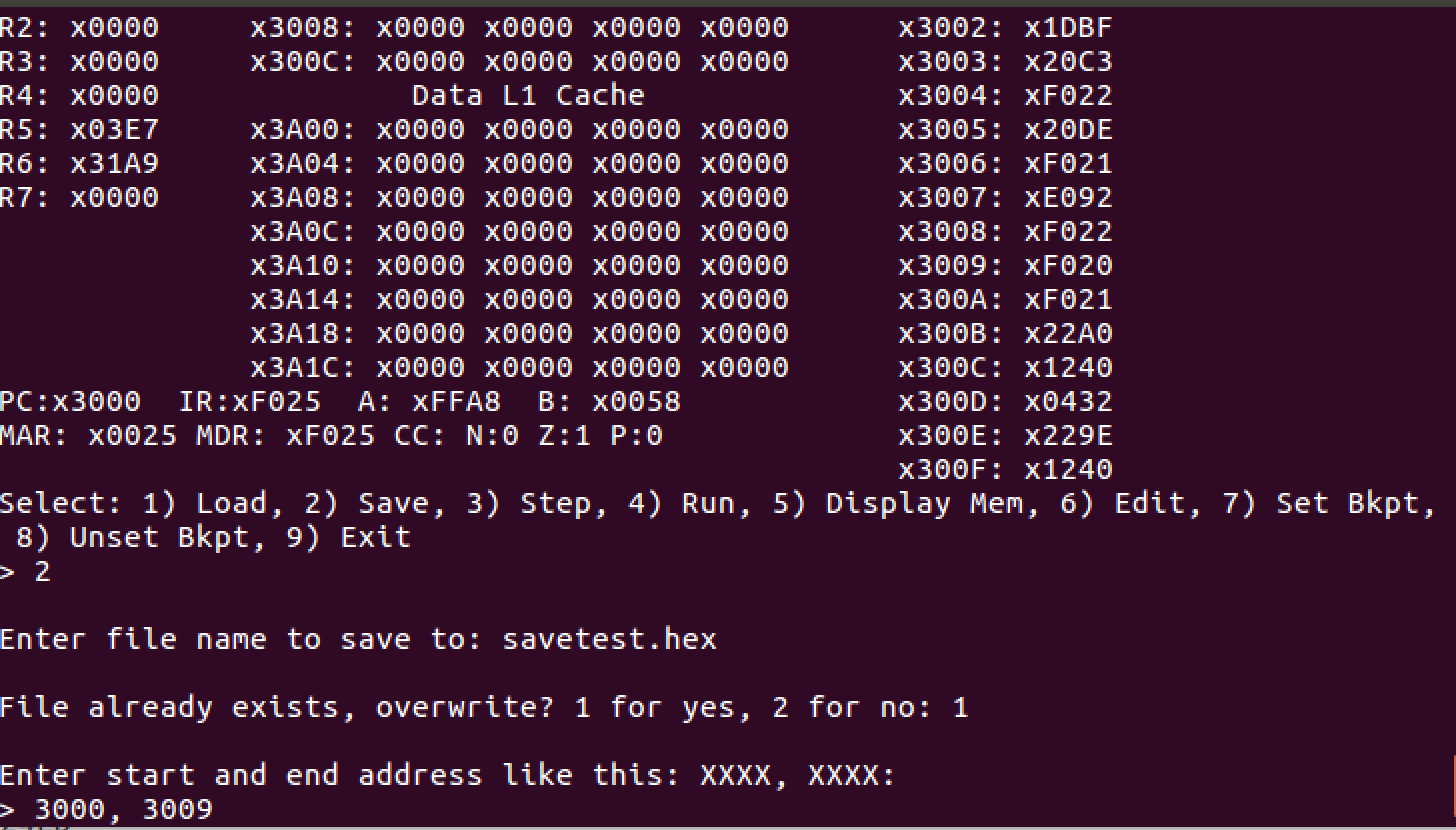
Lovejit Hari

Vladimir Kaganyuk

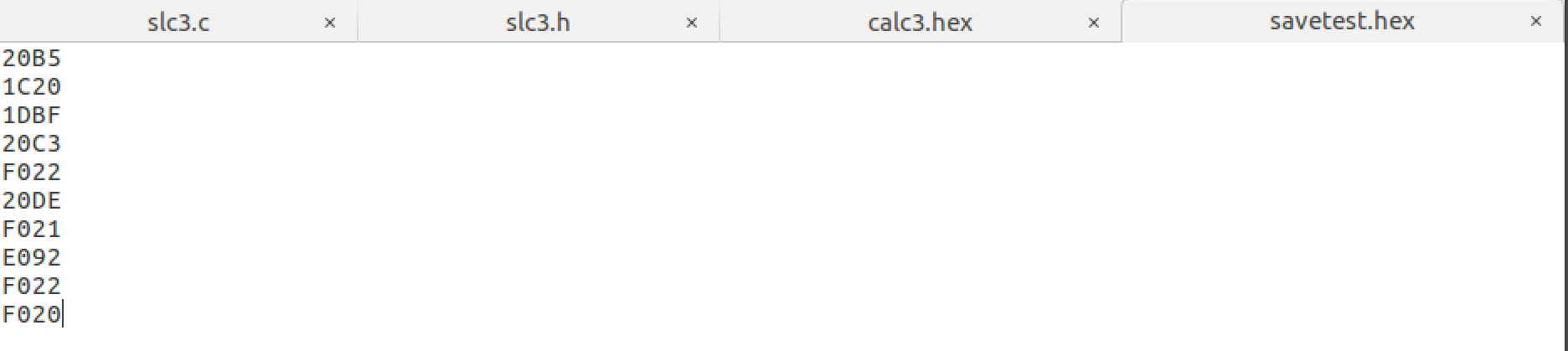
Dongsheng Han

**Save function**

Prompts user for a save file, if it exists a new prompt asks the user if they would like to overwrite that file.

