



Nombre: Dennys Alexander Pucha Carrera

Paralelo: 4to "A"

Fecha: 12/06/2023

Asignatura: Sistemas Operativos

Docente: Ing. Hernán Leonardo Torres Carrión M.Sc.

ENSAYO Nº 4

1. Tema

Arquitectura de los sistemas operativos

2. Antecedentes

Como estudiante considero que el estudio de la arquitectura de los sistemas operativos es fundamental para comprender cómo funcionan y cómo interactúan con el hardware y el software en nuestros dispositivos.

En el presente ensayo se tomará en cuenta el estudio de la arquitectura de los sistemas operativos donde se analizará los sistemas operativos más populares, centrándose principalmente en Windows, macOS, Android y Linux, para comprender cómo se diseñan y cómo influyen en la experiencia del usuario.

Al estudiar la arquitectura de estos sistemas operativos, podremos comprender cómo se gestionan los recursos del sistema, cómo se implementan los mecanismos de seguridad y cómo se interactúa con los diferentes componentes de hardware. Esta comprensión profunda nos permitirá desarrollar aplicaciones más eficientes y seguras, optimizar el rendimiento del sistema y solucionar problemas más rápidamente.

Finalmente se puede decir que el estudio de la arquitectura de los sistemas operativos nos brinda una visión más amplia de la evolución de la informática y nos ayuda a adaptarnos a los constantes avances tecnológicos.

3. Descripción

La comprensión de la definición de arquitectura es muy importante antes de comprender la arquitectura como tal de cada sistema operativo, se puede decir que la Arquitectura de un sistema operativo se refiere a:

Una arquitectura de sistema es una representación de un sistema en el que hay un mapeo de funcionalidad a componentes de hardware y software, un mapeo de arquitectura de software de hardware a arquitectura de hardware e interacción humana con esos componentes. [1]



Básicamente lo que se puede decir es que la arquitectura del sistema operativo es una representación que establece cómo se estructuran los componentes de hardware y software de un sistema, cómo se relacionan entre sí y cómo interactúan con los usuarios. Esto permite comprender y diseñar de manera efectiva el funcionamiento y las interacciones de un sistema.

Basada en esta descripción ahora si se puede establecer las arquitecturas de cada sistema operativo empezando por Windows, podemos ayudarnos de esta imagen la cual hace alusión a la estructura modular de Windows:

