

Name: Wan Muhammad Rayhan Arwindra

NPM: 1806241210

This week, we learnt about modules, module compilations, and drivers. Modules are programs or code that can be attached and detached to the kernel. We can add modules to extend the functionality of the kernel, and once it is no longer needed, we can detach it to free up memory. Since modules are programs, we need to compile it, and the result of the module's compilation is the one that would be attached and detached to the kernel. Drivers are programs which can communicate with hardware. Drivers can be statically built into the kernel, or it can also be set as a kernel module, meaning that it can be attached or detached.

The official term for this plug and play module in Linux/Unix systems is Loadable Kernel Module, or LKM for short. A module is similar to a program, but has some differences. First, a program executes the method `main()` initially, while modules execute the method `init_module` and `cleanup_module`. The module also accesses its resources directly from the kernel, while programs take it from user applications. When the module is loaded, the method `init_module` is called. On the other hand, the method `cleanup_module` is called when it is unloaded from the kernel. Since the module accesses its resources from the kernel, it cannot use the standard file descriptors such as the standard output. Because of this, we cannot use `printf` to print to the output. Instead, we use another syntax which is `printk`, which is just like `printf` but for kernel programs.

To compile a module, the steps are relatively similar to compiling a program. We first need the module file, containing the codes for the module such as the `init_module` and `cleanup_module` methods. Then, we need to compile it, either manually using `gcc` or by using a `makefile`. After compiling it, we would get two object files, one with extension of `.o`, and another with extension of `.ko`. The file we need to install the module is the `.ko` file, we do this by the syntax `"insmod [MODULE_NAME].ko"`. After this is done, we can view the module using the `lsmod` command, which would list all modules loaded into the kernel, we can also filter the modules using pipes and the `grep` command. To unload the module from the kernel, we can use the `"rmmod [MODULE_NAME]"` command, which would remove the module from the kernel and also from the `lsmod` list.

Device drivers are programs which can access and communicate with the hardware of our system. Almost all hardware in our computer has a driver for it, and it allows our applications to communicate with the hardware, not exactly how to use the hardware. The process of how an application can communicate to the hardware is through a system call. The system call then interacts with the specific driver for that device, and then fetches the data or service required. This information is then forwarded back to the application which requests it.