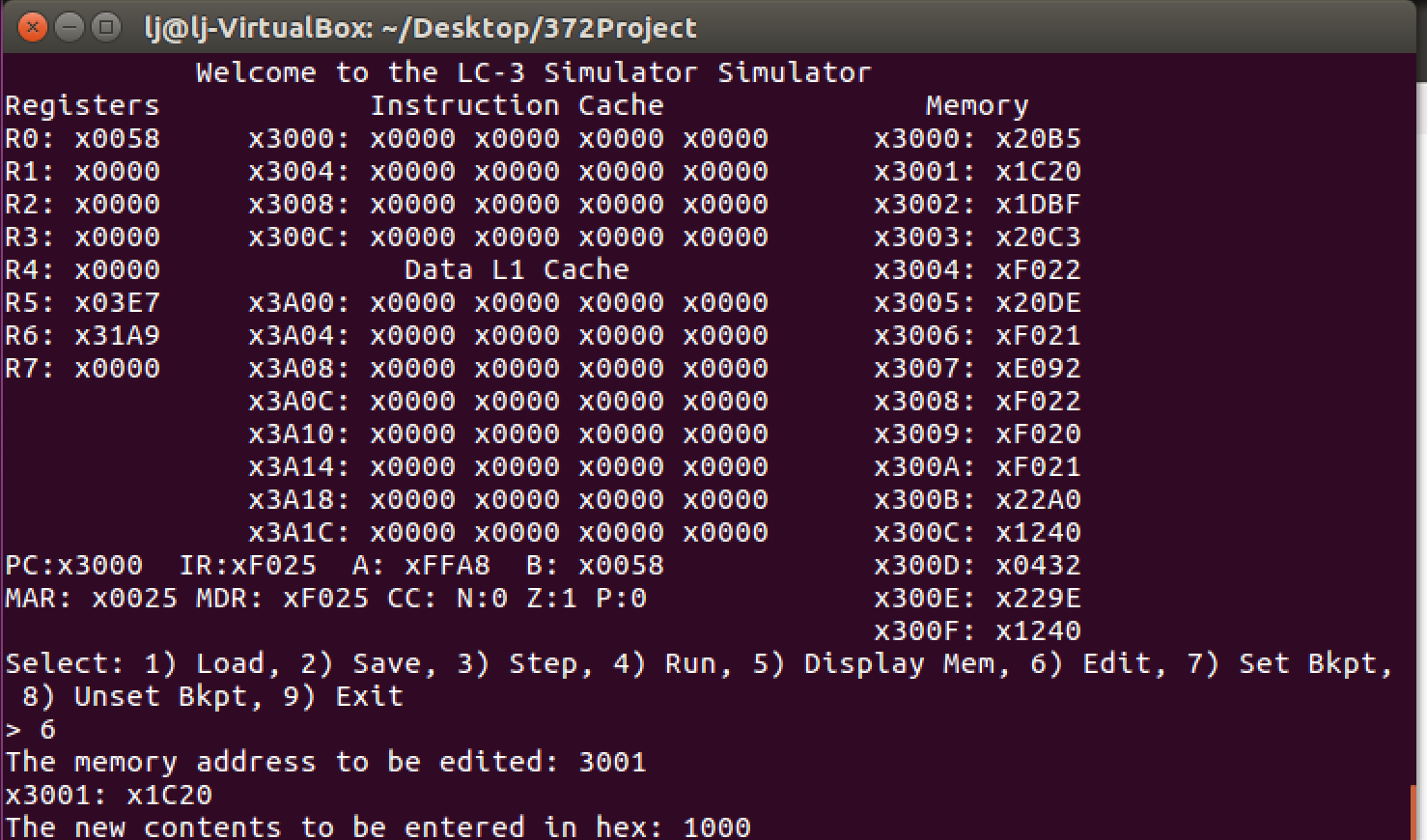


Contents of the save.hex file after displaying from x3000 to x3009

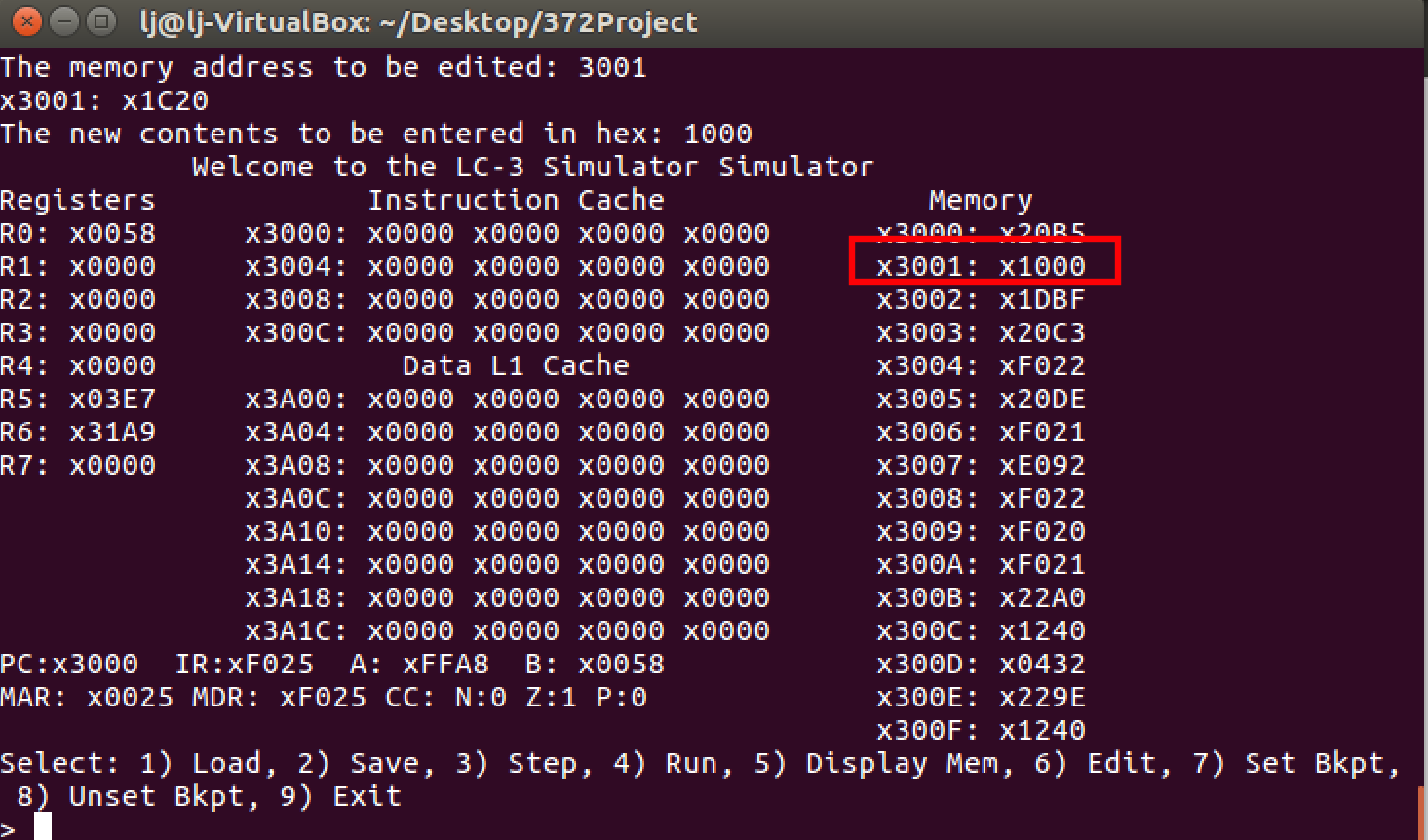


**Edit Function**

The user is prompted for an address to be edited. The address