10927153 資工三甲 吳上玲

SIC + SIC\_XE 程式碼說明文件

1. 開發平台

Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.21 GHz

使用開發環境

Dev C++

1. 所選擇的組合語言( x86, SIC )

SIC

1. 說明你的程式設計(功能，流程，使用的data structure)

\*因為SIC和SICXE精研靜候發現有許多兼容的地方，因此我將它們寫在同一支程式中，利用選單來分別執行兩種功能。

\*我的資料結構使用vector<>存取，包括table1~9 (其中table8為存取指令格式及op而設)、literal，及我將要切的token資訊(程式過程中更方便調整token結構)，因此vector結構struct我設置了兩種分別使用。

\*我的想法是:

程式結構:

Class gettoken {

prepare\_table()

// 一開始執行初始化所有表格，及讀入各種Table，給hash table100的空間。

Check( token字串, 行數)

// 遍歷所有tabel找token性質及在tabel中的位置，若無則使用hashing新增至hash table .

Syntax(現在地址, token指針(指向本行第一個token), 選單指令, 是否為EQU)

// 首先判斷此行語法是否正確(SIC/SICXE), 再由已知資訊作位址的運算，遇到本行第一個token為pseudo，判斷

duplicate error .

Gettoken (選單指令) {

讀一行行 >> 一個個check() >> 一行讀完syntax() >> 直到讀到END >> 結束.

} // Gettoken

} ;

Class SIC : public gettoken {

Assembler() // 翻Mcode + 印出 + 寫檔

} // SIC

Class SICXE : public gettoken {

Assembler() // 翻Mcode + 印出+ 寫檔

} // SICXE

Int main() {

呼叫prepare\_table()

及執行gettoken() 再分別使用SIC/SICXE assembler() 依照格式翻譯Mcode

} // main

pass1 = 讀入一行直到END( 切token + 判斷lexical及組態 + 在放入最後一個token時(辨認語法+duplicated error+給予下一行位址 ) ) .

pass2 = 依據格式翻譯Mcode及判斷undefined error .

\*有處理所有指令格式、literal、EQU、所有pseudo 包括base、 ”@”, “#”, ”=” 四種定址，syntax error。Undifined error, duplicate error

1. 未完成的功能

EQU若是後面為四則運算無法判斷，其實只要在翻MCODE時一一辨認delimeter，再換算的話就能處理這個特例，但是時間來不及