

Universidad Autónoma de Chihuahua  
Facultad de Ingeniería

Ingeniería en Sistemas Computacionales en  
Hardware



Damián López Verduzco - 351773

Sistemas Operativos I - 5HW1  
“Módulo de Kernel”

Asesor: M. I. Iván Miguel Chavero Jurado

## Módulo de Kernel

En sistemas Linux, un *módulo* de kernel es un programa que, como su nombre lo indica, se ejecuta en espacio de kernel: esto quiere decir que su funcionamiento se basa en el manejo directo de operaciones de kernel, opuesto al uso de APIs que programas y scripts de espacio usuario requieren. Muchas veces, se crean con el fin de permitir a la computadora utilizar cierto hardware: a esos módulos en Linux se les conocen como *drivers*.

Para codificar módulos en Linux, se utiliza el lenguaje C y *headers* especiales que se utilizan específicamente para hacer llamadas directas al kernel. A través de los comandos que esos header proveen, el programador es capaz de acceder directamente al kernel de la computadora y hacer operaciones con éste, comenzando con operaciones sencillas como la salida estándar a la bitácora del kernel. Tratándose de C, también puede generar sus propias estructuras y funciones, aunque un script de kernel básico incluye dos funciones básicas: un bloque de inicialización y uno de salida. Cuando se desea correr el programa, se utilizan las funciones *module\_init* y *module\_exit*, pasando las funciones programadas como argumentos.

Una vez desarrollado, el programador escribe un archivo *Makefile* que contiene comandos de terminal para generar los archivos de kernel a partir del archivo C. Utilizando el comando *make* en el directorio, se crean los archivos a través del Makefile. También puede crear otra serie de líneas si lo desea, como para eliminar los archivos que se crearon en primer lugar.

Para cargar el módulo que se creó, se utiliza el comando *lsmod* pasando el nombre del objeto generado (con la terminación KO, o *Kernel Object*) como argumento. El programa *journalctl* en conjunto a *grep* permite verificar que se haya inicializado el objeto de forma correcta (asumiendo que, en el bloque de inicialización, se hayan escrito líneas para pasar información a la bitácora del kernel). El comando *rmmod* quita el módulo sin más.

Si el módulo tiene alguna funcionalidad adicional a su bloque de inicialización, es necesario crear un *dispositivo* para interactuar con éste. Para ello, se utiliza *mknod* con un directorio */dev/* nuevo para el módulo, el argumento *c* (de 'create') y el número mayor del módulo (se puede programar una línea directamente en el script del módulo para enviar ese número a la bitácora del kernel). A partir de ahí, cualquier visualizador de texto puede revisar los contenidos del dispositivo nuevo, u otros programas pueden interactuar con él.