

CS270 Recitation 10

“LC-3 I/O Exercise”

Goals

- To gain a better understanding of how TRAP calls are implemented.
- To reinforce the commands available for managing I/O in LC-3.

Assignment

Make a subdirectory called R10 for the recitation, all files should reside in this directory. Download trap.asm from <http://www.cs.colostate.edu/~cs270/Fall12/recitations/R10/trap.asm>. Now, open the assembly file in your favorite editor.

For this assignment you will be adding a new TRAP routine, GETSP, to the LC-3 simulator. The routine is the counterpart to PUTSP, which is already implemented. Both functions are more memory efficient, since they store and read two ASCII characters from a single sixteen bit word. To accomplish this, the first character is stored in the bits 7 to 0, while the next character is stored in bits 15 to 8. To see both formats, load trap.asm into the simulator and compare the data starting at MESSAGE and MESSAGEP.

This recitation uses TRAP x00 for GETSP. Before you can use TRAP x00, though, the machine needs to know where TRAP_GETSP is. To accomplish this, write the address of TRAP_GETSP to address x0000 using the simulator. This address will now be loaded into PC whenever TRAP x00 is called.

Part 1:

The code for GETSP has been provided in trap.asm. Notice that most of the comments are missing from the code. Your task is to understand what the provided code is doing and use that knowledge to add comments to the file.

The comments should describe what each line is doing in the context of the entire GETSP operation, not necessarily what operation is being performed on a given register at the time. See the trap.asm file for examples.

Part 2:

GETSP, as it is implemented now, is a dangerous routine that makes programs using it vulnerable to exploits. Consider the following problematic code:

```
int myarray[10];
for (int i=0; i < 16; i++) {
    myarray[i] = i;
}
```

How can the assembly routine be changed so that dangerous behavior does not happen? Write your answer in the FIXME comment at the top of TRAP_GETSP.

Hint: Run “man gets” and scroll down to the BUGS section.

Conclusions:

Show your commented trap.asm file to your TA.