

# New York University

## Tandon School of Engineering

### Department of Electrical & Computer Engineering

## Introduction to Operating Systems (CS-GY6233) Spring 2022

### Assignment 2 (10 points)

In this assignment, you shall develop a very simple Linux kernel module that runs on your virtual machine.

Please consult the freely available O'Reilly book "Linux Device Drivers, 3<sup>rd</sup> Edition" (<https://lwn.net/Kernel/LDD3/>), in particular p.16, as well as your text book p.96 to get you started. Note that even though the book is written for kernel version 2.6, most mechanisms are applicable with minor or no changes. The relevant function is copied below as a starting point.

```
#include <linux/init.h>
#include <linux/module.h>
MODULE_LICENSE("Dual BSD/GPL");
static int hello_init(void)
{
    printk(KERN_ALERT "Hello, world\n");
    return 0;
}
static void hello_exit(void)
{
    printk(KERN_ALERT "Goodbye, cruel world\n");
}
module_init(hello_init);
module_exit(hello_exit);
```

The `hello_init()` function is invoked when you insert your module (using the `insmod` shell command), whereas the `hello_exit()` is called when you unload your module (using the `rmmod` shell command).

Modify this module such that:

- 1) The init function prints the tick time in milliseconds (as we define it in lecture 2) after the hello message,
- 2) The exit function prints a goodbye message and the time between the insertion and removal of the module i.e. between init and exit functions) using two different methods:
  - a. Using the difference in the value of jiffies from inserting the module to removing the module (HINT Hint: Search for "jiffies" and "HZ" in the O'Reilly book)
  - b. Using the time difference obtained by reading the timer (Hint: use `ktime_get_boottime(void)`, more documentation may be found at <https://www.kernel.org/doc/html/latest/core-api/timekeeping.html>).

Use the `dmesg` shell command to view messages printed by `printk()`.

You may use the Makefile provided in the O'Reilly book, but you may need to install the kernel headers prior to using it if not already installed:

```
sudo apt-get install linux-headers-$(uname -r)
```

## **Submission file structure:**

Please submit a **single .zip file** named **[Your Netid]\_lab#.zip**. It shall have the following structure (replace # with the actual assignment number):

- └─ [Your Netid] hw# (Single folder includes all your submissions)
  - └─ lab#\_1.c (Source code for problem 1)
  - └─ lab#\_2a.c (Source code for problem 2a, and so on)
  - └─ lab#\_1.h (Source code header file, if any)
  - └─ Makefile (makefile used to build your program, if any)
  - └─ lab#.pdf (images + Report/answers to short-answer questions)

## **What to hand in (using Brightspace):**

- Source files (.c or .h) with appropriate comments.
- Your Makefile if any.
- A .pdf file named **“lab#.pdf”** (# is replaced by the assignment number), containing:
  - Screen shot(s) of your terminal window showing the current directory, the command used to compile your program, the command used to run your program and the output of your program.

## **RULES:**

- You shall **use kernel version 4.x.x or above**. You shall not use kernel version 3.x.x.
- You may consult with other students about GENERAL concepts or methods but copying code (or code fragments) or algorithms is **NOT ALLOWED** and is considered cheating (whether copied from other students, the internet or any other source).
- If you are having trouble, please ask your teaching assistant for help.
- You must submit your assignment prior to the deadline.