is then displayed showing the contents as well. The user then changes the value at that location.

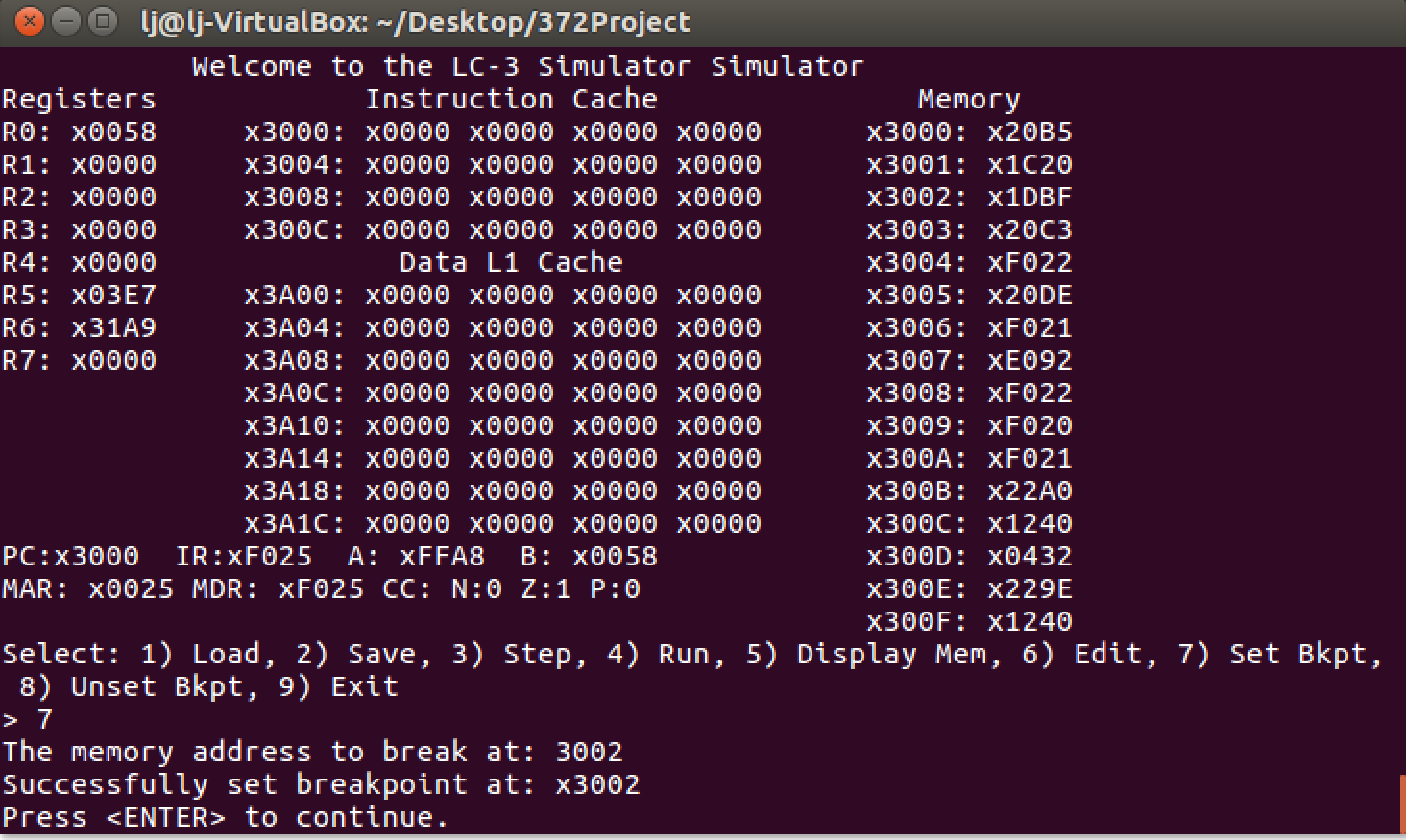


The value entered then changes the value at 3001 while everything else remains the same.

