I have defined a syscall called sh_task_info(), which takes int value(PID) and file name as argument (file should be from the same folder where my linux-5.9.1 folder is stored).
For defining syscall I have used SYSCALL_DEFINEn macro which takes n input where the first argument is the syscall name and other arguments are parameters which you pass in your system call. Since my sycall sh_task_info takes int pid and char file name as input so my syscall looks like -

```
SYSCALL_DEFINE2( sh_task_info,int ,pid,char *,file)
```

After defining my syscall we have to add our syscall entry inside the kernel's system call table.

```
nano arch/x86/entry/syscalls/syscall_64.tbl

440     common    sh_task_info            sys_sh_task_info
```

Now inside my sh_task_info syscall definition , I first check whether a given pid process exists or not. So first get the pid_struct then task_struct of given pid , if task_struct is NULL it means given pid process does not exists , so print the ESRCH error (which is error generated when process does not exists) on the kernel log and return 1 signaling incorrect input).

Now if my process exists I print the fields of task_struct on kernel log and also store it in a buffer for writing to the file given by the user if it exists.

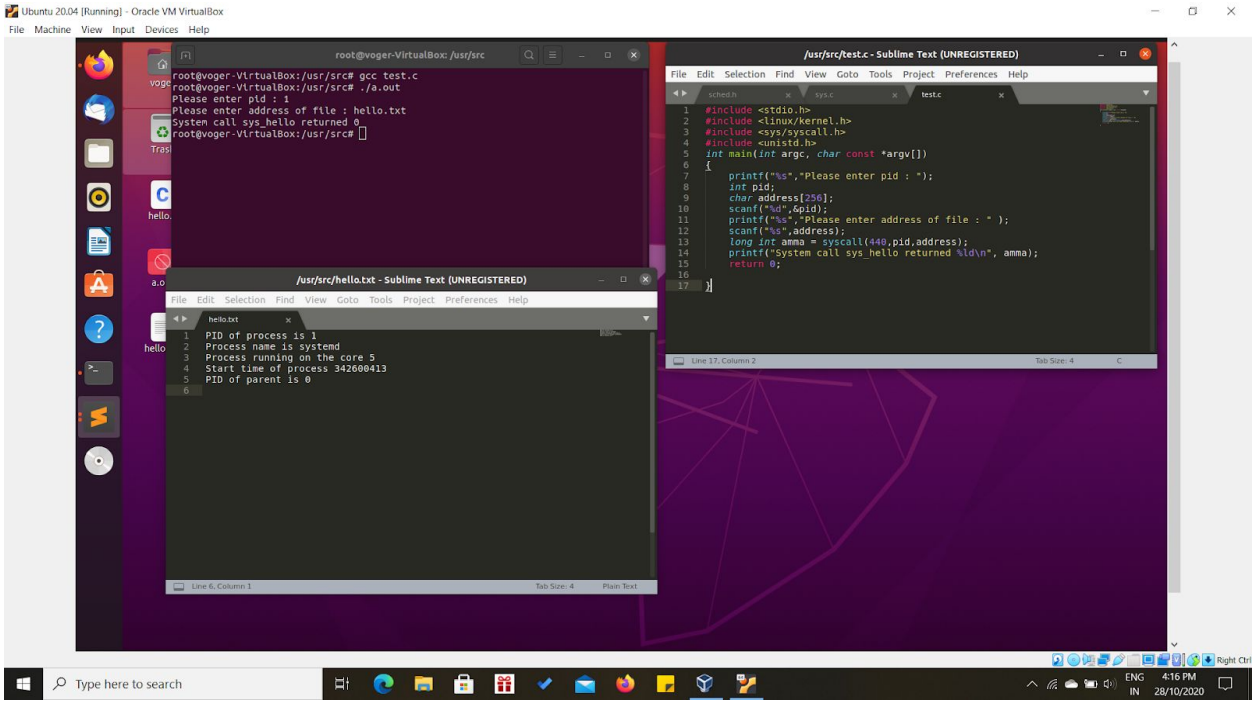
After this for writing to the file given by the user ,call the write function which first try to open the file by using flip_open which returns a file pointer . Then check if there is some error in file opening or if file exists or not.For this use IS_ERR() , which does not return error but merely returns whether the supplied pointer is an error or not. Then we use PTR_ERR() to retrieve the error value from the pointer in case IS_ERR() is true. (PTR_ERR() return -2 which means that file does not exist).