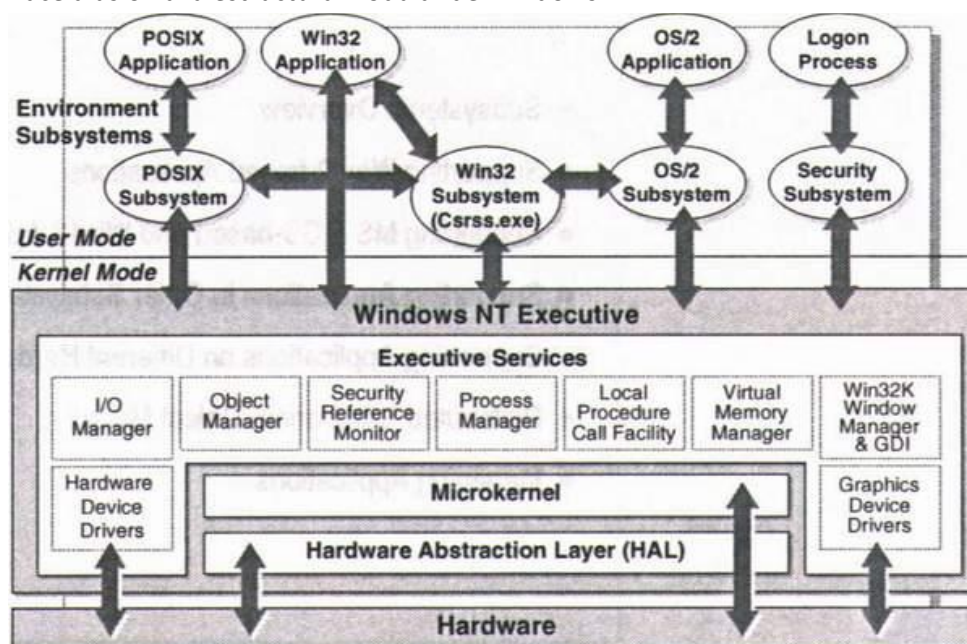


Ilustración 1:Arquitectura de Windows

Una de las características importantes de Windows y otros sistemas operativos avanzados es la división de tareas en diferentes categorías asociadas a los modos de operación soportados por los microprocesadores. Estos modos brindan diferentes niveles de privilegios a los programas, lo que les permite acceder al hardware y a otros programas en el sistema. Windows utiliza un modo privilegiado (Kernel) y un modo no privilegiado (Usuario). [2]

En la arquitectura de Windows como vemos en la imagen se puede visualizar la separación entre los distintos modos de operación que maneja Windows, empezando con el modo Kernel o privilegiado se presentan las siguientes partes:



Componente	Descripción
Capa de Abstracción de Hardware	Interfaz entre el hardware y el sistema operativo, hace que el sistema sea más portable y permite que el sistema operativo se ejecute en diferentes plataformas sin preocuparse por la plataforma específica. Proporciona la interfaz para el multiprocesamiento simétrico (SMP).
MicroKernel	Se encarga de la planificación de la ejecución de hilos, la ejecución de subprocesos, la sincronización multiprocesador y el manejo de interrupciones de hardware. Comunica con el Ejecutor de Windows a través de primitivas del sistema operativo a bajo nivel.
Ejecutor de Windows	Proporciona los fundamentos del sistema operativo a todas las aplicaciones en ejecución.
Administrador de Objetos	Proporciona información sobre el estado de los objetos al sistema operativo y se encarga de mantener cuotas para diferentes tipos de objetos y realizar la recolección de basura.
Administrador de Procesos	Responsable de crear, quitar y modificar los estados de todos los procesos e hilos. Proporciona información sobre el estado de los procesos e hilos al resto del sistema.
Administrador de Memoria Virtual	Gestiona la memoria virtual del sistema. Permite utilizar el espacio en disco en lugar de la memoria física, moviendo las páginas al disco cuando no están en uso y recuperándolas cuando se necesitan.
Servicio de Llamadas a Procedimientos Locales	Se integra al sistema y permite la comunicación entre los componentes del sistema operativo a través de llamadas a procedimientos locales. [2]

Estos componentes estarían en el modo kernel, mientras que en el modo usuario se tendrían los siguientes:

Componente	Descripción
Subsistema de Win32	Es el subsistema nativo y primario de Windows. Actúa como un servidor para los otros subsistemas y es responsable de toda la entrada y salida, controlando la pantalla, el teclado y el ratón. Proporciona compatibilidad



	con aplicaciones de DOS y Windows de 16 bits, protege al sistema y a las aplicaciones de 32 bits de la interferencia de las aplicaciones de 16 bits y DOS.
Subsistema POSIX	Implementa el estándar POSIX (Portable Operating System Interface), que proporciona portabilidad a las aplicaciones desarrolladas para plataformas UNIX.
Subsistema OS/2	Es similar al subsistema POSIX y también actúa como un subsistema de ambiente protegido. Traduce las llamadas API de OS/2 a APIs de Win32 y ofrece soporte para aplicaciones OS/2. Proporciona una interfaz gráfica y llamadas al sistema, incluyendo APIs gestores de LAN como tuberías, NETBIOS y mailslots [2]

Ahora veremos la arquitectura del sistema operativo Linux, al igual que Windows posee una arquitectura bastante similar, sin embargo, con algunas diferencias un poco sustanciales, a continuación, se visualiza una imagen donde se puede observar más a detalle esta arquitectura:



