

**UNIVERSIDAD RAFAEL LANDÍVAR**

**FACULTAD DE INGENIERÍA**

**SISTEMAS OPERATIVOS**

**SECCIÓN 1 VESPERTINA**

**ING. JULIO REQUENA**

# **TAREA 1**

**Julio Anthony Engels Ruiz Coto 1284719**

**GUATEMALA DE LA ASUNCIÓN, ENERO 23 DE 2023**

**CAMPUS CENTRAL**

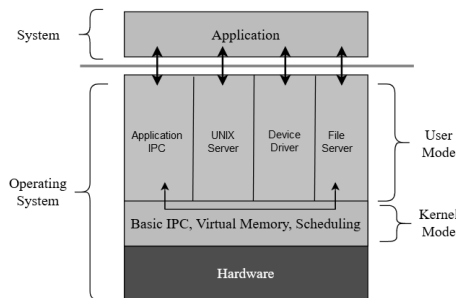
# ÍNDICE

ÍNDICE .....	2
I. INTRODUCCIÓN .....	3
II. ¿QUÉ SE ENTIENDE COMO MICROKERNEL? .....	4
I.I KERNEL .....	4
I.II DIFERENCIAS ENTRE KERNEL MONOLÍTICO Y MICROKERNEL .....	5
I.III ¿SISTEMAS OPERATIVOS EXISTENTES BASADOS EN MICROKERNEL? .....	5
III. COMENTARIO PERSONAL .....	8
IV. REFERENCIAS .....	8

# **I. INTRODUCCIÓN**

Se plantea que los microkernels solo se debe de implementar los elementos básicos o dicho de otra manera las primitivas para el apropiado funcionamiento del sistema, el resto de los elementos se recomienda ejecutarlos a nivel de aplicaciones de usuario, este análisis permite tener un sistema mas modular, flexible y sobre todo tolerante a los fallos. La principal ventaja que se observa de acuerdo con los microkernel sobre los kernels monolíticos es el despliegue del sistema, los sistemas que emplean un microkernel estos pueden agregar funcionalidades de sistemas sin verse afectados sobre el funcionamiento y sin tener la necesidad de ser modificado y recompilado, esto a su vez que las funcionalidades están implementadas como aplicaciones a nivel de usuario y estas se comunican entre si gracias al mecanismo de comunicación. Los kernels monolíticos tanto como los microkernels son dos influencias para el desarrollo de kernel para los sistemas embebidos, los sistemas embebidos se describen como sistemas computacionales los cuales dentro de ellos se guardan información relevante sobre alguna tarea en especifico lo que quiere decir que ayudan al usuario a ejecutar ciertas acciones, actividades de una manera más eficiente e inmediata. En este trabajo se exponen las diferencias entre microkernel y kernel monolítico a su vez la definición de que es un microkernel y los diferentes sistemas operativos que existen basados en microkernels.

## II. ¿QUÉ SE ENTIENDE COMO MICROKERNEL?



Una de las indispensables competencias sobre los microkernels es brindar un mecanismo de comunicaciones entre el programa cliente y los diferentes servicios que se ejecutan también desde la perspectiva del usuario. Dicha comunicación se proporciona mediante un

denominado paso de mensajes, por ejemplo, si el programa cliente desea acceder a un archivo debe interactuar con el servidor de archivos, el programa cliente y el servicio nunca interactúan directamente, sino que se comunican de una forma indirecta intercambiando dichos mensajes con el microkernel.

Los microkernel también proporcionan más seguridad y fiabilidad dado que la mayor parte de los servicios se ejecutan como procesos de usuario, en vez de ser procesos de kernel, si un servicio falla el resto del sistema operativo no se ve afectado.

Lamentablemente los microkernels pueden presentar un rendimiento peor que otras soluciones debido a la carga de procesamiento adicional impuesta por las funciones del sistema por ejemplo Windows NT, la primera versión tenía una organización de microkernel con niveles, dicha versión proporcionaba un rendimiento muy bajo comparándolo con el Windows 95. Cuando se iniciaron los preparativos para el diseño de Windows XP la arquitectura del sistema operativo era más de tipo monolítico que basada en microkernel.

### I.I KERNEL

Se describe como un componente central de un sistema operativo el cual sirve como interfaz principal que se logra comunicar tanto por el hardware físico de la computadora como de los procesos que se ejecutan en ella. Las computadoras modernas dividen la memoria en espacio de kernel y espacio de usuario. Este espacio de usuario es donde se ejecuta el software de la aplicación, por otro lado, el espacio del núcleo se dedica al trabajo detrás de escena indispensable para poder ejecutar una computadora, como la asignación de memoria y gestión de dichos procesos.

