

## **Write UP**

### **OS Assignment-2 Part 2**

#### **IMPLEMENTATION AND LOGIC**

-The kernel is updated to the 5.9.1 version by first copying the .config file from /boot directory to the linux kernel folder. The commands `sudo make menuconfig`, `sudo make -j2`, `sudo make modules_install -j2` and `sudo make install -j2` were executed to update the kernel. The system is then rebooted and the new kernel is 5.9.1.

-In `arch/x86/entry/syscalls/syscall_64.tbl` a new system call is added at the number 440 called `sh_task_info`.

-In `kernel/sys.c`, `SYSCALL_DEFINE2(sh_task_info, long, pid, char*, msg)` is used to define the working of the system call. It takes 2 arguments, the pid (long) of the process and a string msg (char\*) containing the path of the directory where information is to be stored.

-The msg is copied into a buffer string using `strncpy_from_user`. A file pointer (fl) is created and the file is opened using `filp_open(fl, O_WRONLY|O_CREAT|O_APPEND, 0666)`. A `task_struct` pointer p is also initialised to NULL. A `for_each_process(p)` loop is run and if the pid of the process matches with the pid passed as argument, information is printed on the kernel and saved to a file.

-The following fields are accessed and printed : Process ID, Process name, Process state, Process flags and Process priority.

-The int/long values are converted to string using the `sprintf` function. The `sprintf` function is again used to concatenate all the data that is to be printed to the file.

-The data is written to the file using `kernel_write(fl, buff, strlen(buff), &fl->f_pos)` function. The file is then closed using `filp_close(fl, NULL)`.

-In the `test.c` file, system call is invoked using `syscall()` along with the arguments which the user will enter. The error, if any is printed using `perror()` and `errno` is printed. The return status is also printed, which is 0 if successful and -1 if failed.

#### **INPUT/OUTPUT**

-The Inputs given by the user should be 1) The PID 2) The file\_path where the information about the task is to be printed.

Sample input:

Enter PID : 1

Enter filepath: /home/harmansingh/Desktop/a.txt

OR suppose we are in the Desktop directory and want to create the file there, we can also do:

Enter PID : 1

Enter filepath:a.txt

-The outputs will be as follows:

-If successful:

System call executed correctly.

System call returned 0.

The information about the task can be checked in the file and on the terminal, it can be checked using dmesg.

Sample Output:

Process ID NO. : 1

Process Name : systemd

Process State : 1

Process flags : 4194560

Process Priority : 120

The same can be seen in kernel log using dmesg and file both.

If unsuccessful:

System call didn't get executed correctly.

System call returned -1.

The process encountered an error : —Error is printed—

Error No : —errno is printed—

Nothing is printed in the file. A message saying “No such process exists with the given pid” is printed on the kernel log if the process doesn't exist.

## **ERRORS:**

-The first error is handled when a string greater than 256 byte is passed (ASSUMPTION) , -EFAULT is returned, which corresponds to errno 14, meaning BAD ADDRESS.

-The 2nd error is handled if PID<0 is passed as an argument. Here -EINVAL is returned, which corresponds to errno 22, meaning INVALID ARGUMENT.

-The 3rd error is handled when the given string path points to a directory and not a file. Or if the path is NULL. Here, PTR\_ERR(fl) is returned, which is -EISDIR, which corresponds to errno 21, meaning IS A DIRECTORY or \_EFAULT, if path is NULL.  
**SAMPLE I/O IN CASE OF ERRORS:**

Enter PID : -1

Enter filepath:abc.txt

System call did not get executed succesfully

System call returned -1.

The process encountered an error : Invalid argument

Error No. : 22

(suppose there's a directory named lol on Desktop.

Enter PID : 1

Enter filepath: /home/harmansingh/Desktop/lol

System call did not get executed succesfully

System call returned -1.

The process encountered an error : Is a directory

Error No. : 21

Note: For diff file generation, first clean is done and then distclean and then diff.txt is generated.