

系統程式期末報告

SIC simulator

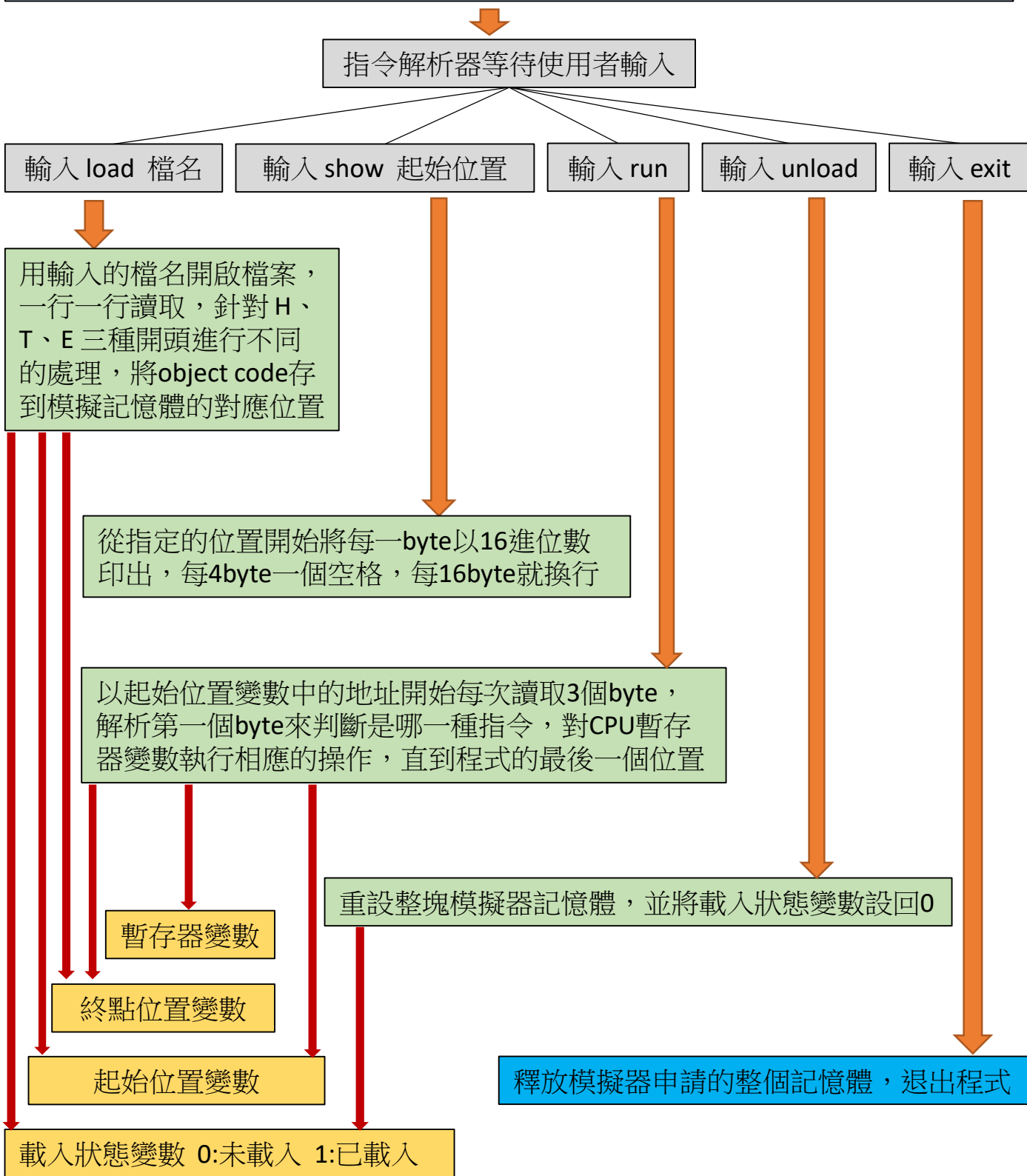
李奕承 611121212

目標概述

這個作業的目標是要做出一個模擬器，他可以讀取object code並載入到記憶體中，還可以dump記憶體中的內容，以及模擬執行object code。

運作原理

剛開始執行時，先申請一大塊記憶體來模擬一台電腦
建立一個結構來代表CPU，成員是 SIC 的五個暫存器(用 24bit 的無號整數來模擬)



測試與結果

開始執行，並申請模擬器需要的空間

```
Lyciih@DESKTOP-CR5NUFU MINGW64 ~/Desktop/github/System-Programming/SIC simulator (main)
$ make run
./sic_simulator 3000
system memory from 0x0 ~ 0x3000
>>> █
```

