**Assignment: Creating Kernel Modules and Kernel Modules for Netfilter**

This assignment has been designed to gradually introduce you to the concept of Linux kernel modules, from the basics to more advanced topics like creating modules for the Netfilter subsystem. The assignment is structured in three parts: easy, intermediate, and hard.

For each assignment, Submit:

1. Your source code

2. Makefile.

3. A screenshot or text capture of your terminal showing the print messages.

**Part 1: Easy**

Task 1: Introduction to Kernel Modules

Research and Write:

* Explain what a kernel module is.

Kernel modules are pieces of code that can be loaded and unloaded into the kernel upon demand. They extend the functionality of the kernel without the need to reboot the system. A module can be configured as built-in or loadable.

* What are the advantages using kernel modules?
  + Easily Add Features: You can add new features or hardware support to the system without restarting it.
  + Save Memory: Only necessary modules are loaded, saving memory space.
  + Customize Resources: You can manage system resources based on your needs.
* What are the disadvantages using kernel modules?
  + Compatibility issues with different kernel versions.
  + Security risks and potential vulnerabilities.
  + Performance overhead during loading/unloading.
  + Complex dependency management.
* Explain the relationship between the kernel and a kernel module.

The kernel is the core component of the operating system, responsible for managing hardware, providing services, and handling system resources. Kernel modules are pieces of code that can be dynamically loaded into or removed from the running kernel to extend its functionality, add device drivers, or introduce new features without requiring a reboot.

* List some examples of real-world kernel modules.
  + Device Drivers: support for various hardware devices, such as graphics cards, network cards, sound cards, etc.
  + Network: implement various networking protocols like TCP/IP, UDP, or specific VPN protocols to enable network communication.
  + Filesystems: extend the kernel's filesystem support to handle different file systems like NTFS.
  + Bluetooth and Wi-Fi: communication with Bluetooth and Wi-Fi devices, providing wireless connectivity.

Task 2: Creating a Basic Kernel Module

Practice:

* Create a simple "Hello, World!" kernel module.
  + Your module should print "Hello, World!" when it's loaded, and "Goodbye, World!" when it's unloaded.
* Include detailed comments explaining each line of your code.
* After you've written your module, compile and load it into your kernel, then unload it.

**Part 2: Intermediate**

Task 3: Kernel Module with Parameters

Extend:

* Modify your "Hello, World!" module so that it accepts an input parameter. This parameter should be a string, which replaces the "World" in your print messages.
* If no parameter is given, your module should default to "World".

Task 4: Introduction to Netfilter

Research and Write:

* What is netfilter in the context of the Linux kernel?
* Explain the role of netfilter hooks and list the available types.
* How is Netfilter useful in the development of firewalls and packet filtering?

**Part 3: Hard**

Task 5: Kernel Module for Netfilter

Develop:

* Develop a kernel module that utilizes a netfilter hook.
* Your module should monitor incoming network traffic and log information about each packet it sees. At a minimum, it should log the source and destination IP addresses.
* Your code should be properly commented on, explaining the role of each function and important variable.
* Compile, load, and unload your module, providing screenshots or text captures of the log messages it generates.

Task 6: Advanced Netfilter Module

Challenge:

* Extend your netfilter module to filter packets based on a specific IP address. This IP should be passed as a parameter when loading the module.
* If a packet's source or destination IP matches the one passed as a parameter, your module should drop the packet and log a message saying that it did so.