Project 1

Introduction to Linux Kernel Modules

Introduction

The project implements the `jiffies` and `seconds` kernel modules. The `jiffies` module reads the current value of the `jiffies` variable and prints it to the console when the file `/proc/jiffies` is read. The `seconds` module gives the time since the module was loaded in seconds when the file `/proc/seconds` is read.

Implementation

Jiffies Module

The `jiffies` module get the current value of the `jiffies` variable with system call `get_jiffies_64` and uses system call `copy_to_user` to copy the value string to the user space. The system will keep calling the `proc_read` function until the `read` function returns 0, which indicates the newly read data is empty.

Seconds Module

The `seconds` module stores the time when the module is loaded in a global variable `init_time`. The `proc_init` function writes the `init_time` with the current jiffies value. The `proc_read` function calculates the time difference between the current jiffies value and the `init_time` and writes the result to the user space.

Correctness

The correctness tests are as follows:

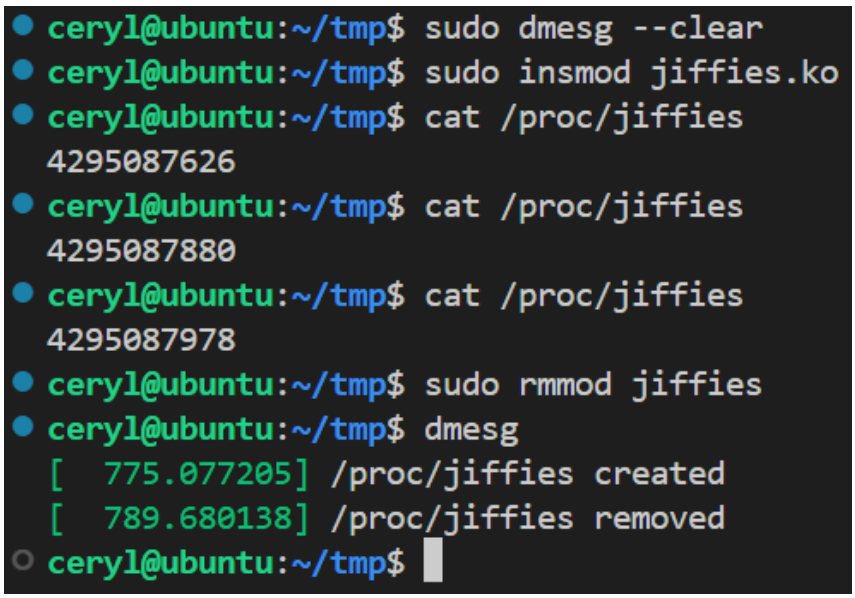

Figure 1: Jiffies Module Test

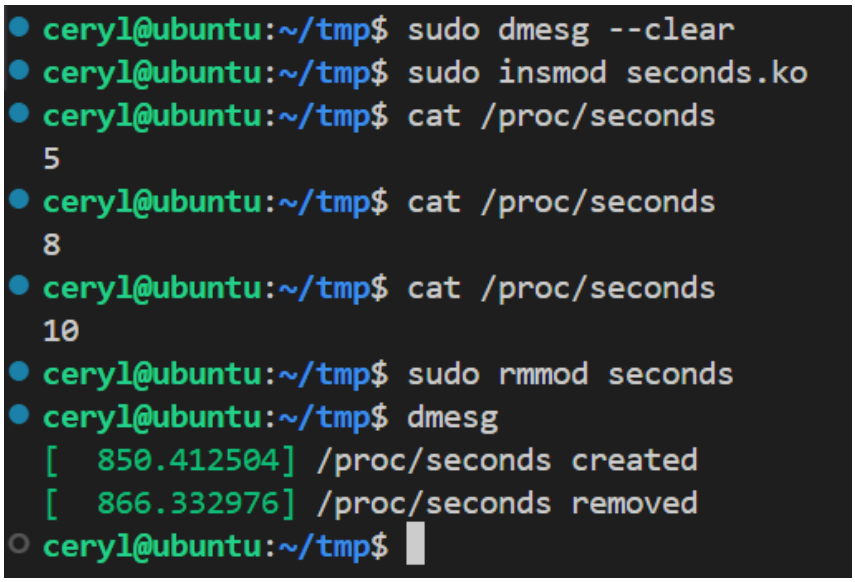

Figure 2: Seconds Module Test

Bonus

The `copy_to_user` system call was used in the kernel modules to copy the data to the user space. In the user space, `memcpy` was used to copy data. The difference between the two is that:

- `copy_to_user` checks the memory access permission.
- `copy_to_user` assures that the data is copied to the user space.
- `copy_to_user` checks the validity of the user space address that it is copying to.

Therefore, `copy_to_user` is safer than `memcpy`, and it is recommended to use `copy_to_user` in kernel modules.