2

說明：

T: record type

001000: object file start from 4096

3: object file have 3 words

F8C0000000000000(=1111\_1000\_1100\_0000...<sub>2</sub>): 第一個 words 的四個 format-1 均要 relocate; 第二個 word 只用第一個 format1 要 relocate, 其餘後面的 1 個 format-2 和 format-1 不用; 第三個 word 只用第一個 format2 要 relocate, 其餘後面的 1 個 format-2 不用。因為 size of word 只有 3, 所以後面全部補 0。

4820C900F03628\_03F2FEC0320120\_16A4B100000: object code

### Advantages and Disadvantages:

優:

1. 利用 Relocation Bit Mask 可以省掉 Linking Loader 還要去計算 Mrecord 的運算, 使 record type 不這麼複雜。
2. 由於 bit mask 是用 hex(4bits)來表示, 可以清楚對應到哪一個 word 的哪一些 format 需要 relocate。

缺:

1. 很明顯看得出整行 record 的長度會高達 265 char, 若是遇到一定要換行 T record 的情形會很浪費空間, 而且 16 個 bit mask 也會有點浪費, 因為不見得每一行 T record 都能寫完 16 個 Word。
2. 若是像第二題作業會有 external reference, 只有 bit mask 是不行的, 還是必須利用 M record 來協助。





## (2)Linking Loader

P. S. 老師給的測資有的 symbol 沒有名子沒有用 space 補齊 6bits(ex\COPY 需改成 →COPY ; RDREC→RDREC)，會導致我的程式把有多加 space 和沒有 space 當成兩個不同的 symbol，造成錯誤，所以我有附上我多加 space 到 6bitss 的測資以供助教測試。

## 操作說明(在 WIN11 環境下操作)

檔案：

1. 測資(in “test” folder):共 7 筆(可輸入總共 5 種排列組合)
2. source code: sLink.cpp
3. execute file: sLink.exe
4. 輸入指令範例: KeyIn.txt(可輸入總共 5 種排列組合)
5. 測資所產生的 image file(in “image” folder):共 5 筆，分別為 7 筆測資所產生的 DEVF2，為了區別各個檔案所產生的 image，把他更改為對應的名稱，若要跑 Simulator 必須再改名回 DEVF2。
6. 中間檔:input\_merge.txt 為若是將 control section 拆成多個 file 輸入時，先將這些 file 合併在一起在拿去產生 image file。

執行 sLink.exe 後須輸入的指令:(可參考 KeyIn.txt 的範例輸入)

1. 先輸入總共有多少 input file 要輸入
2. 接著輸入檔案名稱(主程式必須放第一個順位)

```
C:\Users\oppo1\source\repos  X  +  v
Please input the number of your Control sections! (include num of main program)
1
Please input the file name! (the main program should be the first input)
OBJFILE-SP-SIC_XE-Copy(abs)-all
```

3. 輸入完後按 enter 就會產出 DEVF2 和中間檔 input\_merge.txt
4. 接著就可以拿 DEVF2 去 simulator 跑了

## 實作說明

- Algorithm and program flow with description  
我的 LinkingLoader 主要可以分成 pass1 和 pass2 兩部分。Pass1 主是是先  
去讀檔案，看若有多個 input file，就需要把多個檔案合併成一份再來處  
理。 Ex、鍵入 3  
OBJFILE-SP-SIC\_XE-Copy(abs)-Main OBJFILE-SP-SIC\_XE-Copy(p90)-RDREC  
OBJFILE-SP-SIC\_XE-Copy(p90)-WRREC  
會將三個檔案和成一份 input\_merge.txt 再去進行 linking 和 loading 動  
作。



```
input_merge.txt - 記事本
檔案 編輯 檢視

HCOPY 001010001033
DBUFFER001043BUFEND002043LENGTH00103D
RDREC WRREC
T0010101D1720274B1000000320232900003320074B1000003F2FEC0320160F2016
T00102D0D0100030F200A4B1000003E2000
T00104003454F46
M00101405+RDREC
M00102105+WRREC
M00103405+WRREC
E001010
HRDREC 000000000002B
RBUFFERLENGTHBUFEND
T00000001DB410B400B44077201FE3201B332FFADB2015A00433200957900000B850
T00001D0E3B2FE9131000004F0000F1000000
M00001805+BUFFER
M00002105+LENGTH
M00002806+BUFEND
M00002806-BUFFER
E
HWRREC 000000000001C
RLENGTHBUFFER
T00000001CB41077100000E32012332FFA53900000DF2008B8503B2FEE4F000005
M00000305+LENGTH
M00000D05+BUFFER
E
```

