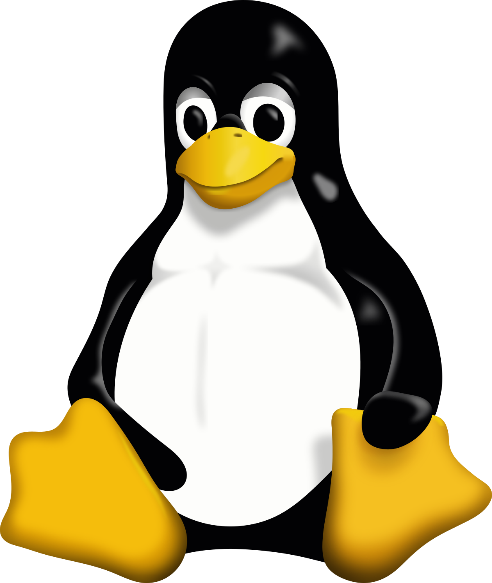


Un dibujo de una persona

Descripción generada automáticamente con confianza baja

UNIVERSIDAD AUTONOMA DE CHIHUAHUA



Integrantes:

Ashly Alessandra Guzmán Ayala 367756

Jaime Rey Mendoza 348691

Tadeo Andrei Nieto Paz 361904

* Materia: Sistemas Operativos II
* Profesor: Iván Chavero

“Módulo De Kernel”

**27/11/2024**

**Introducción**

**¿Como Funciona un Kernel De Linux?**

Funciona como el núcleo del sistema operativo, actuando como intermediario entre el hardware y las aplicaciones que se ejecutan en el sistema. Proporciona servicios esenciales y permite que múltiples procesos funcionen juntos de manera eficiente y segura.

**¿Qué es un modelo de kernel?**

Un modelo de kernel se refiere a la arquitectura o diseño conceptual que define cómo el kernel (núcleo del sistema operativo) interactúa con el hardware y el software del sistema. Cada modelo de kernel organiza las funciones principales del sistema operativo, como la gestión de procesos, memoria, dispositivos y seguridad, de manera diferente, dependiendo de sus objetivos en términos de eficiencia, flexibilidad, estabilidad y facilidad de desarrollo.

**¿Qué es un archivo de dispositivo?**

Un archivo de dispositivo es una representación especial de un dispositivo de hardware en un sistema operativo basado en Unix/Linux. Estos archivos, ubicados típicamente en el directorio /dev, permiten que las aplicaciones interactúen con el hardware de forma uniforme, utilizando las mismas operaciones que se utilizan para manejar archivos regulares. En lugar de que las aplicaciones trabajen directamente con el hardware, el sistema operativo crea un archivo de dispositivo que actúa como intermediario. Este archivo permite que las aplicaciones lean y escriban datos al hardware como si fueran operaciones de entrada/salida en un archivo regular.

**Definición de Kernel:**

Kernel es el núcleo del sistema operativo, el componente que actúa como intermediario entre el hardware y el software, gestionando los recursos del sistema. El kernel proporciona servicios básicos para las aplicaciones y gestiona recursos como la CPU, la memoria y los dispositivos de entrada/salida.

**¿Qué hace el Kernel?**

Interacción con el hardware: Controla la CPU, la memoria, los dispositivos de almacenamiento y periféricos.

* Gestión de procesos: Controla la ejecución de procesos, asignando tiempo de CPU y recursos.
* Gestión de memoria: Administra la memoria RAM, proporciona memoria virtual y gestiona la paginación.
* Entrada/Salida (I/O): Controla cómo las aplicaciones leen y escriben en dispositivos como discos, impresoras, etc.

**Componentes del Kernel de Linux:**

El kernel de Linux está compuesto por varios subcomponentes que cumplen funciones específicas:

**Gestor de procesos:** Maneja la creación, ejecución, suspensión y terminación de procesos.

**Planificador de procesos (Scheduler):** Determina qué proceso debe ejecutarse en qué momento.

**Gestión de memoria:** Administra la memoria RAM, paginación, segmentación y memoria virtual.

**Sistema de archivos:** Gestiona la estructura de archivos en el sistema, proporcionando una interfaz uniforme para interactuar con discos y dispositivos de almacenamiento.

**Controladores de dispositivos (drivers):** Permiten que el kernel se comunique con hardware como discos duros, teclados, pantallas, etc.

**Redes:** Gestiona las interfaces de red y protocolos, facilitando la comunicación de la máquina con otras.

**Interrupciones y excepciones:** Gestiona interrupciones de hardware y errores del sistema, proporcionando una forma eficiente de gestionar eventos externos.

**1°- Construcción del Módulo:**

**Dependencias de compilación**

Antes de construir el módulo, es necesario instalar las herramientas necesarias para compilar código y manejar los encabezados del kernel. Asegúrate de instalar las dependencias según tu distribución:



Este comando instala herramientas de desarrollo como GCC, make y librerías necesarias para compilar código en C.



**¿Qué hace?**  
Instala los encabezados (kernel-headers) y archivos de desarrollo (kernel-devel) del kernel en tu sistema, necesarios para compilar módulos.

**2°- Compilar El Modulo**

* Cambia al directorio del módulo:

****

**¿Qué hace?**  
Este comando te mueve al directorio donde está almacenado el código fuente del módulo.

* Compila el módulo:



**¿Qué hace?**  
make utiliza el archivo Makefile para compilar el módulo y generar el archivo binario con la extensión .ko (kernel object). Este es el archivo que se carga en el kernel.

**3°- Cargar y Probar el Módulo**

* **Cargar el módulo**

Carga el módulo en el kernel:



**¿Qué hace?**  
El comando insmod inserta el archivo del módulo (super\_module.ko) en el kernel en ejecución, permitiendo que interactúe con el sistema.

* **Verificar si el módulo está cargado**

Consulta los mensajes del kernel para confirmar que el módulo se cargó correctamente:



**¿Qué hace?**  
El comando journalctl muestra los registros del sistema. La opción -f permite verlos en tiempo real, y el comando grep kernel filtra solo los mensajes relacionados con el kernel.

* **Obtener el número mayor del dispositivo**

Obtén el número mayor del dispositivo asociado al módulo:



**¿Qué hace?**  
Busca en los mensajes del kernel información sobre el número mayor, que es un identificador único necesario para crear el archivo de dispositivo en /dev.

* **Probar el módulo**

1. Crea el archivo de dispositivo:

**¿Qué hace?**  
El comando mknod crea un archivo especial de dispositivo en /dev. Aquí:

* + **/dev/super\_module:** Es el nombre del archivo de dispositivo.
  + **c:** Indica que es un dispositivo de carácter.
  + **<device\_major\_number>: Es el número mayor que identifica al módulo.**
  + 0: Es el número menor, utilizado en dispositivos secundarios.

**4°- Prueba interactuando con el archivo de dispositivo:**



**¿Qué hace?**  
Este comando lee y muestra el contenido del archivo de dispositivo asociado al módulo.

**Eliminar el módulo**

Para retirar el módulo del kernel:



**¿Qué hace?**  
El comando rmmod elimina el módulo cargado del kernel, liberando los recursos asociados.