**How to test the system through your program**

HOW TO TEST-

1)Reboot the system

2)Write a test code in any directory.

3)In the test code we call the system call and pass 2 values as a parameter.First value is the system call number and the second call is the integer parameter for the pid number of the process.If the system call returns 0 as per the given code then the program executed well otherwise it did not execute well.

4)Type dmesg on the terminal to get the required output.

REQUIRED OUTPUT-

Based on the pid number passed as a parameter we traverse the list and display all the details of the corresponding process.Each process has a unique process ID.In this program we will get-

Process name,Process ID,Process State,Priority,RT\_Priority,Static Priority,Normal Priority.

We have submitted a diff file generated by getting the difference of the original linux file downloaded from website to the modified one which has all system call and everything.

HOW DID WE IMPLEMENT IT-

1)We created a info folder in linux-3.16.The folder had a C file containing the code to print all the details of a process, a Header file and a Makefile.

2)Then we modify the kernels Makefile to include our info folder.

3)Then we modify the syscall\_64.tbl to make a system call entry of our system call.

4)Then we modify the Syscalls.h to include the main function executing all the task of the C file.

5)Then we recompile and reboot.

DESCRIPTION OF THE CODE-

In the code first a character array was traversed to see if each character is a digit to check whether the user gave a integer PID or not.Then it converts the charm cater array to a integer.Then in my code we traverse the task\_struct list through for\_each\_process statement and then compare the pid of each process with the supplied pid in the parameter.As soon as the pid matches we print the process name,PID\_number,process sate, priority,RT\_priority,static priority and normal priority.As soon as one process’s details get printed break is executed and it gets out of the for loop.Then it checks whether while traversing the list any process’s PID got matched and whether the details of that process gets printed.If any process got printed than the flag variable is 1 so nothing happens.If nothing was printed then flag variable was 0 and which implies that the given PID did not match the PID of any process of the list task\_struct so “INVALID PID” message gets printed.

ERROR HANDLING-

1)I passed a character array or a string in the function and checked through a character array by traversing the array that if any character is not a digit.If I found a non digit character then I print that the PID is not an integer.

2)The next error handling I did was to check if a process with that pid passed as parameter exists or not by traversing the list of task\_struct.If that PID does not exist then it shows no process with the given pid exists in the list.