Operating Systems 2 – CS3523
Programming Assignment 2 – Report
Suraj Telugu – CS20BTECH11050

➢ System Calls Working Principle:

1) The macro for systemcall(name) to interact with kernel is defined in usys.S and all the 21 system calls gets implemented in the during booting.

2) All the 21 system calls are defined in defs.h and user.h and their respective functions are defined in proc.c. All the header files and required functions to implement a system call function are included in proc.c.

3) A system call can have any type of input but usually gives void or int output. In sysproc.c special functions int sys_<syscall>(void) are defined which provides the respective return types or success/failure of any system call.

4) By making its input void the macro defined works if these sys_syscall functions are not defined then for every system call we need to write different macro, assembly instruction.

5) Each System call is given a specific number in syscall.h. In static int syscalls[num] array function which stores return values of the system calls form sys_<syscall> function.

6) syscall() function stores the return value of the system call in proc->tf->eax if system call is invalid then it returns -1. syscall() function is invoked in trap function which provides interrupts whenever if there are any errors and helps to transfer from user to kernel mode.

➢ Conclusion:

1) In the first part of the assignment 1 have understood how syscall() function works and why is it necessary. The num variable in the syscall function refers to the type of system call and as explained above syscalls[num]() array function gives the return value of system calls. After understanding how those variables are used, I have added cprintf statements using if – else cases for each system call and have printed the output in the given format which is syscall->return value. write system call is removed for better readability for part 2.

2) In the second part for creation of system call I have gone through the parts of code where system calls are created and how they are created. As explained above the new system call function int date(struct rtcdate*) is defined in defs.h, user.h and added into usys.S. The function for date() is implemented in proc.c, in sysproc.c the return value of date() function is stored in int sys_date(void), it is used in syscall function which stores the return value of sys_date() in eax register i.e proc->tf->eax. The date() system call is added in syscall.h as 22nd system call. In mydate.c the date() system call is called, The date is stored in struct rtcdate* and then printed as per the given format. By adding _mydate into the UPROGS of the make file we can invoke the date system call. Now mydate represents a call function which internally calls the date() system call and prints the present date time in IST.