This lab is about a Linux kernel module that tracks and reports the CPU usage of a processes. The module is accessed through a directory and file it creates in the proc file system. You can write the PID of a process to the status file to insert and track your process, and read from this file to see a list of all process ID’s and CPU usage

To store the data about each process, I used a linked list. Each entry in this list contains the PID of the process and its CPU usage time. Workqueues are utilized to keep an organized scheduler for my program. I have a function that goes through the list, updating or removing entries based on their current CPU usage. I use spinlocks to keep my linked list safe during it’s use.I have a kernel timer that is set up to trigger a callback function every 5 seconds. This callback is to refresh the CPU usage data.

My init function sets up the proc file, initializes the linked list, spinlock, timer, and workqueue. It schedules the timer and allocates memory for the work structure. My cleanup function handles the teardown of the module by cleaning up the linked list, removing the proc directory, deleting the timer, and destroying the workqueue. Ending up with no memory leaks.