接著是進行 Pass1 後半部讀 H, D, R record 的動作。

1. H record 會將 name、start address、length 利用 substr 和字串整數轉換處理存回對應的變數中，而若是 start address 是 000000 就會隨機產生一個 srand 的數(reloca\_offset)給加上。
2. D record 會將每一個 symbol 和其對應的數值(須加上該 section 的起始位置)存回 ESCTAB(利用一個 Vector<pair<string, int>>)裡面。

```
ESCTAB is below!
COPY      : 4112
BUFFER    : 4163
BUFEND    : 8259
LENGTH    : 4157
RDREC     : 8259
WRREC     : 8302
```

3. R record 基本上沒用到，可以拿來 debug 用

Pass2 主要是先處理剩下的 T、M、E record。

1. T record 將所讀到的 object code 填回去要輸出的容器裡，只是要注意是相對或是絕對位置。
2. M record 就是將起始位置加上該 section 的絕對位置再加上該 M record 起始位置的相對位置的值，轉為 half byte 後去該對應的 output 容器拿值(通常是 000000)過來與 ESCTAB 的 symbol 相加，再將他存回去。
3. E record 只是用來檢查用的，看跟起始位置有沒有一樣。

最後 output file 先將 I record 的各個要的數值計算後轉為 Hex string 輸出，並從第二行後將 output 容器裡的值通通輸出到 DEVF2 裡面。



- How to build & run your executable program

執行 sLink.exe 後須輸入的指令:(可參考 KeyIn.txt 的範例輸入)

1. 先輸入總共有多少 input file 要輸入
2. 接著輸入檔案名稱(主程式必須放第一個順位)

```
C:\Users\oppo1\source\repos X + v
Please input the number of your Control sections! (include num of main program)
1
Please input the file name! (the main program should be the first input)
OBJFILE-SP-SIC_XE-Copy(abs)-all
```

3. 輸入完後按 enter 就會產出 DEVF2 和中間檔 input\_merge.txt
4. 接著就可以拿 DEVF2 去 simulator 跑了

## 結果說明

- Screen shots of Linking & loading to produce image file

```
Microsoft Visual Studio 偵錯器 X + v
Please input the number of your Control sections! (include num of main program)
3
Please input the file name! (the main program should be the first input)
OBJFILE-SP-SIC_XE-Copy(abs)-Main OBJFILE-SP-SIC_XE-Copy(p90)-RDREC OBJFILE-SP-SIC_XE-Copy(p90)-WRREC

ESCTAB is below!
COPY      : 4112
BUFFER    : 4163
BUFEND    : 8259
LENGTH   : 4157
RDREC     : 8259
WRREC     : 8302

C:\Users\oppo1\source\repos\system program\Debug\system program.exe (處理序 4732) 已結束，出現代碼 0。
若要在偵錯停止時自動關閉主控台，請啟用【工具】->【選項】->【偵錯】->【偵錯停止時，自動關閉主控台】。
按任意鍵關閉此視窗...
```

執行完會 show 出 ESCTAB 內容並產生中間檔與 DEVF2。

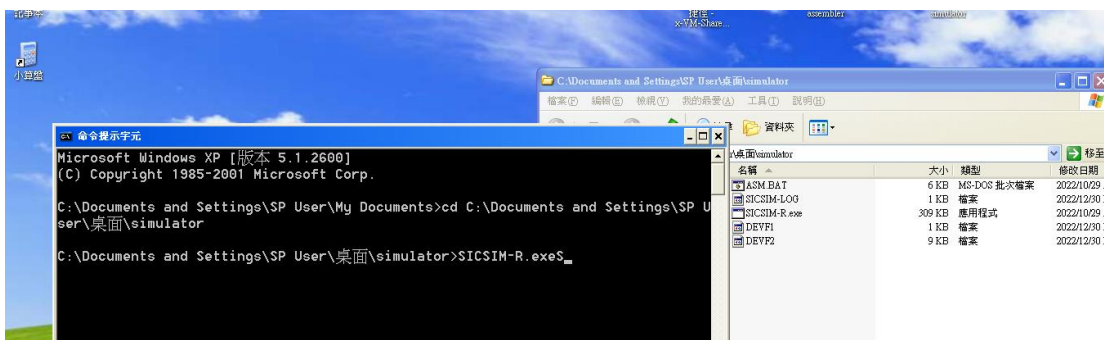
DEVF2	2022/12/30 下午 10:28	檔案	9 KB
input_merge.txt	2022/12/30 下午 10:28	文字文件	1 KB



