

系統程式第一次程式作業

(Cross Assembler 實作)

班級：資訊三甲

學號：10927109

姓名：陳宥蓁

- 開發平台

Windows11

- 使用開發環境

Dev-C++

- 所選擇的組合語言(x86, SIC)

SIC 組合語言

- 說明你的程式設計(功能，流程，使用的 data structure)

功能：將組合語言翻譯成機器碼

流程：先讀組合語言檔 → **pass1 開始** → 將每一道指令切成

token → 判斷 token 是屬於哪一個 table(如果是前 4 個，就找出在 table 中的位置記錄下來，如果是後 4 個(symbol、integer、string、literal)就將它們利用 hash 的方式放入 table 中並將位置記錄下來，此外要特別注意放 symbol 時如果 symbol table 中已經存在此 symbol 那就是 **pass1**

error(duplicate definition) → 計算每一道指令的起始位置與指令長度並同時判斷指令是否有符合指令格式，沒有的話要印 error → 每一道指令皆處理完上述步驟後 → 定義

forward reference 位置 → **pass2 開始** → 將相對應的指令翻譯成機器碼，在翻譯的同時會檢查指令使用到的

symbol 是否有被定義，如果沒有就是 **pass2 error(undefine symbol)** → 重複上述步驟直到翻完所有指令 → 寫檔 →

程式結束!!!

使用的 data structure

(1) ConstantTable

前 4 個 table 定義相同型別。

```
class ConstantTable {  
public:  
    ConstantTable() {  
        table.clear() ;  
        table.push_back("") ;  
    } // constructor  
  
    vector<string> table ;  
  
    void insert(string temp) { //加入表中  
        table.push_back(temp) ;  
    } // insert()  
};  
  
ConstantTable instruction ;  
ConstantTable pseudo ;  
ConstantTable reg ;  
ConstantTable delimiter ;
```

(2) Token

存放每條指令的 token、token 屬於哪個 table 與在
table 的位置、指令的起始位置與指令長度。

```
struct Token{  
    vector <int> tokenType ;  
    vector <int> tokenValue ;  
    vector <string> token ;  
    int startAddress = 0 ;  
    string start = "" ;  
    int length = 0 ;  
};  
  
vector <Token> tokenArray ;
```

(3) MachineCode

存放每道指令的行數、位置、指令、機器碼

```
struct MachineCode{
    int line ;
    string loc ;
    string sourceStatement ;
    string objectCode ;
};

vector <MachineCode> machineCode ;
```

(4) 後面 4 個 table(symbol、integer、string、literal) 型別

```
class SymbolTable { // 5
class IntegerTable : public Hash { // 6
class StringTable : public Hash { // 7
class LiteralTable { // 8
```

```
SymbolTable symbol ;
IntegerTable integer ;
StringTable str ;
LiteralTable literal ;
```

● 未完成的功能

- i. 尚未處理如果在 format3 計算 disp 時使用 PC 與 B 計算出的值皆會溢位，代表指令是錯的，要印 error 且重新計算此指令以下指令的位置。
- ii. 尚未處理從出現 START 這行指令後才開始翻成機器碼的問題。
- iii. 尚未處理沒有 END 指令的問題

FIRST	STL	RETADR
TEST	START	1000
CLOOP	JSUB	RDREC
	LDA	LENGTH
	COMP	ONE
	JEQ	ENDFIL
	JSUB	WRREC
	J	CLOOP
ENDFIL	LDA	EOF
	STA	BUFFER
	LDA	THREE
	STA	LENGTH
	JSUB	WRREC
	LDL	RETADR
	RSUB	
EOF	BYTE	C'EOF'
THREE	WORD	3
ZERO	WORD	0