Ilustración 2:Arquitectura de Linux

Componente	Descripción
Kernel Linux	Capa de software que recubre al hardware y se comunica con él a través de controladores de dispositivos. Proporciona utilidades para la administración de memoria virtual, procesos, entrada/salida y



	red. Gestiona la planificación de procesos, los recursos de almacenamiento principal (RAM y espacio de intercambio) y las peticiones de entrada/salida con dispositivos e interfaces de red.
Sistema operativo GNU	Capa externa al kernel que incluye utilidades como editores de texto, compiladores e intérpretes de comandos. Brinda soporte a las aplicaciones de usuario a través de intérpretes de órdenes, como una shell de línea de comandos o un entorno gráfico.
Interfaz de llamadas al sistema	Elemento intermedio entre el kernel Linux y el sistema operativo GNU. Proporciona un API (POSIX.1) al sistema operativo, que son funciones del kernel accesibles a través de llamadas al sistema. Permite la comunicación y la interacción entre el sistema operativo y el kernel.
Shell de usuario	Comprende las aplicaciones de usuario, tanto las que se ejecutan en línea de comandos como las aplicaciones gráficas. [3]

Como vemos el SO Linux también se puede dividir en tres niveles principales: el nivel de usuario, el nivel del núcleo y el nivel de hardware, empezando por el nivel de usuario, en este nivel se ejecutan todos los procesos de usuario, como aplicaciones y servicios. Los procesos en este nivel no tienen privilegios para ejecutar las instrucciones más privilegiadas del procesador, luego está el nivel del núcleo el cual es el nivel privilegiado del procesador, donde se pueden ejecutar todas las instrucciones sin restricciones. Y por último el nivel de hardware el cual corresponde al hardware específico conectado al sistema, como la CPU, la memoria, los dispositivos de almacenamiento y los periféricos.[3]

La comunicación entre los procesos de usuario y el núcleo se realiza mediante la librería del sistema. Cuando un proceso de usuario necesita acceder a un servicio o recurso del núcleo, realiza una llamada al sistema utilizando una función de la librería del sistema. Esta llamada al sistema genera una interrupción software que cambia el contexto de ejecución del proceso del nivel de usuario al nivel del núcleo, permitiendo que el núcleo atienda la solicitud y devuelva los resultados al proceso de usuario.

Pasando con el sistema Operativo macOS se puede decir que este sistema es conocido por su arquitectura modular que se compone de cuatro elementos fundamentales. Estos componentes son la base sobre la cual se construye este sistema operativo. En primer lugar, tenemos el núcleo central llamado Darwin, que proporciona una base sólida y confiable gracias a la integración de tecnologías probadas como FreeBSD, Mach, Apache y gcc.



Además, el Mac OS X cuenta con un conjunto de estructuras de aplicaciones que facilitan el desarrollo de software. Entre ellas se encuentra Cocoa, una colección de estructuras orientadas a objetos que permite crear potentes interfaces gráficas de usuario con el atractivo diseño de Aqua, la interfaz de usuario característica de Apple. También está Carbon, una tecnología que simplifica la transición de aplicaciones del antiguo Mac OS 9 al entorno Mac OS X. Además, Java ofrece la posibilidad de desarrollar y ejecutar programas multiplataforma en el sistema operativo.

Al igual que en los otros sistemas operativos se tiene la siguiente imagen donde se muestra cada parte de la arquitectura del Sistema Operativo MacOS:

