Aditya Verma 002342095
CS409

Project 1 Learning Experience Report

Task 1:

The objective was to create modules that could interact with the /proc filesystem to expose system metrics in user space.

Jiffies Module: The initial challenge I faced was the "bad address" error when attempting to read the /proc/jiffies file. This was a learning point that made the importance of safely transferring data between kernel and user space very apparent. Utilising the “copy_to_user” function allowed me to solve this issue. I was able to better understand the kernel's protection mechanisms against direct memory access violations.

Seconds Module: This task required me to understand how jiffies and the HZ value worked with each other in orrder to calculate the elapsed time since the module's load. Capturing the jiffies at module load and comparing it with the current jiffies count allowed me to calculate the elapsed seconds. I learned about the kernel's timing mechanisms and how they can be used to measure time intervals with precision.

Task 2:

The objective was to develop a module to display task information based on PID. This involved working with handling user inputs and retrieving process details from kernel structures.

Writing to /proc/pid: The write operation to accept a PID from user space required memory management. I used *kmalloc* to safely transfer user input and *copy_from_user* to safely transfer user input. I also made used of *kstrtoint* in order to parse the input to an integer. This reveals how the kernel handles requests when it needs to deal with user-provided data.

Reading from /proc/pid: Getting the task information meant that I needed to navigate the Linux process management infrastructure. I used *find_vpid* and *pid_task* to obtain a task's *struct task_struct* based on PID. This step provided insight into how processes are represented and managed within the kernel.

Field Name Syntax Challenge: I faced an issue while trying to access the state of the current PID, where the *tsk->state* argument was not getting recognized by the compiler. Upon looking at the header file, we were able to get the correct field name for the *struct* under *struct_task.* This was the correct reference to the current state in the *proc_read()* function and the *Makefile* was able to compile the file. The correct field name was “__state” instead of “state” and this seemed to be because of the different linux kernel versions having different notations for the field names in the headers.