載入test1.obj

```
Lyciih@DESKTOP-CR5NUFU MINGW64 ~/Desktop/github/System-Programming/SIC simulator (main)
$ make run
./sic_simulator 3000
system memory from 0x0 ~ 0x3000
>>> load test1.obj
4096 4218
1000 1410334820390010362810303010154820613C100300102A0C103900102D
101E 0C10364820610810334C0000454F46000003000000
2039 041030001030E0205D30203FD8205D2810303020575490392C205E38203F
2057 1010364C0000F1001000041030E02079302064509039DC20792C1036
2073 3820644C000005
>>> █
```

用show 顯示記憶體中的內容

```
>>> show
0000 .....
0010 .....
0020 .....
0030 .....
0040 .....
0050 .....
0060 .....
0070 .....
0080 .....
0090 .....
00A0 .....
```

```
0FF0 .....
1000 14103348 20390010 36281030 30101548
1010 20613C10 0300102A 0C103900 102D0C10
1020 36482061 0810334C 0000454F 46000003
1030 000000.. .....
```

```
2020 .....
2030 ..... ..041030 001030E0
2040 205D3020 3FD8205D 28103030 20575490
2050 392C205E 38203F10 10364C00 00F10010
2060 00041030 E0207930 20645090 39DC2079
2070 2C103638 20644C00 0005.... .....
```

用run模擬執行，並印出暫存器的內容

```
>>> run

LDCH  A : 000046 X : 000002 L : 001024 PC : 00206D SW : 00003C
WD    A : 000046 X : 000002 L : 001024 PC : 002070 SW : 00003C
TIX   A : 000046 X : 000003 L : 001024 PC : 002073 SW : 00003D
JLT   A : 000046 X : 000003 L : 001024 PC : 002076 SW : 00003D
RSUB  A : 000046 X : 000003 L : 001024 PC : 001024 SW : 00003D
LDL   A : 000046 X : 000003 L : 00207A PC : 001027 SW : 00003D
RSUB  A : 000046 X : 000003 L : 00207A PC : 00207A SW : 00003D

A : 000046 X : 000003 L : 00207A PC : 00207A SW : 00003D
```

用unload卸載test1.obj，重設記憶體與暫存器

```
>>> unload
A : 000000 X : 000000 L : 000000 PC : 000000 SW : 000000

>>>
```

用show看重設後的記憶體內容

```
>>> show
0000 .....
0010 .....
0020 .....

0FF0 .....
1000 .....
1010 .....
1020 .....
1030 .....
1040 .....

2000 .....
2010 .....
2020 .....
2030 .....
2040 .....
2050 .....
2060 .....
2070 .....
```

載入test2.obj

```
>>> load test2.obj
8192 45
2000 00201E1820241C201B0C20270020211820241C201B0C202A4C0000000001
201E 000005000007000003
>>> █
```

用show 顯示記憶體中的內容

2000	00201E18	20241C20	1B0C2027	00202118
2010	20241C20	1B0C202A	4C000000	00010000
2020	05000007	00000300

用run模擬執行，並印出暫存器的內容

```
>>> run
2000 202D
LDA      A : 000005  X : 000000  L : 00202D  PC : 002003  SW : 000000

ADD      A : 000008  X : 000000  L : 00202D  PC : 002006  SW : 000000

SUB      A : 000007  X : 000000  L : 00202D  PC : 002009  SW : 000000

STA      A : 000007  X : 000000  L : 00202D  PC : 00200C  SW : 000000

LDA      A : 000007  X : 000000  L : 00202D  PC : 00200F  SW : 000000

ADD      A : 00000A  X : 000000  L : 00202D  PC : 002012  SW : 000000

SUB      A : 000009  X : 000000  L : 00202D  PC : 002015  SW : 000000

STA      A : 000009  X : 000000  L : 00202D  PC : 002018  SW : 000000

RSUB     A : 000009  X : 000000  L : 00202D  PC : 00202D  SW : 000000

A : 000009  X : 000000  L : 00202D  PC : 00202D  SW : 000000

>>> █
```

用exit退出模擬器

```
>>> exit

Lyciih@DESKTOP-CR5NUFU MINGW64 ~/Desktop/github/System-Programming/SIC simulator (main)
$ █
```

討論

在這個作業的過程中我領悟到原來市面上的很多模擬器好像都可以用這種方法來實現，只要知道該指令集的各種行為，並用變數模擬CPU中的每個暫存器，再加上模擬用的空間以及虛擬出的設備記憶體位置，就可以模擬一台電腦的運作，相當好玩。