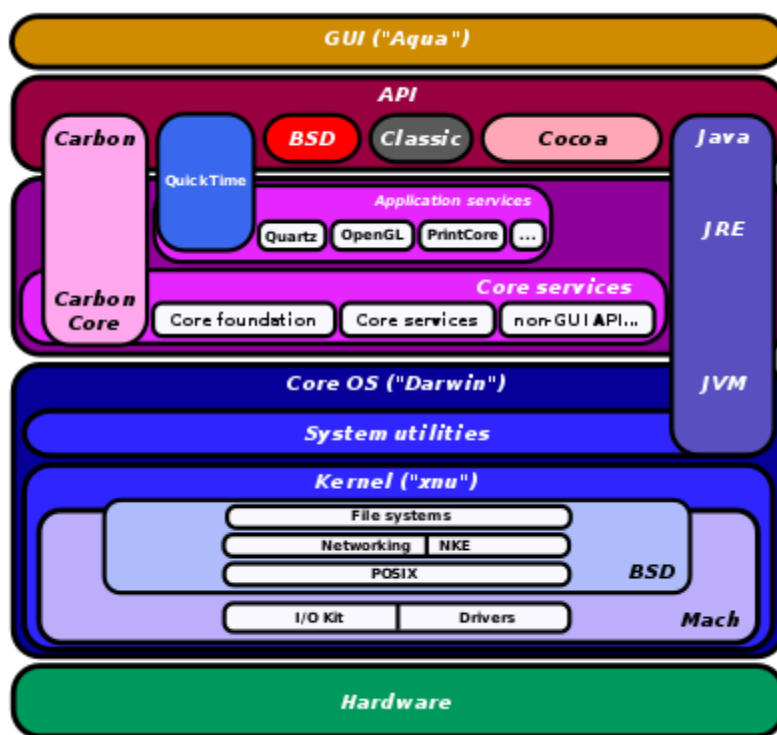


Ilustración 3:Arquitectura de macOS

Al igual que en los otros sistemas se cuenta con una tabla donde se explica cada componente del sistema:

Componente	Descripción
Aqua	Interfaz de usuario desarrollada por Apple para el Mac OS X, que utiliza colores, transparencias y animaciones para mejorar la funcionalidad y coherencia del sistema y las aplicaciones.



Estructuras de aplicaciones	Conjunto de estructuras que permiten a los desarrolladores trabajar con diferentes comunidades. Cocoa es una colección de estructuras orientadas a objetos que agiliza el desarrollo de aplicaciones, mientras que Carbon facilita la migración de aplicaciones del Mac OS 9 al Mac OS X. Java permite el desarrollo y ejecución de programas multiplataforma en Mac OS X.
Sistemas de gráficos	Compuesto por tres tecnologías basadas en estándares integradas en el sistema operativo. Quartz 2D es una biblioteca de gráficos de alto rendimiento basada en el estándar PDF de Adobe. OpenGL es el estándar para visualización 3D. QuickTime es un entorno para crear, reproducir y generar medios digitales.
Darwin	Base del sistema operativo Mac OS X, construida sobre tecnologías como FreeBSD, Mach, Apache y gcc. Es un sistema operativo completo comparable a Linux o FreeBSD, que ofrece un entorno de línea de comandos, red, bibliotecas y kernel familiar para los usuarios de UNIX.[4]

Mac OS X se construye a partir de cuatro componentes básicos: Aqua como interfaz de usuario, estructuras de aplicaciones como Cocoa, Carbon y Java, sistemas de gráficos basados en estándares como Quartz 2D, OpenGL y QuickTime, y el sistema operativo central Darwin basado en tecnologías de código abierto como FreeBSD, Mach, Apache y gcc.

Para finalizar se tiene la arquitectura del sistema operativo Android, este es un sistema operativo de código abierto que se basa en el núcleo de Linux y está diseñado para adaptarse a una amplia variedad de dispositivos y factores de forma. Su arquitectura se ha convertido en un estándar en el mundo de la tecnología móvil, ya que proporciona una base sólida y flexible para una experiencia digital en constante evolución.

La arquitectura del sistema operativo Android se caracteriza por su naturaleza modular, lo que significa que está compuesta por diferentes capas de software interconectadas, a continuación se presenta una imagen con la arquitectura del mismo:

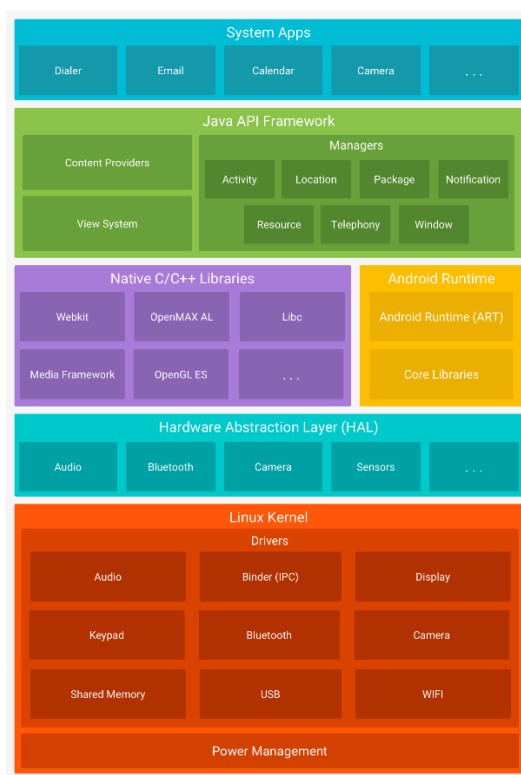


Ilustración 4:Arquitectura de Android

Explicación de cada componente:

