Title Figure: Chinese Mooncakes

CSC317: Computer Organization & Architecture
Program 1: T34 Emulator
Kendra Deziel
Fall 2017

PURPOSE

The purpose of this program is to serve as an emulator of a T34 processor. Right now it is only 4096 words of 24-bit memory which can read in a program file (.obj) and automatically prints all memory that has been loaded to the terminal screen. The program is meant to be run on terminal. After printing all memory, the program prompts the user if they would like to utilize the Parsing Function, which will allow them to input two memory address and receive the ADDR (operand address), OP (opcode), and AM (addressing mode) in 12-6-6 bit binary strings, respectively.

FUNCTIONS

Main Function

The Main Function reads the input file, puts each program info into the appropriate memory location, sets the Program Counter, and controls the user's access to the Parsing Function. It is complete with input error checking and the entire program will terminate when user input dictates an end to use of the Parsing Function.

Input: command line argument of <myProgram.obj> formatted as follows:

```
50 1 000000
c4 5 050404 200800 300800 102840 050c00
101 2 300 9
200 1 30
300 1 10
c4
```

Output: Prints to terminal

Memory Dump

The MemoryDump Function outputs all memory locations and corresponding information ONLY WHERE information exists. Each memory location which contains "0" will be skipped.

Input: none (uses global variables)

Output: Prints to terminal

Parsing Function

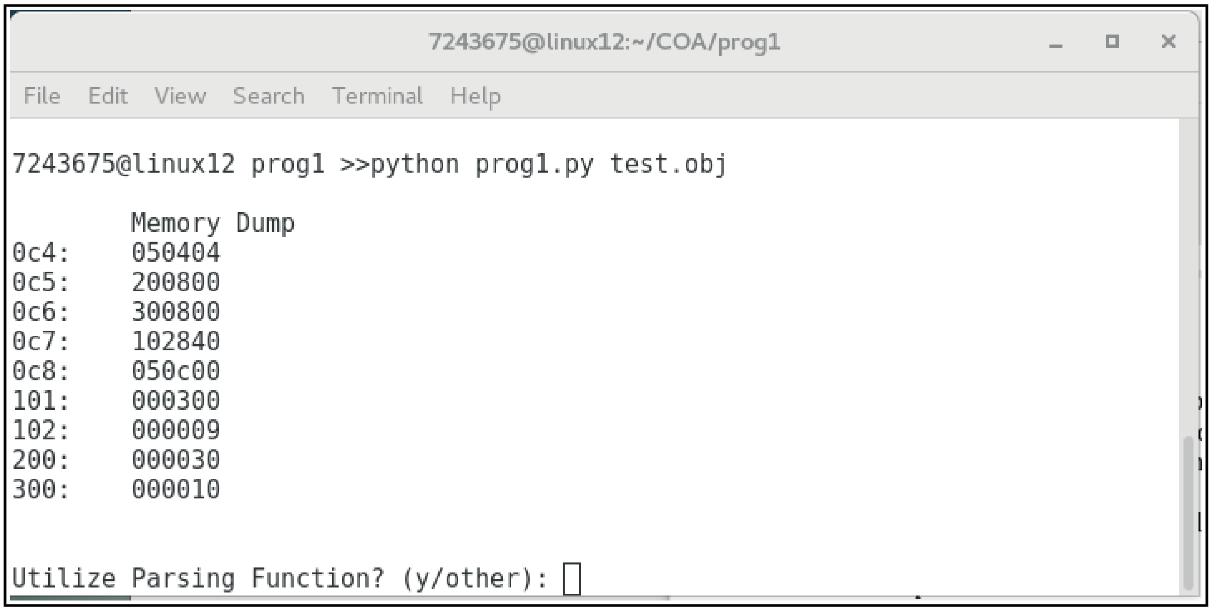
The ParsingFunction Function gives binary info for a sequence of memory locations. The binary is broken down into ADDR (operand address), OP (opcode), and AM (addressing mode). The input is two ints that represent the address location where this info can be found.

Input: one and two, both int addresses to index memory and access requested information

Output: Prints to terminal

INSTRUCTIONS FOR USE

The T34 emulator was created to be run in a terminal window. Figure 1 shows the memory dump output after running the program as written in the terminal. (prog1 directory is given tarred and zipped and must be extracted before use.)