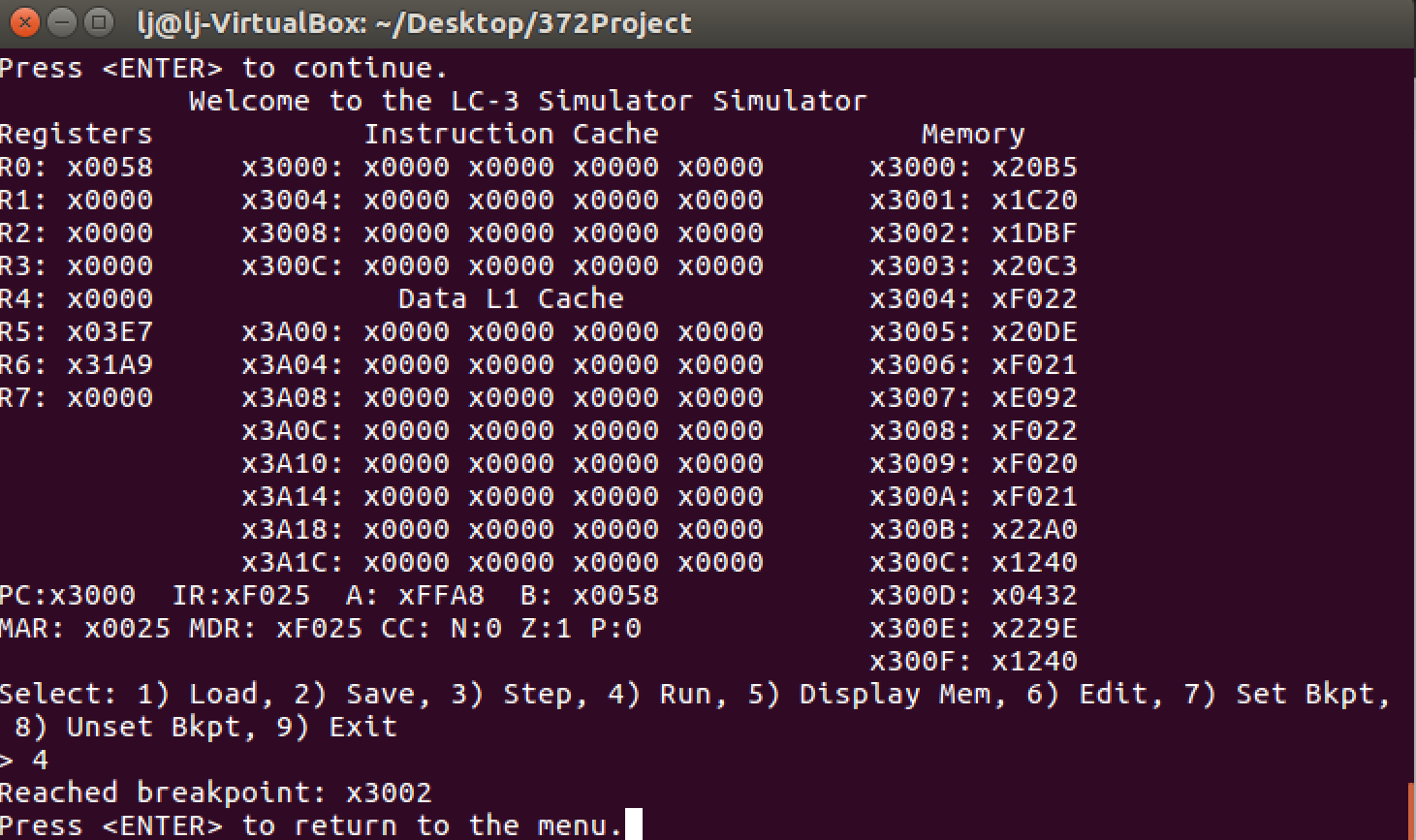


**Set/Unset Breakpoints functionality**

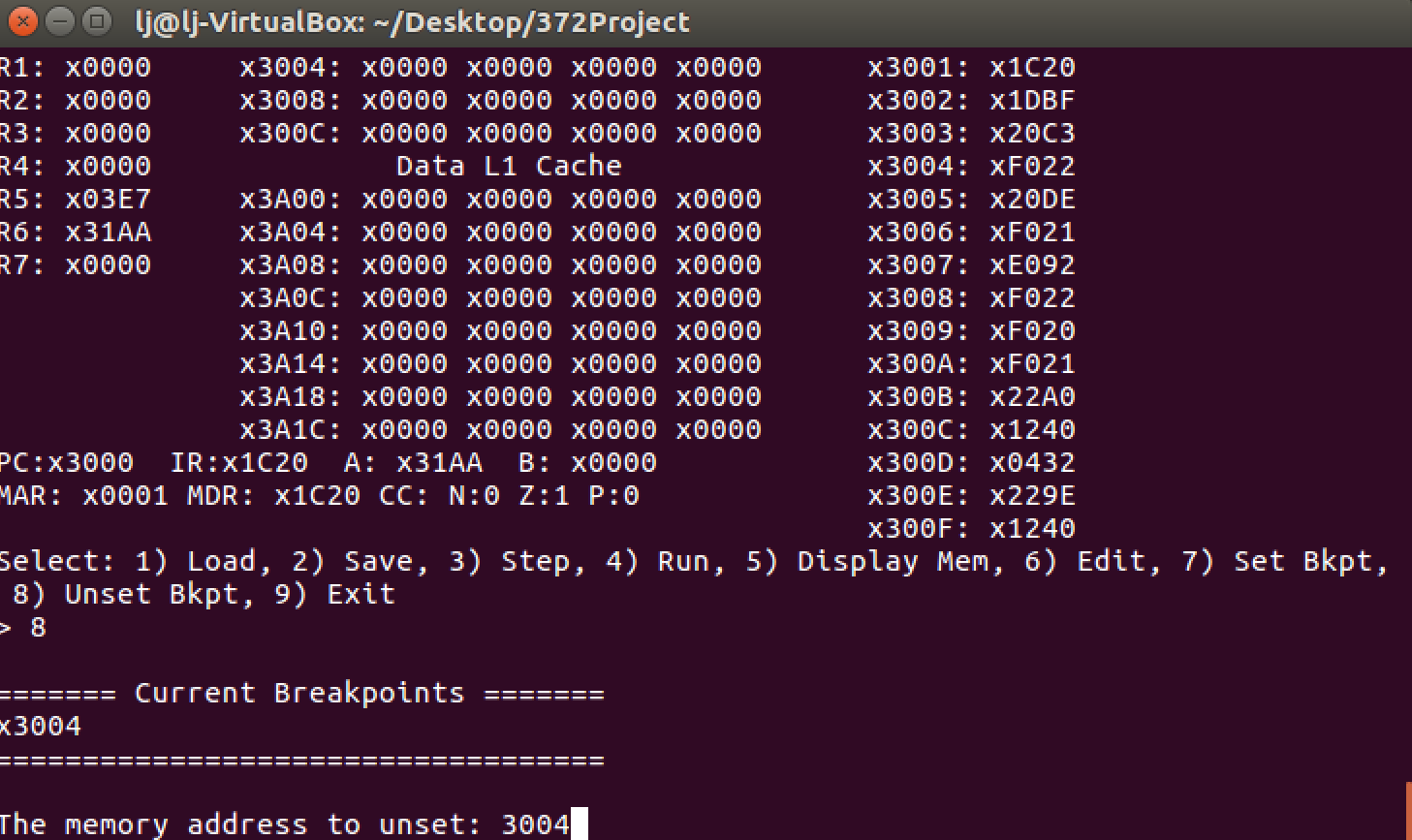
The user is prompted to choose a breakpoint. In the following example I have chosen address 3002 for simplicity purposes.



