Operating Systems-2 : CS3523

January 2022

Programming Assignment 2 : Syscall Implementation

Report

From this programming assignment I learned many things about system calls and some basic things from xv6 .

From 1st part of the assignment I understood :

i) `syscall()` function will create pointer to the current process by invoking `myproc()` function. Which returns the pointer to the current process by disabling the interrupts .

ii) We can load the system call number from the trap frame (%eax) .

iii) We can get return value of the system call invoked from `curproc->tf->eax` as it stores the return value in this step (`curproc->tf->eax = syscalls[num]()`)in the syscall function .

By using above things I was able to print all the names of the system calls (except write for readability purpose) and their return values .

From 2nd part of the assignment I understood :

i) I learned how system calls take arguments from the user , even though they take void as argument. They take help of the following helper functions `argint, argptr , argstr .`

ii) `I` learned how `argint, argptr , argstr` these function able to pass arguments from the user . I got more idea about this from syscall.c and also i saw how sleep system call takes input argument from user using argint.

iii) Each system call has reserved one unique number (syscall.h) . And they can be refered using the system call number . (syscall.c)

I learned still many more things by doing this assignment .