Software Project

SVR4 POSIX → system V release 4
UNIX system call

- **Design and implement an assembly for SIC/XE**

The test data look like Figure 2.5. and are generated by MS Notepad.

The possible OPCODE in the test data contains all the OPCODEs which appear in Figure 2.5. They include STL, LDB, LDA, COMP,…

The assembly directives include all those appear in Figure 2.5., e.g., START, +, BASE, BYTE, RESW, RESB, END, ….

The name of labels are not allowed to be the same as OPCODEs or assembly directives.

The format of the object program generated by your assembly should conform to Figure 2.8.

Software Project

- **Bonus points**
  - If your assembly is a one-pass assembly.
  - If the implementation of your assembly includes “Literals”
  - If the implementation of your assembly includes “Symbol-defining Statements”
  - If the implementation of your assembly includes “Program Blocks”
  - If the implementation of your assembly includes “Control Sections”

Project Report

- **The student should prepare a report which contains at least the follows:**
  - The architecture of the implemented assembler
  - What you have learned and experienced during the implementation.
    - E.g. You could show your daily record of the implementation.
  - In case you implement more than the required specification, please itemize it.
    - If you implement something mentioned in the previous slice (bonus points), show your test codes (in SIC/XE), and the generated object programs.
  - Copyright Claim
    - Do you make the implementation yourself?
  - Any thing you would like to let G.H.Hwang know.
    - E.g. Suggestion, …
- **Who will be reading the report?**
  - Not TAs but G. H. Hwang

How to hand in your report?

- **Please deliver your project report in Moodle system**
  - Attached filename: your_student_id.zip
  - It should have at least the following items:
    - Electronic files of your report
      - MS word and PDF
    - Source codes
      - OS, Used language, and how to compile your code

if(ins[i]) print

① 先成 SYMTAB
if (RESW || RESB || BYTE) { <3 byte +3
>3 byte +__byte
else 從起始位置一直+3 (16進位)

每個指令對應該 opcode → 也成16進位算 Loc
if (var[i] 有 #) {
else if (ins[i] 有 +)

else if (disp = loc[i+1] - Loc[var[i]]

(string)loc[i] 轉成 10進位
+3後轉回 16進位 (string)

var_loc → 要找的變數位置
code → object code (2進位)
result → 2進位轉16進位的結果

Line	Loc	Source statement			Object code
5	0000	COPY	START	0	
10	0000	FIRST	STL	RETADR	17202D
12	0003		LDB	#LENGTH	69202D
13			BASE	LENGTH	
15	0006	CLOOP	+JSUB	RDREC	4B101036
20	000A		LDA	LENGTH	032026
25	000D		COMP	#0	290000
30	0010		JEQ	ENDFIL	332007
35	0013		+JSUB	WRREC	4B10105D
40	0017		J	CLOOP	3F2FEC
45	001A	ENDFIL	LDA	EOF	032010
50	001D		STA	BUFFER	0F2016
55	0020		LDA	#3	010003
60	0023		STA	LENGTH	0F200D
65	0026		+JSUB	WRREC	4B10105D
70	002A		J	@RETADR	3E2003
80	002D	EOF	BYTE	C'EOF'	454F46
95	0030	RETADR	RESW	1	
100	0033	LENGTH	RESW	1	
105	0036	BUFFER	RESB	4096	
110			.		
115			.	SUBROUTINE TO READ RECORD INTO BUFFER	
120			.		
125	1036	RDREC	CLEAR	X	B410
130	1038		CLEAR	A	B400
132	103A		CLEAR	S	B440
133	103C		+LDT	#4096	75101000
135	1040	RLOOP	TD	INPUT	E32019
140	1043		JEQ	RLOOP	332FFA
145	1046		RD	INPUT	DB2013
150	1049		COMPR	A,S	A004
155	104B		JEQ	EXIT	332008
160	104E		STCH	BUFFER,X	57C003
165	1051		TIXR	T	B850
170	1053		JLT	RLOOP	3B2FEA
175	1056	EXIT	STX	LENGTH	134000
180	1059		RSUB		4F0000
185	105C	INPUT	BYTE	X'F1'	F1
195			.		
200			.	SUBROUTINE TO WRITE RECORD FROM BUFFER	
205			.		
210	105D	WRREC	CLEAR	X	B410
212	105F		LDT	LENGTH	774000
215	1062	WLOOP	TD	OUTPUT	E32011
220	1065		JEQ	WLOOP	332FFA
225	1068		LDCH	BUFFER,X	53C003
230	106B		WD	OUTPUT	DF2008
235	106E		TIXR	T	B850
240	1070		JLT	WLOOP	3B2FEF
245	1073		RSUB		4F0000
250	1076	OUTPUT	BYTE	X'05'	05
255			END	FIRST	

Figure 2.6 Program from Fig. 2.5 with object code.

```
HCOPY  000000001077
T000000 1D 17202D 69202D 4B101036 032026 290000 332007 4B10105D 3F2FEC 032010
T00001D 13 0F2016 010003 0F200D 4B10105D 3E2003 454F46
T001036 1D B410 B400 B440 75101000 E32019 332FFA DB2013 A004 332008 57C003 B850
T001053 1D 3B2FEA 134000 4F0000 F1 B410 774000 E32011 332FFA 53C003 DF2008 B850
T001070 07 3B2FEF 4F000005
M00000705
M00001405
M00002705
E000000
```

Figure 2.8 Object program corresponding to Fig. 2.6.

每個指令的 object code

用 map 建 opTAB (2進, 刪2位)

↓

if (ins[i] = CLEAR)

if (type[i] = BYTE)

↓

if (immediate addressing)

if (format 4)

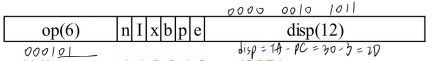
if (indirect addressing)

if (index addressing)

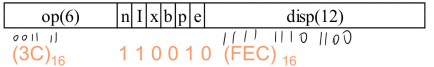
if (pc-relative)

else base-relative

- **PC-relative** (simple address)
  - 10 0000 FIRST STL RETADR 17202D



  - displacement= RETADR - PC = 30-3 = 2D
  - 40 0017 J CLOOP 3F2FEC



  - displacement= CLOOP-PC= 6 - 1A= -14= FEC

創1陣列存 object code

opTAB 轉二進放前6碼

nixbpe 照 if/else 填

disp = 變數位置 - pc

寫1function 把陣列每4個數換成16進位丟入 string[i]

去寫T record 的地方決定怎麼換行