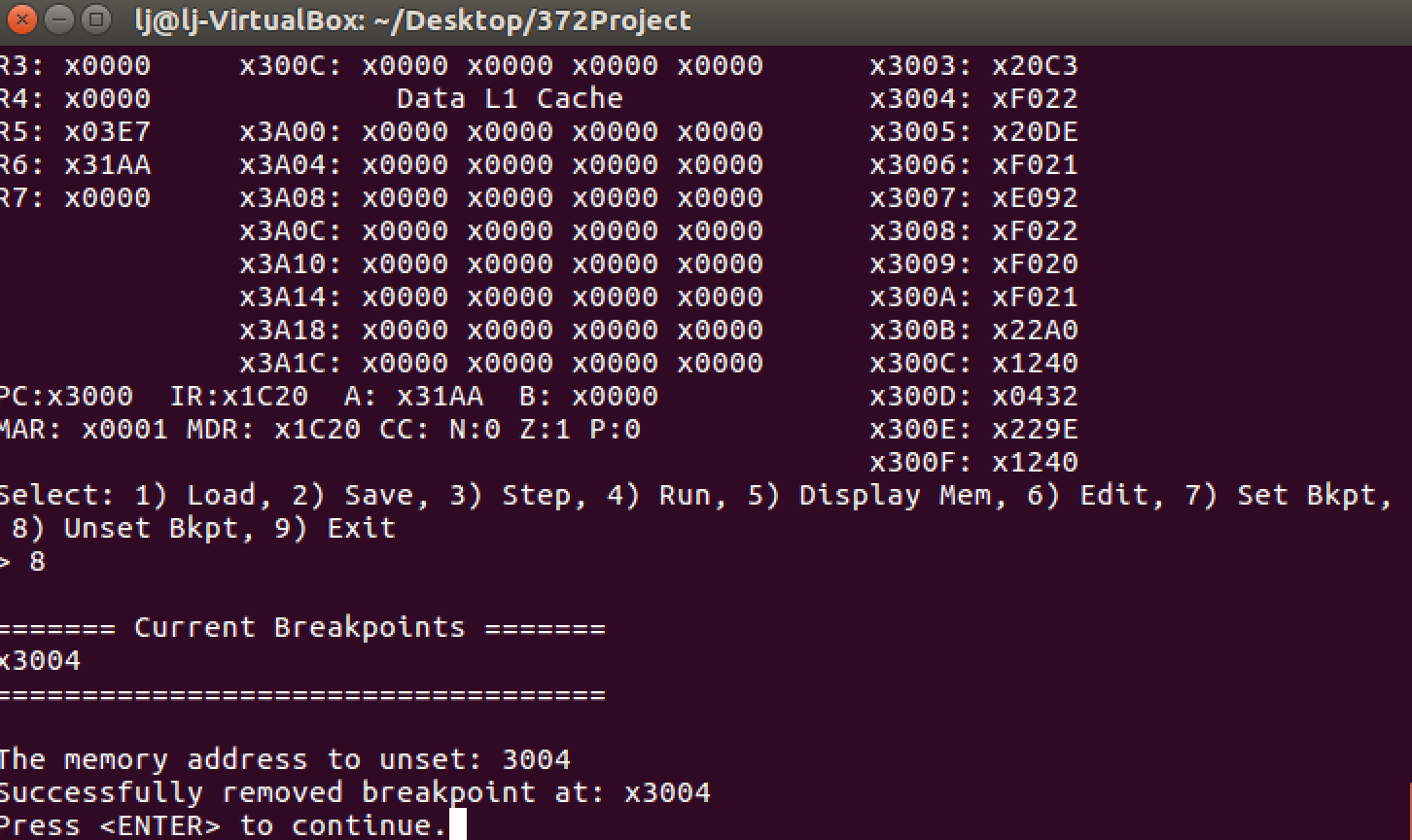
After running the program, the user then gets prompted with a message letting them know they have encountered a breakpoint.



