Gloria Nduka
February 2021
Project 1

The files I edited in the xv6 folder are bolded below.

proc.h
In proc.h I added the function prototype "void trace(void)" on line 15 so it can be used in syscall.c
I added 3 variables in the proc struct in lines 53-55:
int onffint onoff - To keep track if tracing is on or off for this process
int currsys - syscall id
int numsys - number of syscalls on current process

proc.c
In proc.c I created an array, "sysname", of strings in line 24 to associate unique syscall number with string name and then I implemented the trace function, void trace (void), to implement printing process in lines 542-550
In the function I implemented it to print each system call from a process, the system call unique number, the process id and the process name when turned on. I also check if the on/off tag of that process is greater than 0 (on).

sysproc.c
I also edited the sysproc.c file. This file is a collection of process-related system calls. The functions in this file are called from syscall. I implemented 2 functions, int sys_trace(void) and int sys_date(void), lines 96-113 and 118-177 respectively. In int sys_trace(void) I used the argint helper function to check if input if 0 or non zero then update the tracing bit in the proc struct based on the input. Then I return the number of syscalls.
In sys_date(void), I print out the current time in UTC time. I used helper function, cmostime(), defined in lapic.c which reads real time clock. I then retrieve the pointer through argptr and cast that pointer to a rtcdate struct pointer. I then pass that casted pointer into the cmostime() function as it populates the passed rtcdate with proper data. I then organize the information into something readable. I convert the month into strings instead of printing the number; I use the array of month strings. The format I print out is Month Day, Year Hour:Minute AM/PM. I am doing the 12 hour system not 24 hours so I included am/pm.

user.h
I edited the user.h file and this file contains the user-side function prototypes of the system calls and the utility library functions. I added:
int trace(int);
int date(struct rtcdate*);
on lines 26-27.

usys.S
I edited usys.S and this file I included macros for the assembler code for each system call. The macro places the system call number into the eax register and then invokes the system call. I added my macro entry for the trace and date system calls on lines 32-33
SYSCALL(trace)
SYSCALL(date)

syscall.h
I edited this file and it has definitions of the system call numbers. I defined consecutive unique numbers for trace and date on lines 23-24. These numbers are indices of a table of pointers defined in syscall.c.
#define SYS_trace 22
#define SYS_date   23

syscall.c
I edited syscall.c and included line 87:extern int sys_date(void) and line 103: extern int sys_trace(void). These were listed in alphabetical order along with other syscall functions.
I also added [SYS_trace]   sys_trace and [SYS_date]   sys_date in the *syscall[] array on lines 131, 132.
I also edited the syscall function in lines 137-153. The syscall(void) function is the entry function for all system calls. The syscall function checks the unique syscall integer to ensure that it is in the correct range and then calls the corresponding function. I basically took the current process and updated/applied the new/current system call then updating it in the proc struct in proc.h to help keep track of system call called for that process. I am also keeping track of the number of syscalls for that process and updating it; then calling the trace function to print out the tracing information required - the process ID, the process name, the system call number, and the system call name.

date.c
I didn't need to change anything in date.c since I did all the printing functionality in sysproc.c in my sys_date function; a TA at OH said it was okay if I didn't implement anything in date.c. however I did include the source code that was provided in the project write up.

Makefile:
In the Makefile under UPROGS I added lines 184-185, "_test_project1\ _date\"
and in line 258 I added this line under EXTRA, test_project1.c\ so the date.c and test_project1.c can compile and run when type "make" in the terminal.

pr.pl, sign.pl
I didn't modify this file but the system claimed I modified it since I accidentally copied old pr.pl and sign.pl files into the new directory that I was pushed to github.

vectors.pl

I had this error when running the project: “Had undefined reference to vectors” so I entered these commands:

```
sudo chmod +x *.pl
make clean
make qemu (or make qemu-nox)
```

which then modified this file.