Assignment Number 3:

Hamidreza Moradi “jbu904”

1. I started with a simple SYSTEM CALL to learn first the process, then added the functionality to the system call.
2. I did not put my SYSTEM CALL To the default folder so I change the Linux MAKEFILE and added the folder of my system call to compiling directory: In line with “core-y” string added my syscall folder : MyModules/assignment3
3. Made a new make file in the assignment directory: with this line: obj-y:=getproctimes.o
4. I made a “processInfo.h” header file with structures definition and “asmlinkage”: asmlinkage long sys\_getproctimes();
5. Then I created the “getproctimes.c” with required functionality. Note that for iterating through processes I used kernel function “for\_each\_process()”
6. I added my system call name and function to “nano ./arch/x86/entry/syscalls/syscall\_64.tbl” with syscall number 333.
7. Also into “./include/linux/syscalls.h”
8. Then I recompiled the kernel “sudo make -j 4 && sudo make modules\_install -j 4 && sudo make install -j 4”
9. I put the print screen of “sudo dmseg” showing my system call writes to the system log with my name.
10. I wrote one TEST.c file so it’s easier to see the results by running just once it will do all 4-different test needed.
11. The result picture of the “test.c” is in the same folder. As you can see in the results:
    1. Sleeping for 5 second gives the result of 5000202 microsecond = 5 sec
    2. The floating-point multiplication gives 10198 microseconds
    3. The floating-point addition gives 3569 which is about 1/3 of the time of multiplication
    4. The normal fork takes 82 microseconds in my system.
    5. After assigning the 1G memory to app and then accessing it to write a character into it, I called the fork as the fork will copy the page table it will take longer about 5 time (325 microsecond)

All the code and make files, print screen are in the same folder.

Thanks for you time

Hamidreza Moradi