Assignment 1 - Writing your own System Call

Computer Architecture and Operating System

1. Description of Code and Implementation of System Call

- Download the kernel source using the wget command.
- Extract the kernel from the tar file, and copy the kernel directory in the directory **/usr/src/**.
- Create the directory in kernel directory with the same name of the system call (**sh_task_info**) to be implemented.
- Enter the created directory, and create three blank files of name **sh_task_info.c, task_info.h, Makefile**.
- Edit the **sh_task_info.c** file, and write the code for the system call working.
- Edit the header file **task_info.h**, and add a single line the declaration of the system call function.
- Edit the **Makefile**, and add a line of code to ensure that the hello.c file is compiled and included in the kernel source code.
- Now go back to the Linux directory, and edit the kernel **Makefile**, to instruct the compiler to tell the compiler that the source files of our new system call (sys_sh_task_info(int pid, char* filename)) are in present in the sh_task_info directory.
- Add the new system call (sys_sh_task_info(int pid, char* filename)) into the system call table (**syscall_64.tbl file**) in the **directory arch/x86/syscalls.**
- Add the new system call(sys_sh_task_info(int pid, char* filename)) in the system call header file, which is located in the file **syscall.h in directory include/linux/**.
- To configure your kernel use the following command **sudo make menuconfig**.
- Now compile the new kernel with the three parallel commands by **sudo make && sudo make modules_install && sudo make install**.
- After the compilation completion, reboot the system and test the command using the Sample C program, **task_info.c**.

2. Input for Testing

Use **gcc task_info.c -o tes**t to compile the sample C program and save the executive with the name **test**.

```
1.  ./test 1 output_file.txt
            Output stored in file  : PID Number : 1
                                     Process : init
                                     Priority : 120
                                     Process State : 120
2.  ./test -1 output_file.txt
            Output : Error : Invalid argument
                     Error no. : 22
3.  ./test abc output_file.txt
             Output : Error : Invalid argument
                      Error no. : 22
4.  ./test 1 /home/navya
             Output : Is a directory
                      Error no. : 21
```

3. Errors Handled

1. If an invalid pid such as a char or float is entered, error is thrown as ‘invalid argument error’.
2. If entered pid <=0 or greater than 32768, the function returns the invalid argument error (EINVAL).
3. If the sys_open() for creating a file if it doesn't exist, with write access returns an int less than 0, the function returns the ‘is a directory’ (EISDIR).