

Programming Assignment #2

Assignment Description

(100 points = 70% of demonstration + 30% of report)

Write a simple SIC simulator that reads a standard SIC object program into memory.

Basic Functionalities

(70 points for demonstrating the functionality)

The simulator takes two commands:

- (a) **load *ObjectFileName***: loads a simple SIC simulator that reads a standard SIC object program into assigned location in memory.
- (b) **show**: shows the contents in the memory location for the loaded object program.

Advanced Functionalities

(10 more points for demonstrating the functionality)

In addition to basic functionalities, the simulator takes two more commands:

- (c) **unload**: releases the data structure for the loaded object program.
- (d) **exit**: exits the simulator successfully.

Moreover, the **load** command will show an error message if there is already an object program loaded in memory.

Full Functionalities

(20 more points for demonstrating the functionality)

In addition to advanced functionalities, the simulator takes one more command:

- (e) **run**: simulates the execution of the loaded object program, and shows the contents of all registers when it ends.

Goals

1. Get familiar with C programming language.
2. Learn to use the I/O facilities and library functions provided by standard C.
3. Get experience with the system-level programming.
4. Get experience with separate compilation, make utility, and C debugger.

Guideline:

1. You have to demonstrate your program in person.
2. You will get 15% bonus if you succeed in demonstrating your program on December 23rd. Official due for demonstrating program is December 30th. After that, 15% penalty will be given for lateness. More precisely, if you get X in demonstration, and Y for the report:
 - (12/23) In-class demonstration = $X * 70\% * 115\% + Y * 30\%$
 - (12/30) Your score = $X * 70\% + Y * 30\%$
 - Late = $(X * 70\% + Y * 30\%) * 85\%$
3. Your report has to include the following elements:
 - 1) A cover page.
 - 2) The problem description.
 - 3) Highlight of the way you write the program.
 - 4) The program listing.
 - 5) Test run results.
 - 6) Discussion.

Programming Assignment #2 Addendum

Example Object Programs

1)

HCOPY 00100000107A

T0010001E1410334820390010362810303010154820613C100300102A0C103900102D

T00101E150C10364820610810334C0000454F46000003000000

T0020391E041030001030E0205D30203FD8205D2810303020575490392C205E38203F

T0020571C1010364C0000F1001000041030E02079302064509039DC20792C1036

T002073073820644C000005

E001000

2)

MATH 00200000002D

T0020001E00201E1820241C201B0C20270020211820241C201B0C202A4C0000000001

T00201E09000005000007000003

E002000

Format of Output

(Similar to the following figure)

Memory address	Contents			
0000	xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx
0010	xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx
⋮	⋮	⋮	⋮	⋮
0FF0	xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx
1000	14103348	20390010	36281030	30101548
1010	20613C10	0300102A	0C103900	102D0C10
1020	36482061	0810334C	0000454F	46000003
1030	000000xx	xxxxxxxx	xxxxxxxx	xxxxxxxx
⋮	⋮	⋮	⋮	⋮
2030	xxxxxxxx	xxxxxxxx	xx041030	001030E0
2040	205D3020	3FD8205D	28103030	20575490
2050	392C205E	38203F10	10364C00	00F10010
2060	00041030	E0207930	20645090	39DC2079
2070	2C103638	20644C00	0005xxxx	xxxxxxxx
2080	xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx

When demonstrating

1. Enter your program and a prompt.
2. Load test1.obj 1000
3. Show 1000
4. Run 1000
5. Show 1000
6. Unload
7. Load test2.obj 2000
8. Show 2000
9. Run 2000
10. Show 2000
11. Exit