Operating Systems
Assignment 2

SUBMITTED BY : **ANANYA KANSAL**
ROLL NO : **2019458**

Q2. SYSTEM CALL:

- The user passes the file location and name to the system call along with the process pid. The copy_from_user function allows the kernel to access the file given by the user and then write to it.
- The file is opened using the flags O_WRONLY and O_CREAT which specify the functionality that we can write to the file and enables the kernel to make a file if not present already.
- The function sprintf() stores a string and prints the process details in the file using vfs_write() which takes the string as one of the inputs along with the file pointer, the length of the string and a flag.
- printk() is used to print to the kernel log.
- find_get_pid() is used to obtain the struct pid* for a given process id.

ERROR HANDLING :

- IS_ERR detects if the pointer returned while opening the file by the function filp_open() is an invalid pointer if so PTR_ERR returns a negative value.
- If the process corresponding to the pid is a null process (pid value=199 or out of bounds) then EINVAL is used indicating that one or more parameters had illegal values.
- Perror used in test.c program if the syscall returns a negative value (not functional).

Test.c PROGRAM :

- The user program to test the system call asks for an integer input from the user which is the pid.
- The program invokes the system call by passing it the pid and the file to which we want to write to.
- If the system call fails a negative value is returned and perror is used to specify the error.

- If the system call is functional ,the user can write dmesg on the terminal which displays all messages from the kernel ring buffer i.e. the details of the process like its priorities, process state , PID number etc..