Abdelrahman Ahmed Nabil
221000309

SIC-XE Assembler

Supervised by : Dr/Noha Seddik & Eng/Hoda Osama

OVERVIEW

The project consists of a GUI that allows the user to pick a SIC-XE assembly file from their computer, and the assembler does the relevant functions:

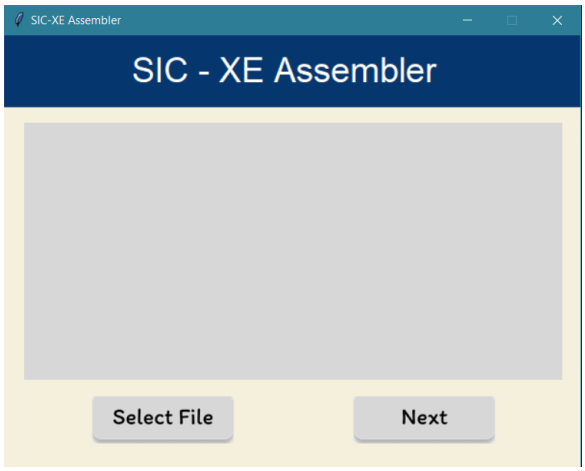
1- Pass 1 for generating locations , symbols and literals

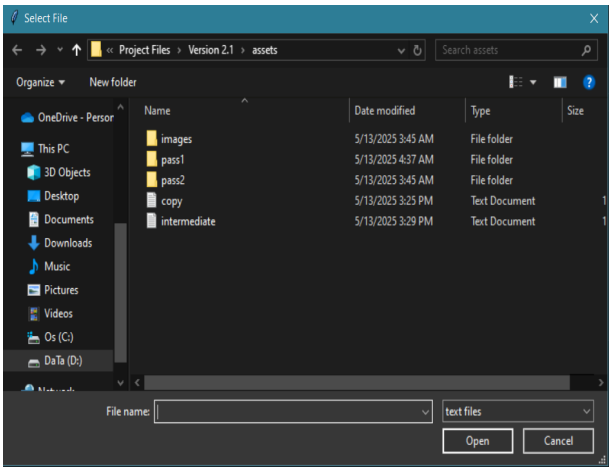
2- Pass 2 for generating object code

3- Generating HTME records

HOW TO USE

First open the “gui.py” file , and press run .

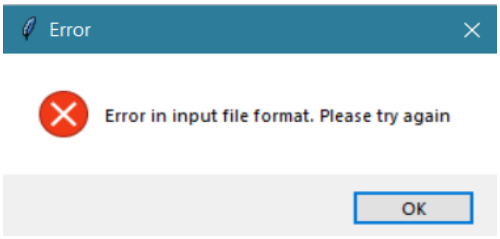




The GUI window should appear as the picture on the left.

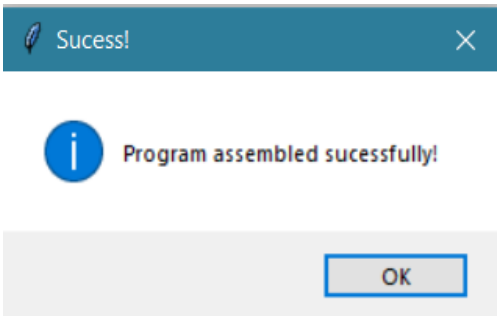
Next, click on select file to choose the assembly file. A file dialog will appear as in the second picture on the right

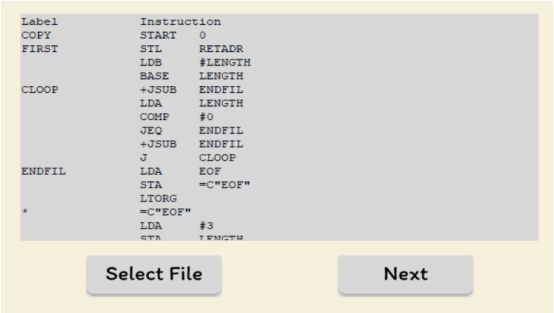
The program will *only* work if the format of the file is such that it includes the columns :Line Number , Label and Instruction , each separated by 2 tabs. The file is checked to see if it matches said format. If not , an error message box is displayed upon clicking “Next”