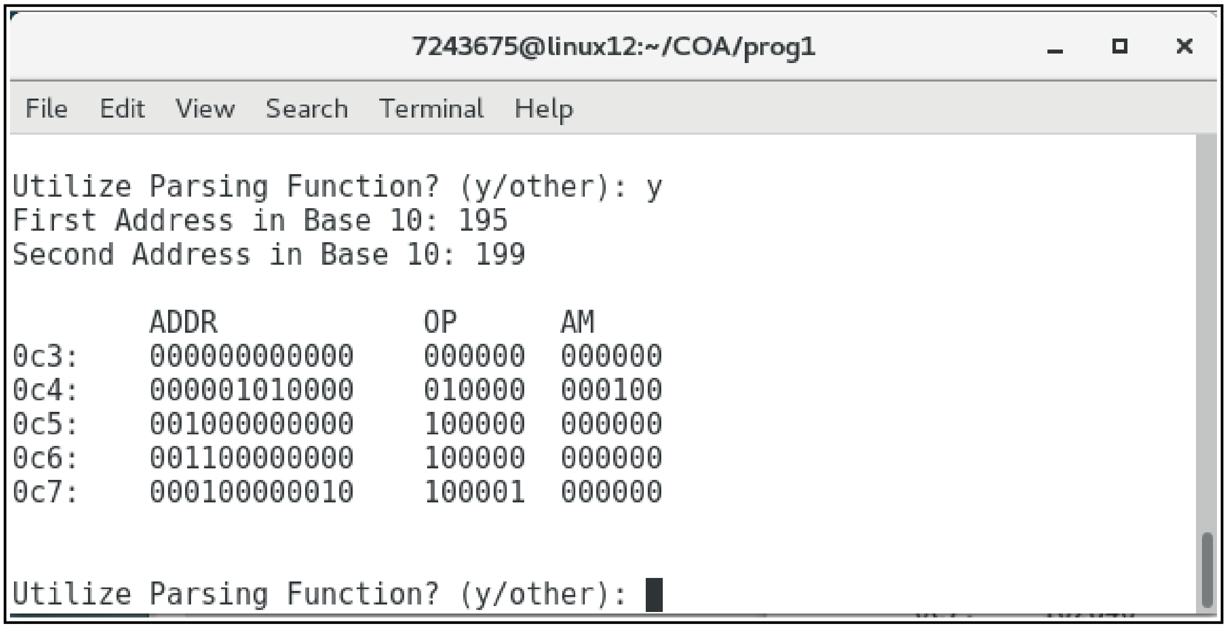
```
7243675@linux12 prog1 >>python prog1.py test.obj

        Memory Dump
0c4:    050404
0c5:    200800
0c6:    300800
0c7:    102840
0c8:    050c00
101:    000300
102:    000009
200:    000030
300:    000010


Utilize Parsing Function? (y/other):
```

Figure 1: prog1.py terminal output after running

Figure 1 also shows that after memory dump, the program asks the user if they would like to use the parsing function. If they type “y” or “yes”, the program prompts for an address. If the address is properly entered, it will prompt for a second address and then display the parsed information for all addresses between the two addresses, including information for the two addresses given. (See figure 2.) If the first address given is greater than the second address given, no information will be printed.



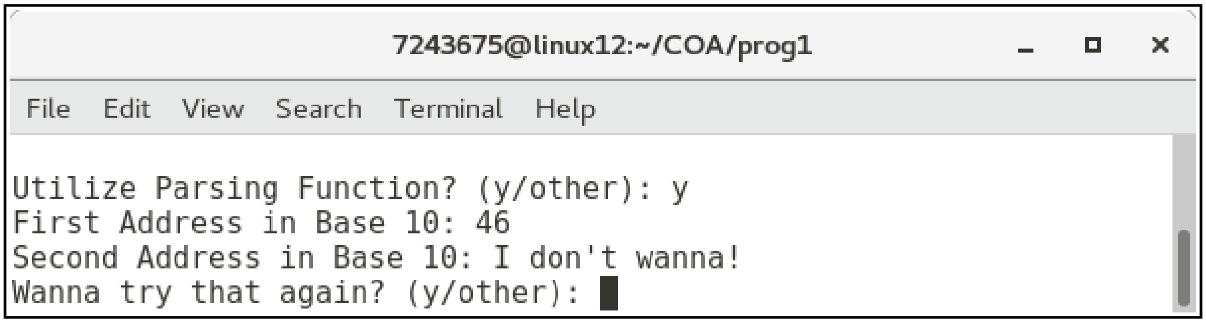
```
Utilize Parsing Function? (y/other): y
First Address in Base 10: 195
Second Address in Base 10: 199

        ADDR            OP      AM
0c3:    000000000000    000000  000000
0c4:    000001010000    010000  000100
0c5:    001000000000    100000  000000
0c6:    001100000000    100000  000000
0c7:    000100000010    100001  000000


Utilize Parsing Function? (y/other):
```

Figure 2: correct use of Parsing Function in prog1.py

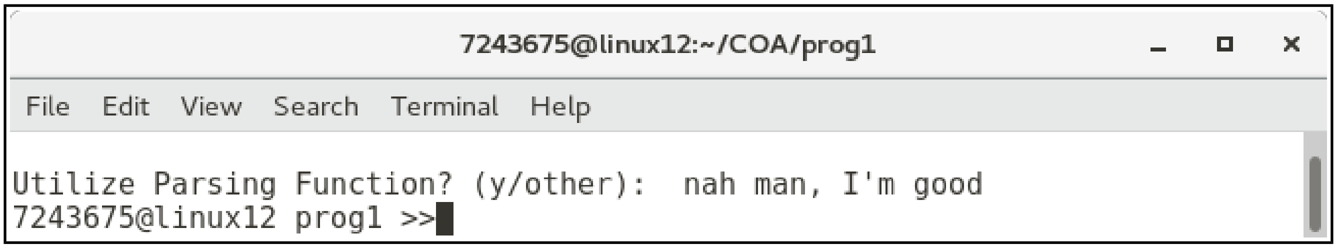
If the user gives an address that is not in base 10, or cannot be converted to an integer, figure 3 shows how they will receive a snarky response prompting if they would like to start the process over.


7243675@linux12:~/COA/prog1
File Edit View Search Terminal Help

```
Utilize Parsing Function? (y/other): y
First Address in Base 10: 46
Second Address in Base 10: I don't wanna!
Wanna try that again? (y/other): 
```

Figure 3: incorrect address input for the parsing function

When the user is done utilizing the parsing function, they can enter any input other than “y” or “yes” and the program will terminate, as shown in figure 4.


7243675@linux12:~/COA/prog1
File Edit View Search Terminal Help

```
Utilize Parsing Function? (y/other):  nah man, I'm good
7243675@linux12 prog1 >>
```

Figure 4: exiting the T34 emulator