1. The /proc/jiffies Kernel Module

Purpose:

This module creates a virtual file in the /proc file system named /proc/jiffies that shows the current value of the jiffies kernel variable when read. Jiffies is a counter that keeps track of the number of ticks since the system was booted. It’s a way of measuring time in the Linux kernel.

Key Concepts:

Proc File System: /proc is a virtual file system in Linux that provides information about system and kernel data. This module adds an entry in /proc that users can read.

Jiffies: This kernel variable is incremented by the system clock at a constant rate based on HZ, the kernel tick rate, and it provides a simple way to measure time since boot.

How It Works:

Module Initialization: When the module is loaded, it creates the /proc/jiffies file using proc_create().

Reading Jiffies: When the user reads the /proc/jiffies file the module outputs the current value of the jiffies variable.

Cleanup: When the module is unloaded, it removes the /proc/jiffies entry.

Code Overview:

We define a proc_read function to display the current jiffies value.

The module uses proc_create() to create the entry and remove_proc_entry() to clean up.

2. The /proc/seconds Kernel Module

Purpose:

This module creates a file named /proc/seconds that shows how many seconds have passed since the kernel module was loaded. It uses the jiffies counter to calculate the time elapsed in seconds.

Key Concepts:

Jiffies to Seconds: Since jiffies count ticks, and ticks are based on the kernel's HZ value we can convert jiffies to seconds using the formula: seconds=jiffiesHZ\text{seconds} = \frac{\text{jiffies}}{HZ}seconds=HZjiffies

Tracking Elapsed Time: When the module is loaded, it saves the value of jiffies at that moment. Each time /proc/seconds is read, the module subtracts this saved value from the current value of jiffies to calculate how many ticks have passed, and then converts that to seconds.

How It Works:

Module Initialization: When loaded, the module records the value of jiffies at that moment.

Reading the Seconds: When the user reads /proc/seconds, the module calculates the elapsed time in seconds since the module was loaded by comparing the current jiffies with the saved jiffies value.

Cleanup: Like the /proc/jiffies module, it removes the /proc/seconds entry when the module is unloaded.

Code Overview:

We create a proc_read function that computes the difference between the current jiffies and the jiffies recorded at load time, converting that to seconds.

The /proc/seconds file is created with proc_create(), and the entry is removed when the module is unloaded.

How the /proc File System Works in These Modules

Both modules use the /proc file system to allow users to interact with kernel space data through simple read operations. When a user reads /proc/jiffies or /proc/seconds, Linux calls the corresponding functions in the module to generate the content on the fly, rather than storing it on disk.

What We Learned:

Basic Structure of a Kernel Module:

__init function: This initializes the module when it is loaded into the kernel.

__exit function: This cleans up resources when the module is unloaded from the kernel.

Proc File System Interaction:

proc_create(): Used to create virtual files in the /proc file system.

remove_proc_entry(): Used to remove the virtual files during cleanup.

Handling Time and System Variables:

Using the jiffies variable to track time and measure intervals.

Converting jiffies to seconds by dividing by HZ.

Issues I encountered:

One issue i encountered was with the Makefile. At first I did not use ‘tab’ when writing the statements so i got an error when I attempted to use the make command in the terminal.

Another was with the creation of /proc/jiffies and /proc/seconds. It was a simple syntax error i made while writing and was able to immediately fix that with the error message i got. Something else that added on to this was after I defines PROC_NAME, things got easier to write and for me to understand