OS Assignment-1

Arushi Chauhan (2016019)
Raghav Sood (2016259)

`

In this assignment we have created the following system call:
sys_sh_task_info(long pid, char *filename)

Description of Function

Logical Details

Sys_sh_task_info takes two arguments ,long pid and char* filename. It searches through all the processes to find a process with matching pid. First a struct task_struct named cur_task is defined which iterates through all processes. The for_each_process loop is used for iteration and the return value of the function task_pid_nr(cur_task) gives the pid of the current process. Thus the pid of each process is matched with our input pid and if we get a match we print all details of the specified process. This is done on the console as well as in the file specified by char* filename. To print onto the file/stdout, we first append all the data to be printed in a char array :data using sprintf function,then we change the filesystem to Kernel using set_fs() (Earlier it was the user space). Using sys_open(), we open the specified file and store the output in fd variable which gives us the file descriptor number of that file (We explicitly set it to 1 in the case of printing it to stdout). Finally to print in the file we use the sys_write and vfs_write functions and close the file descriptor. To print on stdout we only need sys_write. Finally we use set_fs() again and set the file system to the old filesystem which it initially was (User Space).

Implementation Details:
Used the following functions:

1.get_fs()
2.set_fs()
3.vfs_write()
4.sys_open()
5.sys_write()
6.task_pid_nr()

Also the task_struct was used to get details about the process.

Inputs given by user:

sys_sh_task_info(long pid, char *filename)

1. long pid: An integer of long type is entered by the user which must be the pid of a process and should lie between 0 and 4194304
2. char *filename: A string indicating the path of the file is entered by the user in which the output of the system call is stored. The file will contain details of the process which matches the pid or remain empty if some error is generated.

Expected Output

If everything works fine the first line printed on the console is System Call sh_task_info returned 0
This is then followed by the details of the process as follows:

Process: string(name of process
PID: long value (pid of process)
Process State: long value (state of process)
Priority: long value (priority of process)
RT_Priority: long value (Real Time priority of process)
Static Priority: long value (static priority of process)
Normal Priority: long value (Normal priority of process)

Parent Process: string (Parent Process name if it exists)
PID Number: long value (PID of parent process if it exists)

However if we get an error,the following lines are printed on output:

System Call sh_task_info returned 1
Error generated: description of error in words (This description is based on the type of error value (errno) returned by the system call and converted into a description using strerror(errno))

Error Values and how to Interpret them

Our system call can deal with the following three errors. If any of the 3 errors is generated, the function returns a value of 1(error generated) otherwise it returns 0 (no error generated)

Errors:

1. If the pid argument is less than 0 or greater than 4194304 then error number 22(EINVAL) is returned.
2. In case the file cannot be opened, the error code returned is 2(ENOENT) i.e. no such file or directory.
3. If it cannot find a process with matching pid, t returns the value 3 which corresponds to ESRCH i.e no such process.