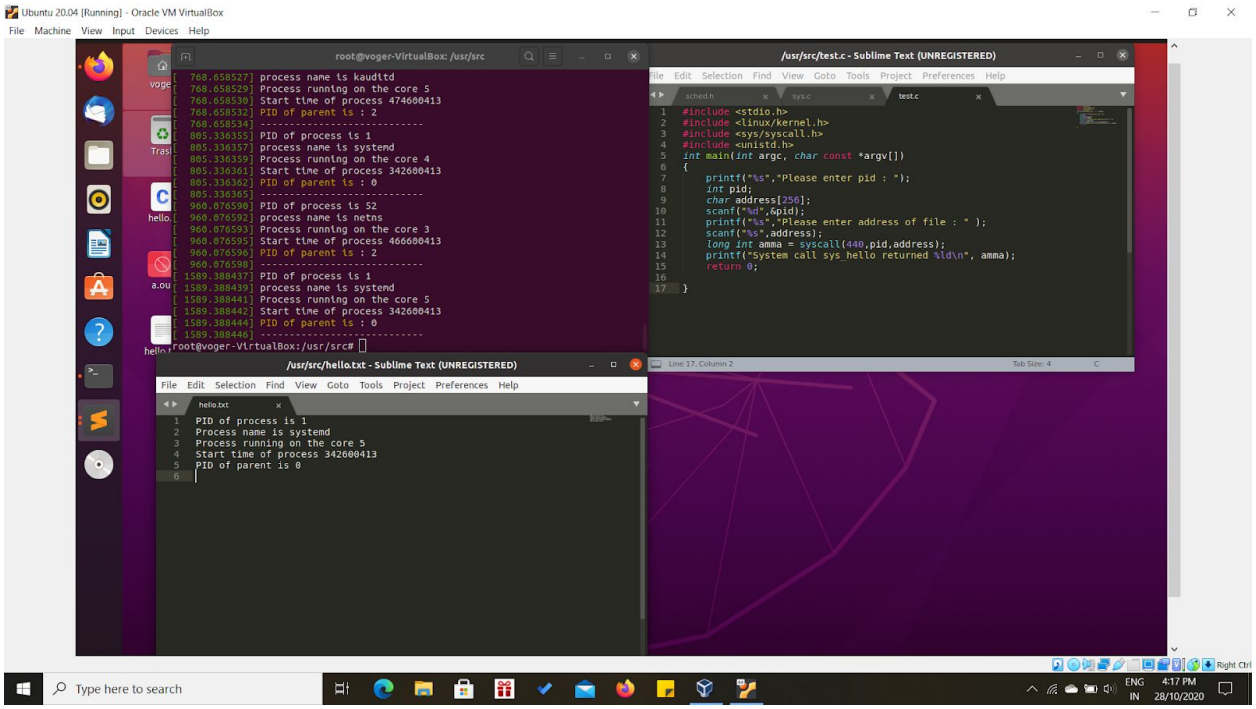
If the file exists then kernel_write() write the buffer value to the file then close the file using flip_close.

If syscall runs successfully then return 0 else if there is some error (either process not exist or file not exist) return 1 (e.g. 0 signaling correct input, while 1 signaling incorrect input).

My syscall sh_task_info() takes 2 inputs . First one is integer value for pid and second one is file name. If a user gives wrong input i.e. it gives char value for pid then my program prints no such process exists and the process will be killed and the same will be output for pid values which do not exist like -1. Second it takes the file name (which should exist in the same folder) .

If the pid and file given by user exists then it prints the task_struct fields on kernel log as well as file given by user.We can see the message on kernel log using dmesg.





Program return error values when a given file or given pid process does not exist. It will print error number 2 which is generated when file does not exist and error number 3 which means that process does not exist.