Embedded Software development course

Summer 2022

Assignment 2

Task 1:

Write a simple character device driver kernel module. Print different function names when they are invoked. Also see if it appears in /dev directory:

https:/olegkutkov.me/2018/03/14/simple-linux-character-device-driver/

https:/linux-kernel-labs.github.io/refs/heads/master/labs/device_drivers.html

Task 2:

Write a simple kernel module with two .c files. The secondary .c file should contain a print function that prints your name. When the module is loaded, call the print function in the secondary .c from the init_module() from the primary .c file.

Task 3:

Repeat Task #2 but use two separate kernel modules with each have one .c file.

a. Define a function in the second module and call that function from your first module. If successful, define a function in each of the two modules. Whenever a module is loaded, it will call other module’s function if the module is loaded, else it should print "module not loaded".

b. This is module inter-dependency issue and kernel will not load the module if the other module it depends on is not loaded. How can we load the module irrespective of the other module being loaded or not and simply prints "other module not loaded" message if the other module is not loaded.

Task 4:

Declare a NULL pointer in your kernel module. Dereference that NULL pointer and observe the behavior. What system component are invoked when we issue this illegal operation?

Task 5:

Write a kernel module that creates two kthreads. Both kthreads should run for one minute before stopping. In each kthread, print thread number after every 10 seconds. We should see total of around 6 prints per thread.

Task 6:

Write a kernel module that allocates dynamic memory from kmalloc and vmalloc. What are the advantages/disadvantages of memory allocated using kmalloc vs vmalloc?

Task 7:

Write a kernel module that takes function name as input argument and prints the function name whenever that function gets called. Moreover, try to see if we can print function input arguments.

Task 8:

Read / Write from a file in kernel module.

Task 9:

Write a user space app and a kernel module, send user input data from user-space to kernel module by using IOCTL system call.

Task 10:

Do scanf in kernel module to receive user input directly in kernel module.