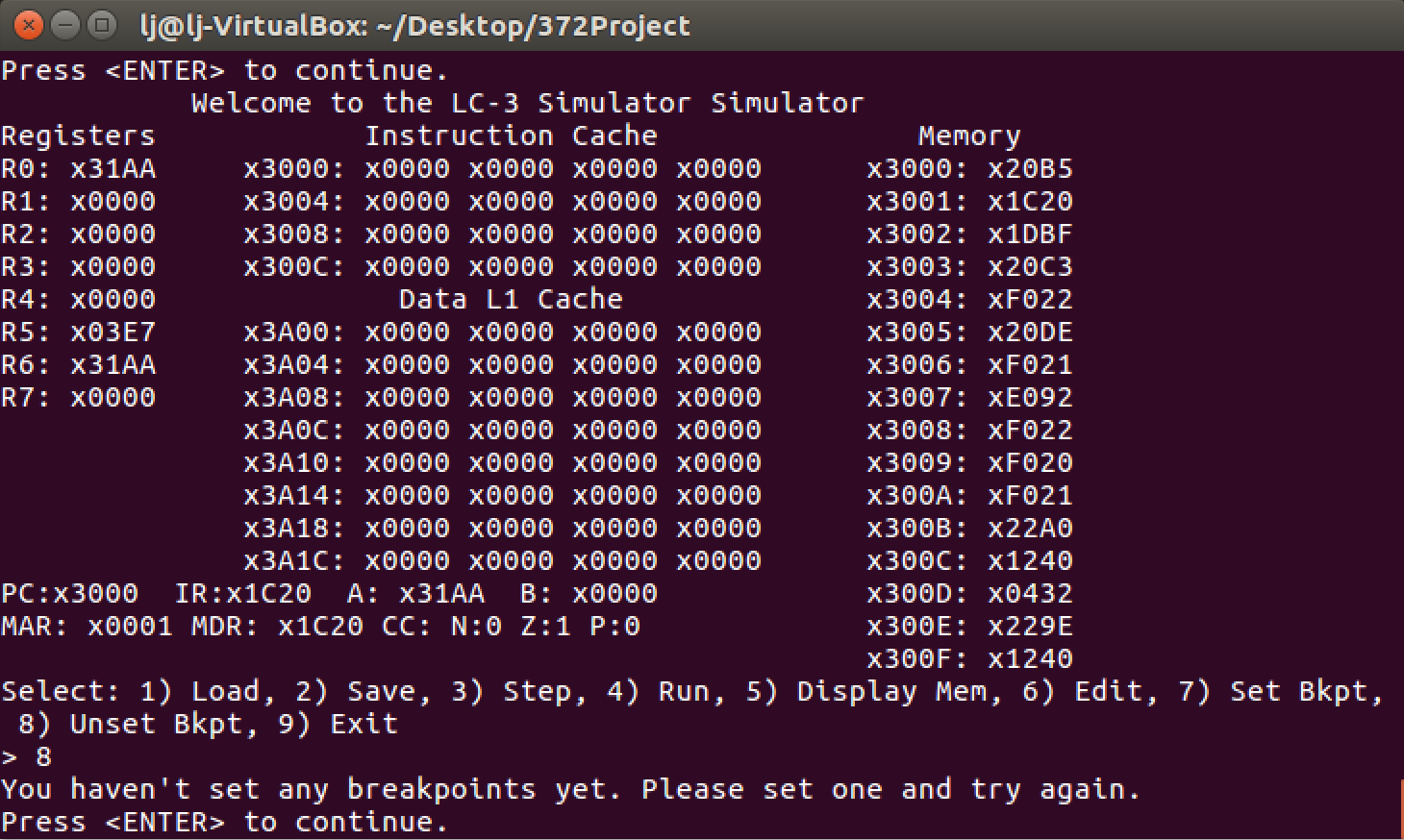
The user then has the option to unset a breakpoint, they are shown what breakpoints are currently set in the program.



The breakpoint is then successfully removed from the program



If no breakpoints exist, then the user is prompted with a message letting them know that they haven’t set any.