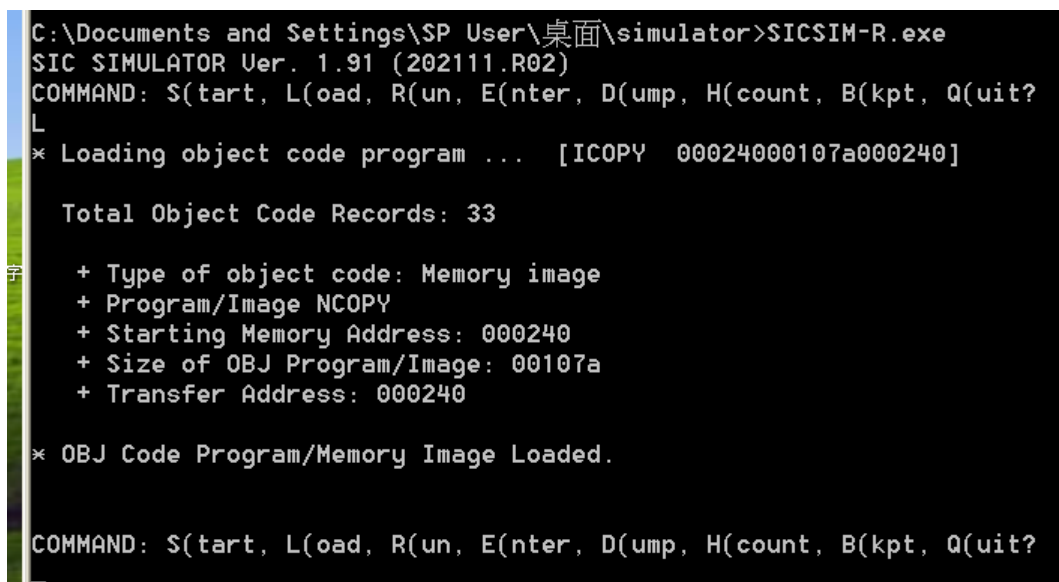




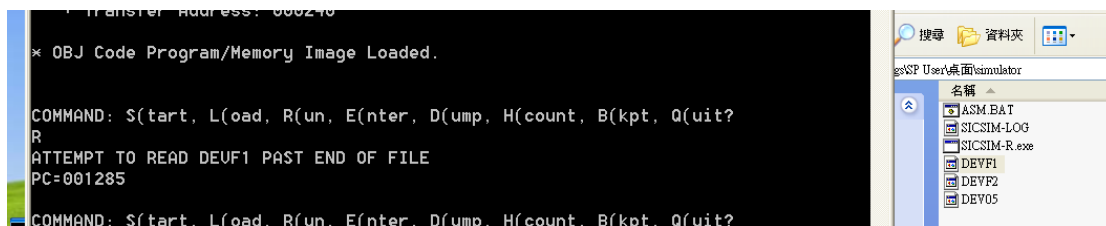
- Screen shots of Loading & running with SIC simulator



在 SIC simulator 資料夾裡面放入剛剛 sLink 產出的 DEVF2，以及想被複製的 input 測資 DEVF1，並用 CMD 執行 SICSIM-R.exe。



先 load 進 image file。



接著按 R 就會產生出 copy 後的檔案 DEV05。





DEVF1:

```
1 System program TA is very handsome/pretty.
2 System program TA is very nice.
3 System program teacher is nice.
4 I have learned a lot from System program.
5 Thanks for teaching.
```

DEV05:

```
1 System program TA is very handsome/pretty. System program TA is very nice. System program teacher is nice. I have learned a lot from System program. Thanks for teaching.
```