Tema 2

Sistemas Operativos

Características de los Sistemas Operativos

Índice

Pág.

2.1. Estructura y Funciones de los sistemas operativos

Antes de especificar sobre las funciones y estructuras de los Sistemas Operativos, a continuación, se presenta una lista de características importantes del sistema operativo:

- Modo protegido y supervisor
- Permite el acceso al disco y los sistemas de archivos Controladores de dispositivos Redes Seguridad
- Ejecución del programa
- Gestión de memoria Memoria virtual Multitarea
- Manejo de operaciones de E / S
- Manipulación del sistema de archivos
- Detección y manejo de errores
- Asignación de recursos
- Protección de la información y los recursos

2.1.1. Funciones del sistema operativo

En un sistema operativo, el software realiza cada una de las funciones:

Gestión de procesos : - La gestión de procesos ayuda al sistema operativo a crear y eliminar procesos. También proporciona mecanismos de sincronización y comunicación entre procesos.

Gestión de memoria: - El módulo de gestión de memoria realiza la tarea de asignación y desasignación de espacio de memoria a los programas que necesitan estos recursos.

Gestión de archivos: gestiona todas las actividades relacionadas con los archivos, como el almacenamiento, la recuperación, la asignación de nombres, el uso compartido y la protección de archivos de la organización.

Gestión de dispositivos: la gestión de dispositivos realiza un seguimiento de todos los dispositivos. Este módulo también responsable de esta tarea se conoce como controlador de E / S. También realiza la tarea de asignación y desasignación de los dispositivos.

Gestión del sistema de E / S: Uno de los principales objetos de cualquier SO es ocultar al usuario las peculiaridades de esos dispositivos de hardware.

Administración de almacenamiento secundario: los sistemas tienen varios niveles de almacenamiento que incluyen almacenamiento primario, almacenamiento secundario y almacenamiento en caché. Las instrucciones y los datos deben almacenarse en el almacenamiento principal o en la caché para que un programa en ejecución pueda hacer referencia a ellos.

Seguridad: El módulo de seguridad protege los datos y la información de un sistema informático contra amenazas de malware y acceso autorizado.

Interpretación de los comandos : este módulo está interpretando los comandos dados por los recursos del sistema y actuando para procesarlos.

Redes: un sistema distribuido es un grupo de procesadores que no comparten memoria, dispositivos de hardware o un reloj. Los procesadores se comunican entre sí a través de la red.

Contabilidad de trabajos: realizar un seguimiento del tiempo y los recursos utilizados por varios trabajos y usuarios.

Gestión de la comunicación: Coordinación y asignación de compiladores, intérpretes y otro recurso software de los distintos usuarios de los sistemas informáticos.

2.1.2. Estructura del Sistema Operativo

Una o más CPU, los controladores de los dispositivos se conectan a través de un bus común que proporciona acceso a la memoria compartida, el Sistema Operativo presenta el siguiente funcionamiento sobre la estructura general (Figura 1).

Ejecución simultánea de CPU y dispositivos que compiten por los ciclos de memoria

- Los dispositivos de E/S y la CPU pueden ejecutar simultáneamente
- Cada controlador de dispositivo está a cargo de un tipo de dispositivo particular
- Cada controlador de dispositivo tiene un búfer local

- La CPU mueve los datos desde/hacia la memoria principal a/de los buffers locales
- La E/S es del dispositivo al buffer local del controlador
- El controlador del dispositivo informa a la CPU que ha terminado su operación causando una interrupción
- La interrupción transfiere el control a la rutina del servicio de interrupción en general, a través del vector de interrupción, que contiene las direcciones de todas las rutinas de servicio
- La arquitectura de la interrupción debe guardar la dirección de la instrucción interrumpida
- Una trampa o excepción es una interrupción generada por el software, causada por un error o una petición del usuario
- Un sistema operativo es impulsado por la interrupción
- El sistema operativo preserva el estado de la CPU almacenando registros y el contador del programa.

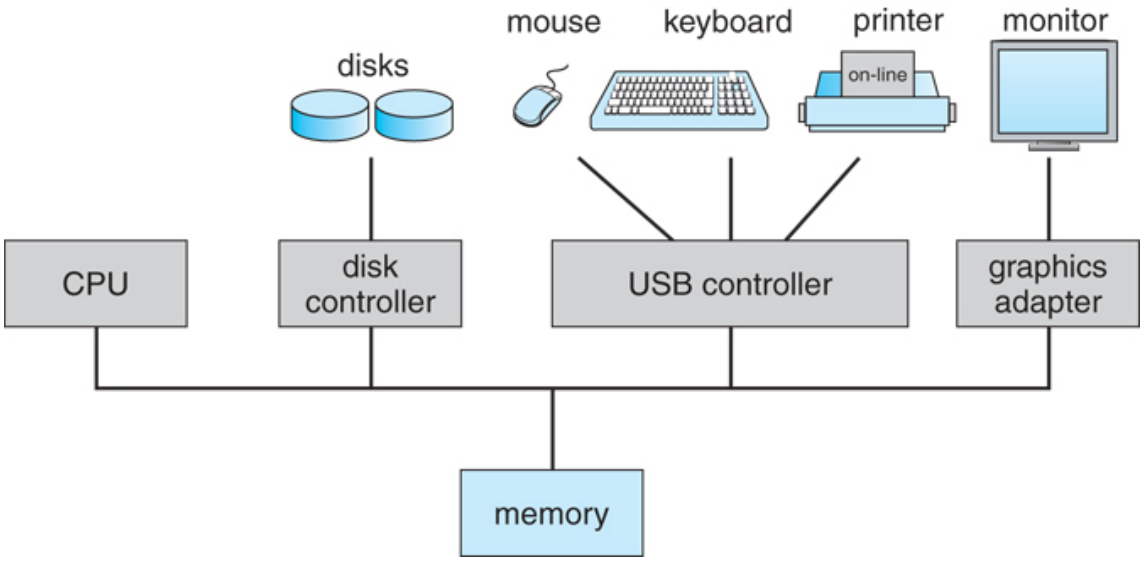


Figura 1. Estructura del Sistema Operativo.