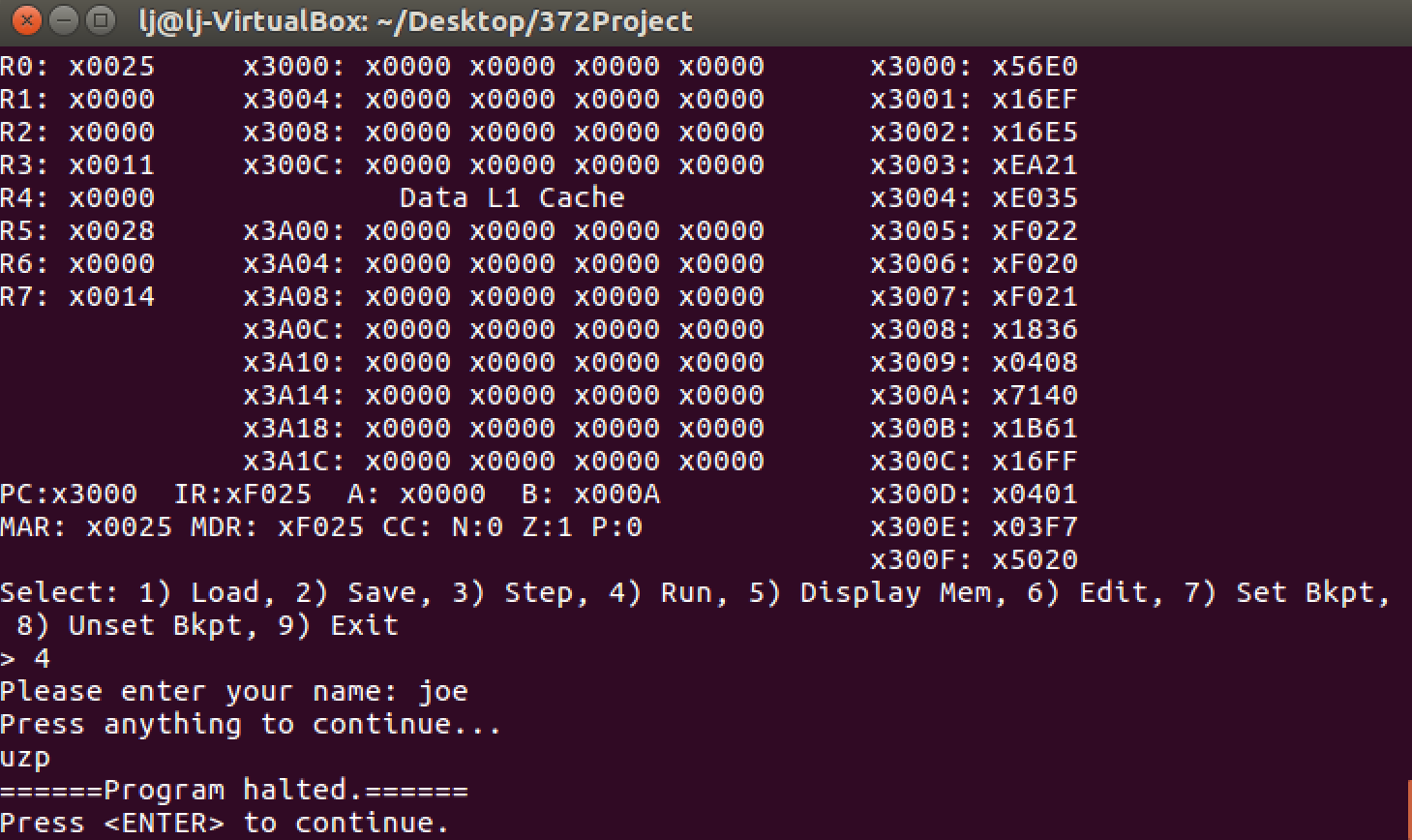


**Cache functionality**

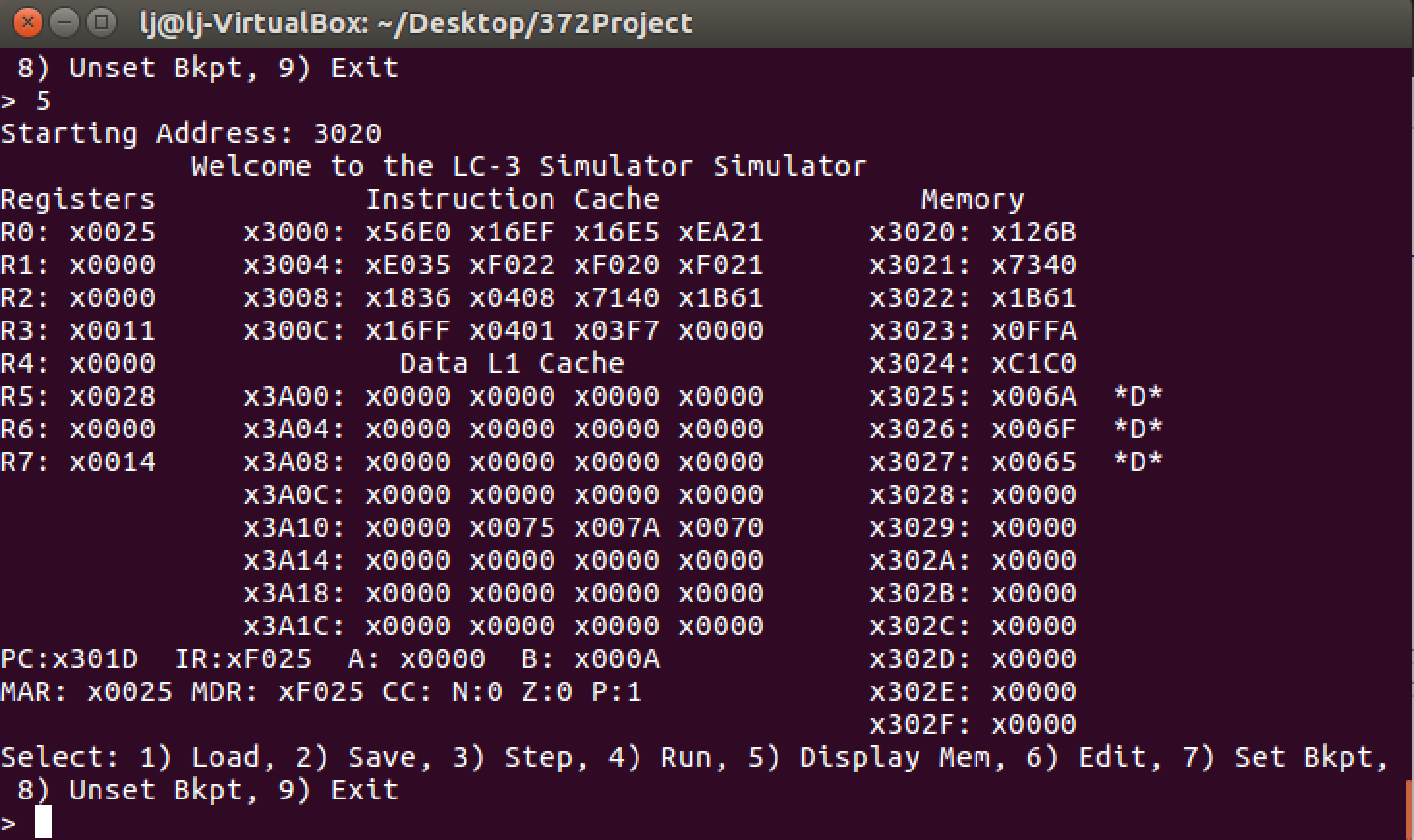
We used a write-back policy for our direct-mapped cache.