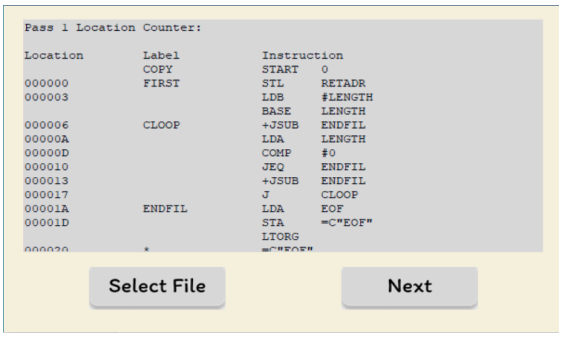
If it is indeed in the correct format , the program will proceed. If the assembly process was a success , this info box is displayed.

However , if any errors do occur , an error message box just like the one previous will display with said error , such as : “Instruction doesn’t exist error”





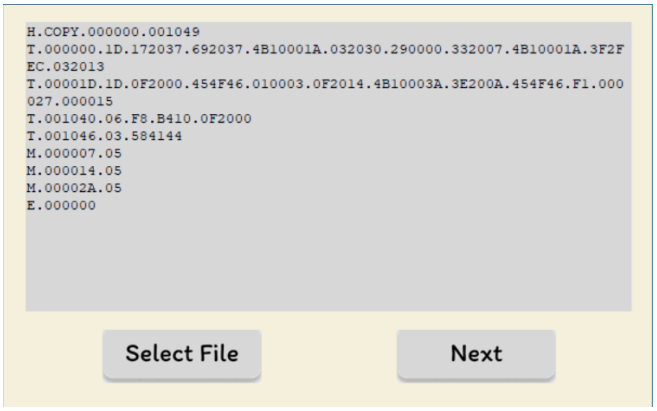
After closing the info box, the text field in the GUI displays the intermediate file , with line numbers and comments removed.



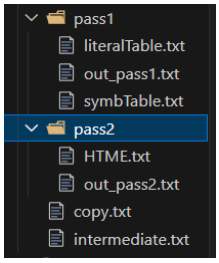
Pressing Next again will display the location counter , symbol table and literal table. You can scroll down to see each in full

The next step will display the object code file , and the final step will be the HTME record file.

Location	Label	Instruction		Object Code
	COPY	START	0	
000000	FIRST	STL	RETADR	172037
000003		LDB	#LENGTH	692037
		BASE	LENGTH	
000006	CLOOP	+JSUB	ENDFIL	4B10001A
00000A		LDA	LENGTH	032030
00000D		COMP	#0	290000
000010		JEQ	ENDFIL	332007
000013		+JSUB	ENDFIL	4B10001A
000017		J	CLOOP	3F2FEC
00001A	ENDFIL	LDA	EOF	032013
00001D		STA	=C"EOF"	0F2000
		LTORG		



The respective files are also saved on the device according to the directive defined in the "constants.py" file in the "DEFAULT_PATHS" dictionary.



Alternatively , you can run without GUI by using the "main.py" file, and then you can open the respective files to check the output

MAIN CLASSES

The program consists of multiple modules that work together to achieve the assembling process

Assembler

It is considered the entry point of the program . It uses the Pass1 and Pass2 modules to execute the assembling process

Pass1

This module contains functions that concern pass 1 , which includes reading the input file and assigning each line to its relevant location counter while also keeping track of symbols and literals in respective dictionaries, to be later written to their respective files

Pass2

This module contains functions that involve generating the object code of each instruction and assigning each object code to its relevant location , and finally generating the HTME records. The module makes use of a crucial submodule called **InstructionDecoder** for decoding each instruction according to its format

Utils

Contains several useful functions used by multiple modules.

It contains checks for format and intermediate file generation , as well as hexadecimal/denary conversions for address calculation and formatting according to the relevant instruction type

Constants

Contains all important constants needed to be used by the program, such as:

- I/O file paths in the FILE_PATHS dictionary
- Dictionary for instructions of each instruction type (Format 1 ,2 ,3)
- Dictionary for directives
- Dictionary for opcodes of instructions
- Dictionary for saved symbols
- Dictionary for saved literals