******COMSATS University Islamabad (Lahore** **Campus)**

**LabTask#1– Fall2023**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Course Title: | Operating Systems | Course Code: | CSC322 | Credit Hours: | 3(2,1) |
| Course Instructor: | Nadeem Ghafoor Chaudhry | Programme Name: | BSE | | |
| **Due Date:** | **Thursday, September 28th 2023** | **Maximum Marks:** | | **20** | |
| **Important Instructions / Guidelines:** | | | | | |

Question No 1. CLO:5,6; Bloom Taxonomy Level: <Applying>

In most of Deitel & Deitel books on programming an exercise called Simpletron is given (Java How to Program 11th edition exercise 7.36, C How to Program exercise at end of chapter 7). In this exercise a very simple computer called Simpletron is described. Your job is to develop Simpletron in C. Your program should take the program written in SML as a command line argument, load it in Simpletron’s memory automatically and start executing it.

You will split your c code into two files. One called simple.c, it will contain the main function. The other, called util.c will contain utility functions like reading SML program from file. Functions of util.c will be put in a static library called libsimple.a Symbolic constants like LOAD, STORE will go in a header file called simple.h. You will use make utility to compile the code. The name of the executable file should be simple.exe

Your program will be different from the statement given in the book in one respect and that is the loading of the SML program. Instead of prompting the user to enter the entire program from the keyboard, you will provide a file which will contain the SML program to be executed.

A sample run of the program is as follows:

**$ ./simple.exe sml.txt**

Accumulator: 0

Instruction Counter: 0

Instruction Register: 0

Opcode: 0

Operand: 0

0 1 2 3 4 5 6 7 8 9

0 1007 1008 2007 3008 2109 1109 4300 0 0 0

1 0 0 0 0 0 0 0 0 0 0

2 0 0 0 0 0 0 0 0 0 0

3 0 0 0 0 0 0 0 0 0 0

4 0 0 0 0 0 0 0 0 0 0

5 0 0 0 0 0 0 0 0 0 0

6 0 0 0 0 0 0 0 0 0 0

7 0 0 0 0 0 0 0 0 0 0

8 0 0 0 0 0 0 0 0 0 0

9 0 0 0 0 0 0 0 0 0 0

Executing instruction: 1007

Enter a number:

20

----------------------------------------------------------------------------------

Accumulator: 0

Instruction Counter: 1

Instruction Register: 1007

Opcode: 10

Operand: 7

0 1 2 3 4 5 6 7 8 9

0 1007 1008 2007 3008 2109 1109 4300 20 0 0

1 0 0 0 0 0 0 0 0 0 0

2 0 0 0 0 0 0 0 0 0 0

3 0 0 0 0 0 0 0 0 0 0

4 0 0 0 0 0 0 0 0 0 0

5 0 0 0 0 0 0 0 0 0 0

6 0 0 0 0 0 0 0 0 0 0

7 0 0 0 0 0 0 0 0 0 0

8 0 0 0 0 0 0 0 0 0 0

9 0 0 0 0 0 0 0 0 0 0

Executing instruction: 1008

Enter a number:

30

---------------------------------------------------------------------------------

Accumulator: 0

Instruction Counter: 2

Instruction Register: 1008

Opcode: 10

Operand: 8

0 1 2 3 4 5 6 7 8 9

0 1007 1008 2007 3008 2109 1109 4300 20 30 0

1 0 0 0 0 0 0 0 0 0 0

2 0 0 0 0 0 0 0 0 0 0

3 0 0 0 0 0 0 0 0 0 0

4 0 0 0 0 0 0 0 0 0 0

5 0 0 0 0 0 0 0 0 0 0

6 0 0 0 0 0 0 0 0 0 0

7 0 0 0 0 0 0 0 0 0 0

8 0 0 0 0 0 0 0 0 0 0

9 0 0 0 0 0 0 0 0 0 0

Executing instruction: 2007

------------------------------------------------------------------------------------

Accumulator: 20

Instruction Counter: 3

Instruction Register: 2007

Opcode: 20

Operand: 7

0 1 2 3 4 5 6 7 8 9

0 1007 1008 2007 3008 2109 1109 4300 20 30 0

1 0 0 0 0 0 0 0 0 0 0

2 0 0 0 0 0 0 0 0 0 0

3 0 0 0 0 0 0 0 0 0 0

4 0 0 0 0 0 0 0 0 0 0

5 0 0 0 0 0 0 0 0 0 0

6 0 0 0 0 0 0 0 0 0 0

7 0 0 0 0 0 0 0 0 0 0

8 0 0 0 0 0 0 0 0 0 0

9 0 0 0 0 0 0 0 0 0 0

Executing instruction: 3008

-----------------------------------------------------------------------------------

Accumulator: 50

Instruction Counter: 4

Instruction Register: 3008

Opcode: 30

Operand: 8

0 1 2 3 4 5 6 7 8 9

0 1007 1008 2007 3008 2109 1109 4300 20 30 0

1 0 0 0 0 0 0 0 0 0 0

2 0 0 0 0 0 0 0 0 0 0

3 0 0 0 0 0 0 0 0 0 0

4 0 0 0 0 0 0 0 0 0 0

5 0 0 0 0 0 0 0 0 0 0

6 0 0 0 0 0 0 0 0 0 0

7 0 0 0 0 0 0 0 0 0 0

8 0 0 0 0 0 0 0 0 0 0

9 0 0 0 0 0 0 0 0 0 0

Executing instruction: 2109

------------------------------------------------------------------------------------

Accumulator: 50

Instruction Counter: 5

Instruction Register: 2109

Opcode: 21

Operand: 9

0 1 2 3 4 5 6 7 8 9

0 1007 1008 2007 3008 2109 1109 4300 20 30 50

1 0 0 0 0 0 0 0 0 0 0

2 0 0 0 0 0 0 0 0 0 0

3 0 0 0 0 0 0 0 0 0 0

4 0 0 0 0 0 0 0 0 0 0

5 0 0 0 0 0 0 0 0 0 0

6 0 0 0 0 0 0 0 0 0 0

7 0 0 0 0 0 0 0 0 0 0

8 0 0 0 0 0 0 0 0 0 0

9 0 0 0 0 0 0 0 0 0 0

Executing instruction: 1109

Result: 50

------------------------------------------------------------------------------------

Accumulator: 50

Instruction Counter: 6

Instruction Register: 1109

Opcode: 11

Operand: 9

0 1 2 3 4 5 6 7 8 9

0 1007 1008 2007 3008 2109 1109 4300 20 30 50

1 0 0 0 0 0 0 0 0 0 0

2 0 0 0 0 0 0 0 0 0 0

3 0 0 0 0 0 0 0 0 0 0

4 0 0 0 0 0 0 0 0 0 0

5 0 0 0 0 0 0 0 0 0 0

6 0 0 0 0 0 0 0 0 0 0

7 0 0 0 0 0 0 0 0 0 0

8 0 0 0 0 0 0 0 0 0 0

9 0 0 0 0 0 0 0 0 0 0

Executing instruction: 4300

**Note on submission :** Put all your files (.c .h and makefile) in a tar file. The name of your tar file should be **YOUR\_SID.tar** e.g **FA21\_BSE\_000.tar** (note that I have used **\_** not **-** ) To be sure that all the required files are in your .tar file untar it before submitting it. Your program should compile without errors. I will only issue the make command and your makefile should build simple.exe **Make sure that you follow the instructions about names exactly as stated, else I will not check your program and you will automatically get 0.**