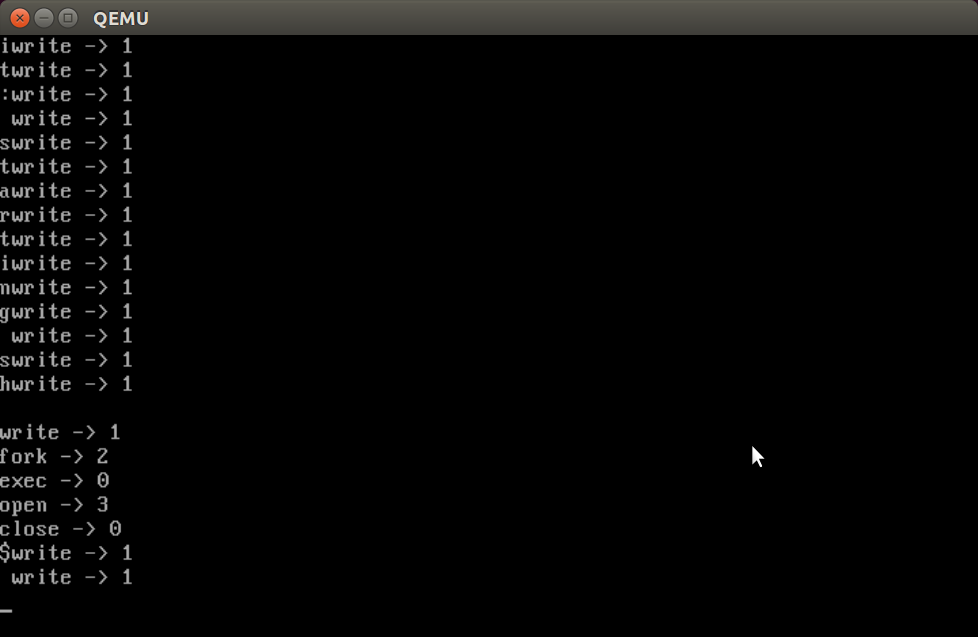
William Tran

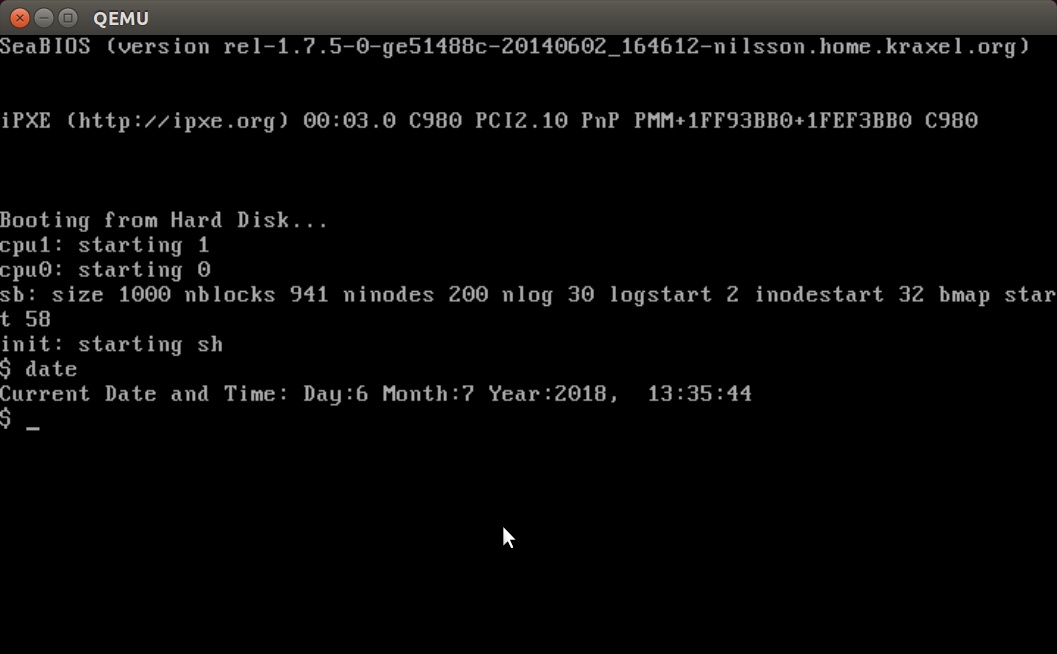
2018/07/04

Homework (3) Report

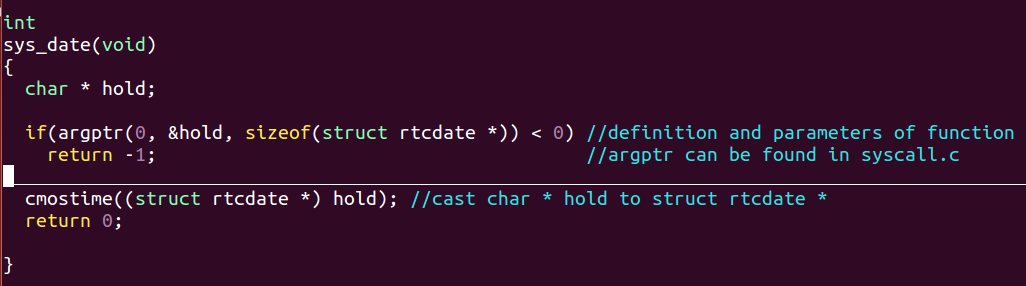
1. My output from part 1 (System Call Tracing):



2. My output from part 2 (Date System Call):



Modifications in files for date system call:

1. usys.S: first of all, I added a system alarm for date in usys.S, so when I input *date* in the xv6 shell prompt, it invokes a system call.
2. Also, to make the program runnable from the xv6 shell, I add \_date/ to the UPPROGS definition in the Makefile.
3. syscall.h: in syscall.h, I define a new system call number (22) for SYS\_date, so syscall can load it from the trap frame and index it into the system call tables. Then, when the system call is invoked, syscall will invoke the 22nd entry of the system call table, which corresponds to invoking sys\_date.
4. In syscall.c, I have to define the function sys\_date, , and I match the value SYS\_date to sys\_date. 
5. In user.h, I initialized the date function. It takes in struct rtcdate, then stores the date information in the struct and returns. 
6. In sysproc.c, I implement function sys\_date(void) to use cmostime to get the current date and time. 
7. Finally, I format the date and time in date.c so the information can appear on the shell. Then, I input make, then make qemu in the terminal to get the result above. 