In the module entry point proc_init(), we create the new `/proc/hello` entry using the `proc_create()` function. This function is passed `proc_ops`, which contains a reference to a struct file operations. This struct initializes the `.owner` and `.read` members. The value of `.read` is the name of the function `proc_read()` that is to be called whenever `/proc/hello` is read.

Examining this `proc_read()` function, we see that the string `"Hello World\n"` is written to the variable buffer where buffer exists in kernel memory. Since `/proc/hello` can be accessed from user space, we must copy the contents of buffer to user space using the kernel function `copy_to_user()`. This function copies the contents of kernel memory buffer to the variable `usr_buf`, which exists in user space.

Each time the `/proc/hello` file is read, the `proc_read()` function is called repeatedly until it returns 0, so there must be logic to ensure that this function returns 0 once it has collected the data (in this case, the string `"Hello World\n"`) that is to go into the corresponding `/proc/hello` file.

Finally, notice that the /proc/hello file is removed in the module exit point `proc_exit()` using the function `remove_proc_entry()`.

cpsc 351-Project One --- **Jiffies Project** --- due 24 Mar 2021 at 2359
Your name: Eduardo Gomez
Repository: https://github.com/ egomez3412/Jiffies-Project
Verify each of the following items and check the correct column. Incorrectly marked items will incur a 5% penalty.

Completed	Not Completed	**Jiffies Project**
✓	☐	**Modify the hello.c project, and confirm that it correctly makes (using make), and displays the GOLDEN_PRIME_RATIO on module load, and the gcd of 3300 and 24 on module removal.**
✓	☐	Create a kernel module that creates a *proc file named proc*/jiffies, and that reports the current value of jiffies when the *proc*/jiffies file is read, such as with the command: **cat proc/jiffies**
✓	☐	Be sure /proc/jiffies is removed when the module is removed
✓	☐	**Save the output of /proc/jiffies, and submit it with your project**
✓	☐	Create a kernel module that creates a proc fie named /proc/seconds, that reports the number of elapsed seconds since the kernel module was loaded, such as with the command **cat proc/seconds**
✓	☐	Your calculation involves jiffies and HZ, and includes the header files needed to compile and run your code
✓	☐	Be sure /proc/seconds is removed when the module is removed
✓	☐	**Save the output of reading from /proc/seconds, and submit it with your project**
Comments		