Componente	Descripción
Kernel de Linux	Base de Android que aprovecha las funciones de seguridad y permite a los fabricantes de dispositivos desarrollar controladores de hardware para un kernel conocido.
Capa de abstracción de hardware (HAL)	Proporciona interfaces estándares que exponen las capacidades de hardware del dispositivo al marco de trabajo de la API de Java. Cada módulo de biblioteca implementa una interfaz para un componente de hardware específico.
Tiempo de ejecución de Android (ART)	Permite que cada aplicación ejecute sus propios procesos con sus propias instancias de ART.
Bibliotecas C/C++ nativas	Incluyen componentes y servicios centrales de Android, como ART y HAL. Se accede a estas bibliotecas a través de API de Java o utilizando el NDK de Android para acceder directamente desde el código nativo.



Marco de trabajo de la API de Java	Conjunto de API escritas en Java que permiten a los desarrolladores acceder a las funciones del sistema, como el sistema de vista, administrador de recursos, administrador de notificaciones y proveedores de contenido, entre otros.
Apps del sistema	Conjunto de aplicaciones preinstaladas en Android, como correo electrónico, mensajería SMS, calendarios y navegadores, que ofrecen funcionalidades clave a las que los desarrolladores pueden acceder desde sus propias aplicaciones. [5]

Basado en la descripción de esta arquitectura se puede decir que estos componentes trabajan en conjunto para brindar una plataforma sólida y flexible que permite a los desarrolladores crear aplicaciones Android con diversas funcionalidades y aprovechar las capacidades del hardware de los dispositivos.

4. Conclusiones

- Los sistemas operativos estudiados cuentan con similitudes en el manejo de modos de usuario, considero que esta es una parte principal de los sistemas de hoy en día ya que permite al sistema diferenciar y especificar que procesos y que componentes se realizaran en cada modo y garantizar seguridad para el sistema
- Linux es sin duda una base fundamental para los sistemas operativos ya que al estudiar la arquitectura de los sistemas pude constatar que tiene muchas semejanzas con los otros sistemas en especial con el sistema Android.
- Cada sistema maneja los procesos de una forma distinta pero que se asemeja un poco entre cada sistema, además se pudo constatar que estos sistemas manejan una estructura modular lo que permite manejar mejor errores o problemas que puedan existir al momento que trabaja el sistema operativo.

5. Bibliografía

[1] "¿Cuáles son las arquitecturas del sistema operativo?," Pregúntame. [En línea].

Disponible en: <https://preguntame.es/cuales-son-las-arquitecturas-del-sistema-operativo/>. [Accedido: 10-Jun-2022].

[2] "Arquitectura de Windows," Monografias.com. [En línea]. Disponible en:

<https://www.monografias.com/trabajos26/arquitectura-windows/arquitectura-windows>. [Accedido:10 -Jun-2022].

[3] D. Córdoba, "GNU/Linux: Arquitectura básica del sistema," Junco TIC, 12-Jul-2014. [En línea]. Disponible en: <https://juncotic.com/gnulinix-arquitectura-basica-del-sistema/>.

[Accedido:11 -Jun-2022].



UNIVERSIDAD NACIONAL DE LOJA
FACULTAD DE ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES
CARRERA DE COMPUTACIÓN

CARRERA DE COMPUTACIÓN



HE-CIS-2022

[4] "Arquitectura Del Mac OS X," Scribd. [En línea]. Disponible en:
<https://es.scribd.com/doc/75512104/Arquitectura-Del-Mac-OS-X>. [Accedido:11 -Jun-2022].

[5] "Platform architecture," Android Developers. [En línea]. Disponible en:
<https://developer.android.com/guide/platform?hl=es-419>. [Accedido:12 -Jun-2022].