## I.II DIFERENCIAS ENTRE KERNEL MONOLÍTICO Y MICROKERNEL

Como una de las diferencias es que los microkernel son un tipo de kernel que implementa un sistema operativo proporcionando métodos, incluida la gestión del espacio de direcciones de bajo nivel, y de la gestión de subprocesos. En cambio, un kernel monolítico es un tipo de kernel en el que el sistema operativo completo se ejecuta en el espacio del kernel.

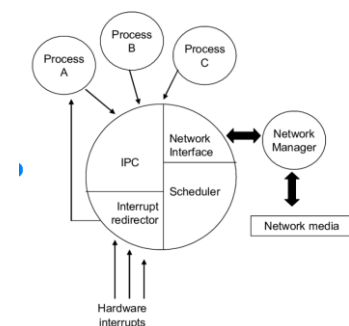
Los microkernel estos ejecutan servicios de usuario y kernel en diferentes espacios de direcciones. El kernel monolítico al contrario se encarga de ejecutar tanto el núcleo como los servicios de usuario en un espacio igual de direcciones. Cabe destacar que en los microkernels según Peter Baer Galvin & Greg Gagne (2006) señala: *“se experimentan los procesos esenciales como la gestión de memoria y programación tienen lugar en el espacio del núcleo.”*

Según Peter Baer Galvin & Greg Gagne (2006) afirman que: *“El tamaño del microkernel es más pequeño que el kernel monolítico porque solo los servicios del kernel se ejecutan en el espacio de direcciones del kernel. En cambio, el tamaño del núcleo monolítico es mayor porque los servicios del usuario y del núcleo se ejecutan en el mismo espacio de direcciones.”*

## I.III ¿SISTEMAS OPERATIVOS EXISTENTES BASADOS EN MICROKERNEL?

A continuación, se describen algunos de los SO que están basados en microkernel:

### I.III.I QNX:

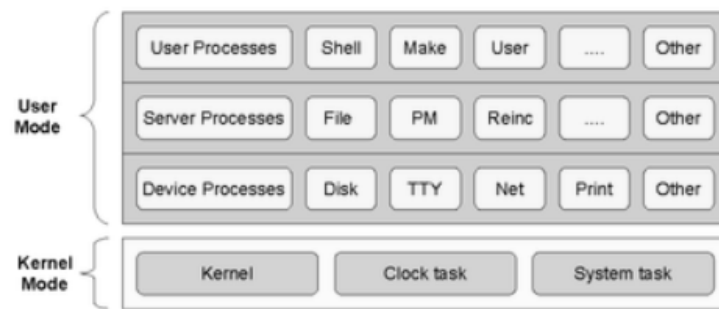


Greg Gagne (2006)

Es un sistema operativo comercial en tiempo real similar a Unix propiedad de BlackBerry, dirigido principalmente al mercado de sistemas integrados. En 1982 fue uno de los primeros sistemas operativos microkernel comercialmente exitosos. Para el no experto, el microkernel y el estilo Unix parecen estar asociados con Linux, pero ese no es el caso. Si bien QNX y las plataformas de microkernel en general comparten algunas similitudes con Linux, existen diferencias significativas.

### I.III.II MINIX:

MINIX representa un sistema operativo gratuito, de código abierto, diseñado para ser altamente confiable, flexible y seguro. Se basa en un pequeño microkernel que se ejecuta en modo kernel con el resto del sistema operativo ejecutándose como una serie de procesos aislados y protegidos en modo de usuario. Tiende a ejecutarse en arquitectura, x86 y ARM, es compatible con NetBSD y ejecuta miles de paquetes de NetBSD.

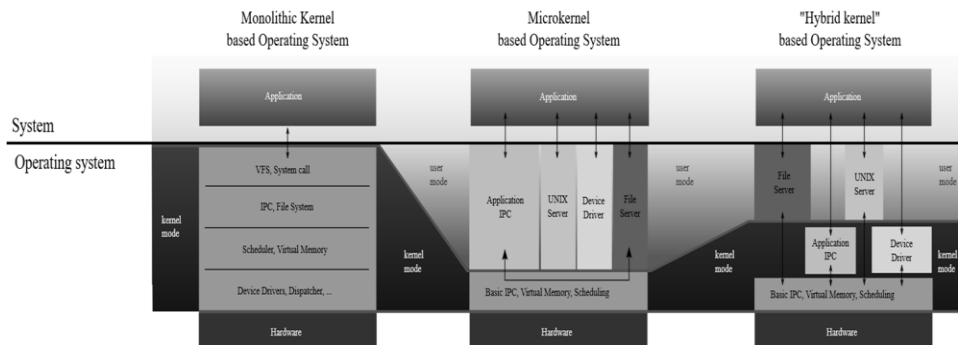


The MINIX 3 Microkernel Architecture

Greg Gagne (2006)

