



Área Académica de Ingeniería en Computadores

Introducción a los Sistemas Embebidos

Taller 8

Profesor:

Luis Alberto Chavarría Zamora

Estudiante:

Jose Ignacio Granados Marín

Grupo 1

Semestre II 2022

## Teoría

El espacio del sistema operativo juega un papel muy importante en el diseño de controladores o drivers, para un cierto sistema computacional, debido a que la implementación puede variar en entre un espacio u otro. Dichos entornos son los siguientes:

### Espacio de usuario

Los programas de usuario final, como UNIX o aplicaciones con interfaz gráfica (GUI), forman parte del espacio de usuario de un determinado sistema operativo. Dichos sistemas de software necesitan interactuar con el hardware del sistema de una u otra manera. Sin embargo, no lo hacen directamente, sino a través de las funciones que brinda el espacio del kernel [1].

### Espacio de kernel

Adicionalmente, el sistema operativo de Linux gestiona el hardware del computador de una manera sencilla y eficiente, de tal manera que el kernel y sus respectivos drivers, forman un puente o interfaz entre el usuario final y el hardware, ofreciendo una interfaz de programación sencilla y uniforme. Además, cualquier tipo de subrutina o función que forme parte del kernel se considera parte de dicho espacio [1].

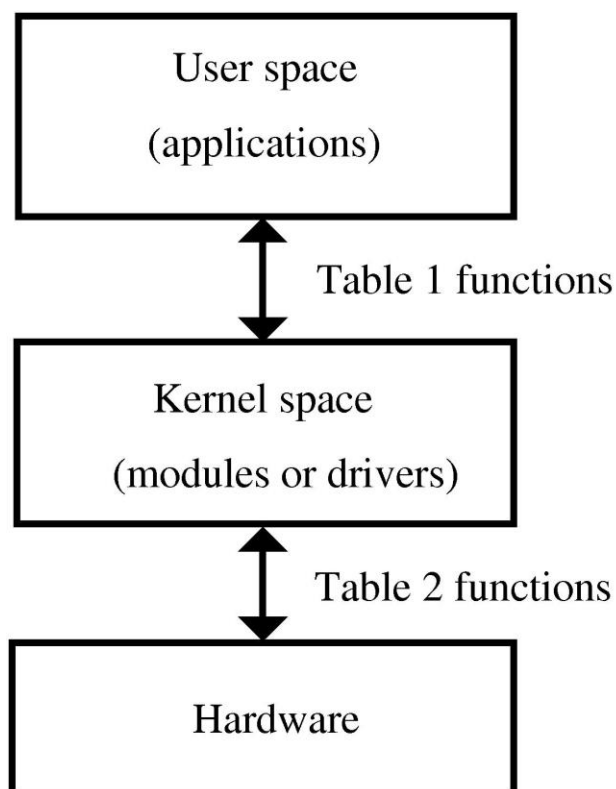


Figura 1. Distribución e interacción entre los espacios del sistema operativo y el hardware [1].

Por otra parte, el espacio de kernel ofrece varias funciones en el espacio de usuario, que permiten a los usuarios interactuar con el hardware. Dicho medio de comunicación se realiza mediante el uso de subrutinas para leer y escribir archivos. Por lo general, para cada función en el espacio de usuario, existe un equivalente en el espacio del kernel, el cual permite la transferencia de información entre ambos espacios [1]. Las principales rutinas de ambos espacios se muestran a continuación:

Eventos	Funciones de usuario	Funciones del kernel
Módulo de carga	insmod	module_init()
Abrir dispositivo	fopen	file_operations: abierto
Cerrar dispositivo	banda de rodadura	file_operations: leer
Dispositivo de escritura	fwrite	file_operations: escribir
Cerrar dispositivo	fclose	file_operations: lanzamiento
Eliminar módulo	rmmod	module_exit()

Tabla 1. Funciones o subrutinas equivalentes entre el espacio de kernel y usuario.

## Referencias

[1] *Writing device drivers in Linux: A brief tutorial*. (2006, 26 abril). Recuperado 5 de octubre de 2022, de [http://freesoftwaremagazine.com/articles/drivers\\_linux/](http://freesoftwaremagazine.com/articles/drivers_linux/)