Here are fruits of our labor…

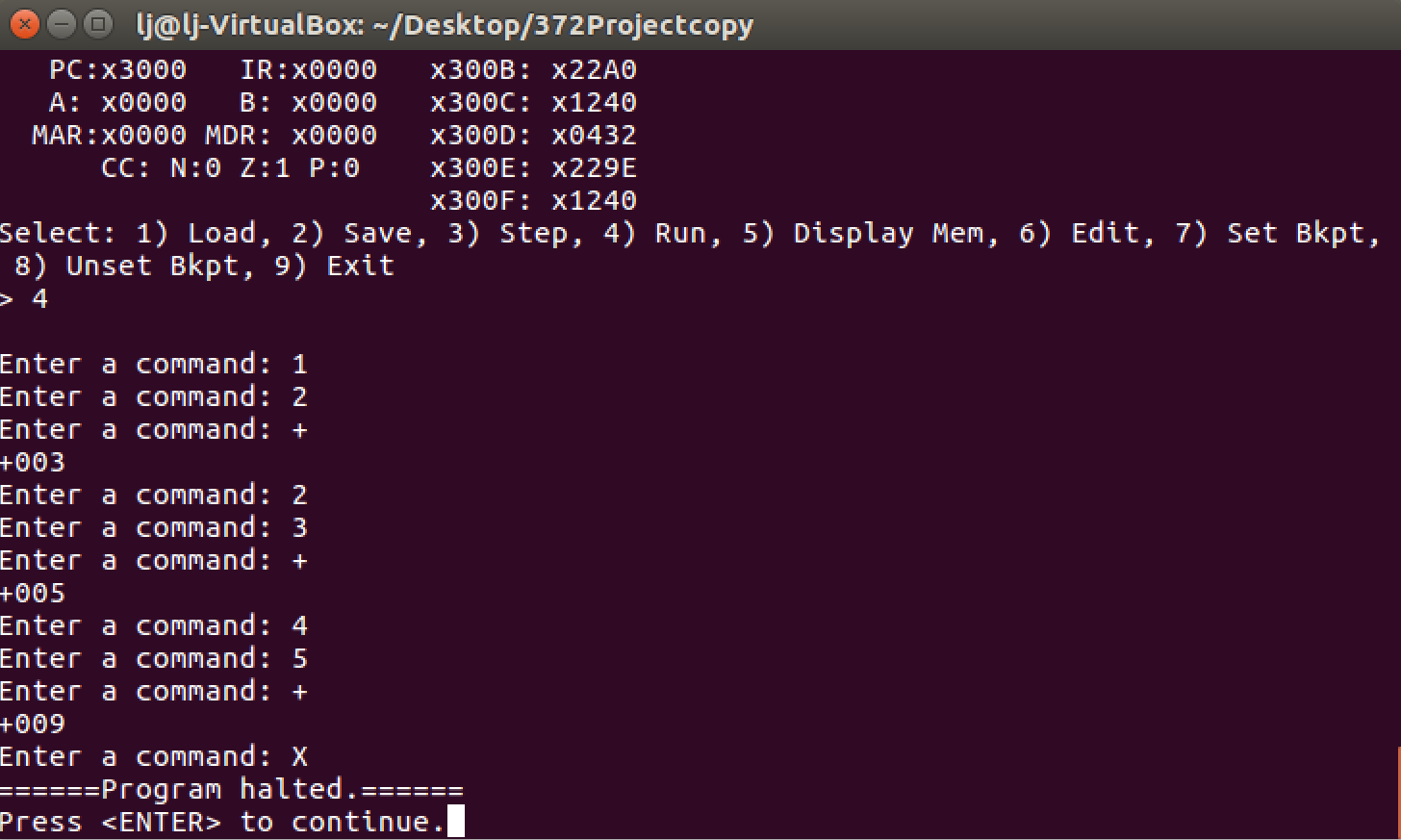
First we enter in a name, for example: joe



Then we display memory starting from location 3020 to show where the dirty bit was set. The cache is inconsistent at 3 places therefore there are 3 set dirty bits.

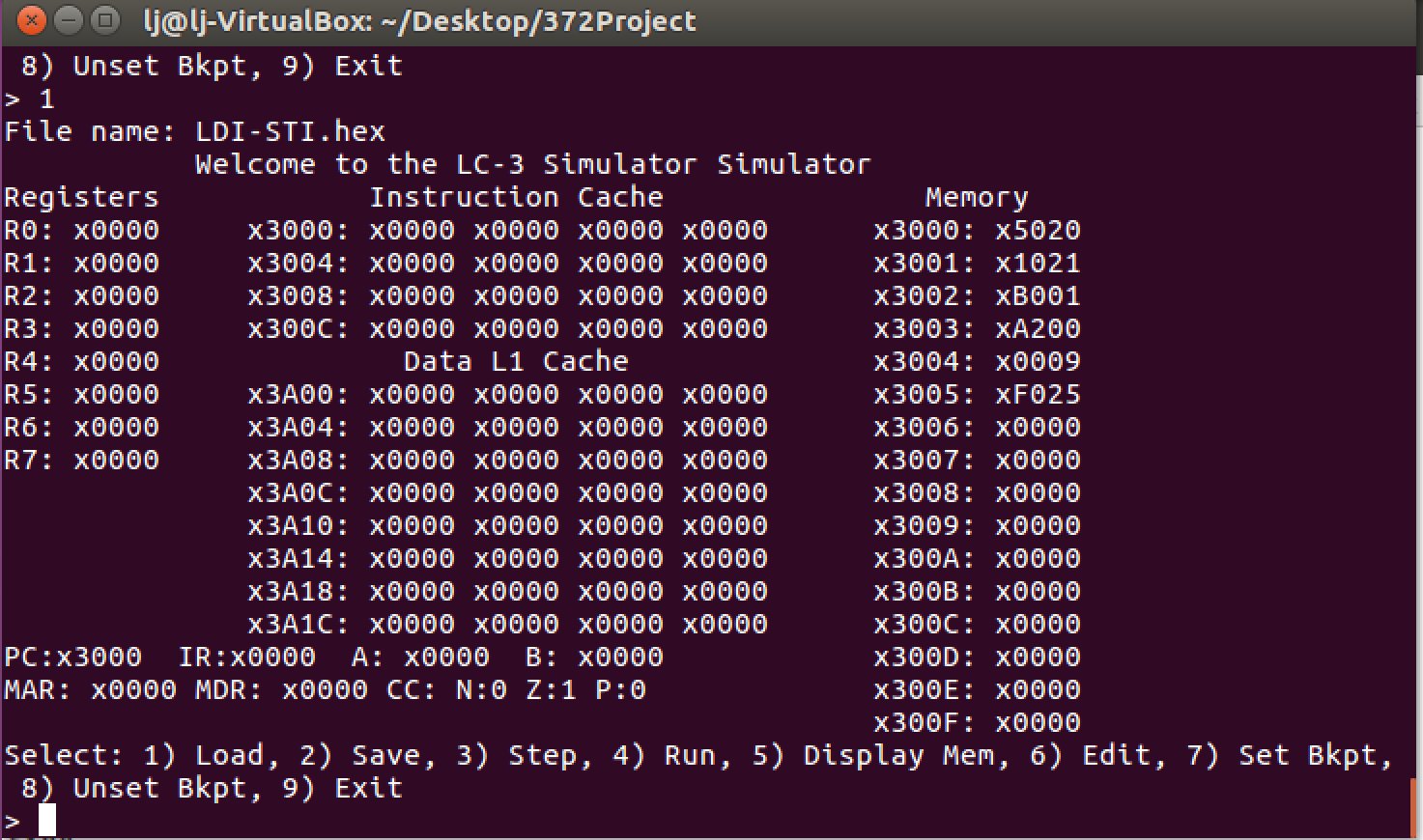


**RPN Calculator**



**LDI/STI test**

Loading in the program



1st/2nd steps of the program simply AND R0 to 0, and ADD 1 to R0