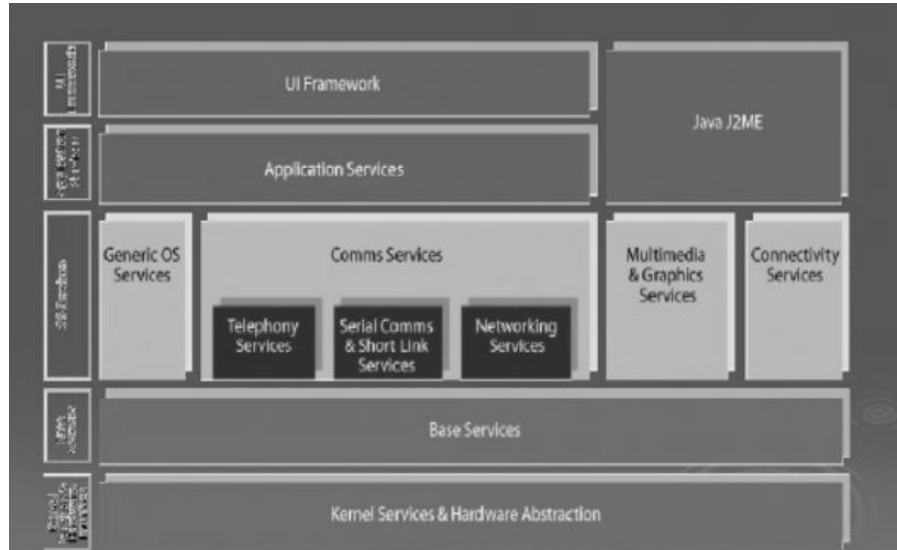
### I.III.III GNU:

GNU Hurd corresponde al reemplazo del proyecto GNU para el kernel de Unix. Es una colección de servidores que se ejecutan en el microkernel de Mach para implementar sistemas de archivos, protocolos de red, control de acceso a archivos y otras características implementadas por el kernel de Unix o kernels similares (como Linux).



Greg Gagne (2006)

### **I.III.IV SYMBIAN:**



*Greg Gagne (2006)*

Symbian contiene una arquitectura de microkernel, que lo hace diferente de los sistemas operativos competidores como Microsoft Windows y Linux. En el sistema operativo Symbian, los servicios centrales que estarían dentro del núcleo en un sistema operativo monolítico se trasladan al exterior. El uso generalizado de la arquitectura cliente-servidor y la protección del código del sistema de los clientes que se derivan de ella garantiza tanto la solidez como la alta disponibilidad de estos servicios. El objetivo es un sistema robusto que también sea receptivo y extensible; la experiencia sugiere que el diseño lo logra.

### **I.III.V CHORUS OS:**

Representa un sistema operativo en tiempo real de microkernel diseñado como un modelo informático de paso de mensajes. ChorusOS dio sus inicios como el proyecto de investigación del sistema operativo en tiempo real distribuido Chorus en el Instituto Francés de Investigación en Informática y Automatización en 1979. Durante la década de 1980, Chorus fue uno de los dos primeros microkernels con el que fue desarrollado comercialmente por la empresa emergente Chorus “Systèmes SA.” Con el paso del tiempo, el esfuerzo de desarrollo pasó de los aspectos de distribución al tiempo real para los sistemas integrados.

### **III. COMENTARIO PERSONAL**

Conforme a las fuentes que se consultaron puedo decir que los microkernels implementan un uso mas eficiente de las comodidades que nos brinda el hardware de la plataforma en donde se esta empleando, lo cual se interpreta que pueden obtener niveles mas altos de rendimiento y esto a su vez los convierte en una verdadera mejor opción para el desarrollo de los sistemas embebidos. Por otro lado, puedo mencionar que una de sus problemáticas que puede observar, que presenta este estilo arquitectónico es su alta complejidad al momento de desarrollar y también la escalabilidad, lo cual en algunas ocasiones puede ser una limitante tanto para los sistemas que deben ser diseñados conforme a su alto escalamiento. Una de las partes que me llamo mucho la atención es que el microkernel puede reconocer interrupciones, pero no las maneja lo que hace es que debe transformar esas interrupciones en mensajes.

### **IV. REFERENCIAS**

*Abraham Silberschatz, Peter Baer Galvin & Greg Gagne (2006). Fundamentos de Sistemas Operativos. 7ma Edición. Editorial Mc Graw-Hill.*