Estructura entrada / salida

- Después de que la E/S se inicia, el control vuelve al programa de usuario sólo cuando se completa la E/S.

- Espere la instrucción que deja inactiva la CPU hasta la próxima interrupción.
- Bucle de espera (contención para el acceso a la memoria)
- A lo sumo una solicitud de E/S está pendiente a la vez, no hay un procesamiento de E/S simultáneo

- Después de que la E/S se inicia, el control vuelve al programa de usuario sin esperar a que se complete la E/S.
  - **Llamada de sistema** - solicitud al sistema operativo para permitir al usuario esperar a que se complete la E/S
  - **La tabla de estado de los dispositivos** contiene una entrada para cada dispositivo de E/S que indica su tipo, dirección y estado
  - El sistema operativo se indexa en la tabla de dispositivos de E/S para determinar el estado del dispositivo y modificar la entrada de la tabla para incluir la interrupción

Estructura de almacenamiento:

La estructura de almacenamiento se encuentra dividida en memoria principal, almacenamiento secundario, discos duros, discos de estado sólidos.

Memoria principal - sólo los grandes medios de almacenamiento a los que la CPU puede acceder directamente

- Acceso aleatorio
- Típicamente volátil

Almacenamiento secundario - extensión de la memoria principal que proporciona una gran capacidad de almacenamiento no volátil

Discos duros - platos rígidos de metal o vidrio cubiertos con material de grabación magnética. La superficie del disco se divide lógicamente en pistas, que se subdividen en sectores. El controlador del disco determina la interacción lógica entre el dispositivo y la computadora

Discos de estado sólido - más rápidos que los discos duros, no volátiles

- Varias tecnologías
- Se está volviendo más popular
- Sistemas de almacenamiento organizados en jerarquía

- Velocidad
- Costo
- Volatilidad

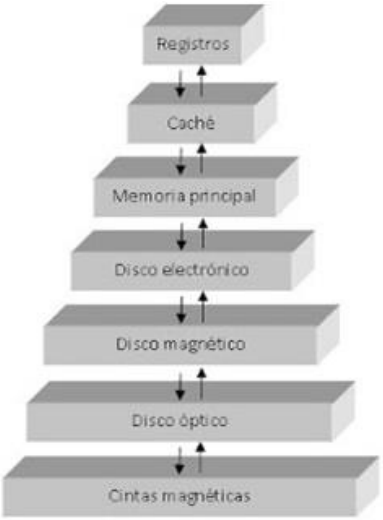


Figura 2. Estructura del disco

Caching - copiar la información en un sistema de almacenamiento más rápido; la memoria principal puede ser vista como un cache para almacenamiento secundario. Información en uso copiada de un almacenamiento más lento a otro más rápido temporalmente

Un almacenamiento más rápido (caché) comprobado primero para determinar si la información está ahí.

- Si es así, la información utilizada directamente del caché (rápido).
- De no ser así, los datos copiados a la memoria caché y utilizados allí.

Caché más pequeño que el almacenamiento que se está almacenando

- El manejo de la caché es un importante problema de diseño
- Tamaño del caché y política de reemplazo

Controlador de dispositivos para cada controlador de dispositivos para gestionar las E/S

- Proporciona una interfaz uniforme entre el controlador y el núcleo.

Estructura de la Memoria

Se utiliza para dispositivos de E/S de alta velocidad capaces de transmitir información a velocidades cercanas a la de la memoria.

El controlador del dispositivo transfiere bloques de datos del almacenamiento en el buffer directamente a la memoria principal sin la intervención de la CPU.

Sólo se genera una interrupción por bloque, en lugar de la única interrupción por byte.

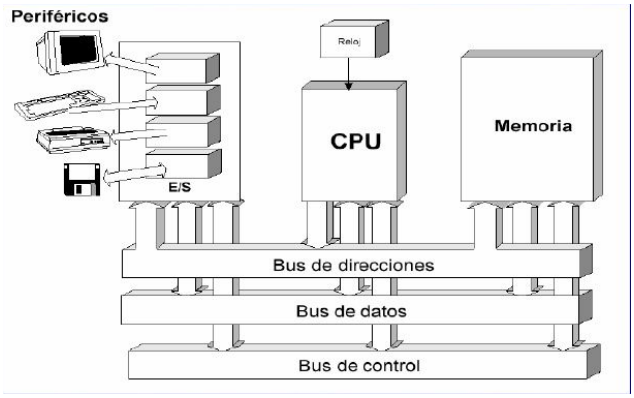


Figura 3. Estructura de la Memoria.

2.2. Arquitectura de los Sistemas Operativos

Los componentes de software centrales de un sistema operativo se conocen colectivamente como *kernel* . El kernel tiene acceso ilimitado a todos los recursos del sistema. En los primeros *sistemas monolíticos* , cada componente del sistema operativo estaba contenido dentro del kernel, podía comunicarse directamente con cualquier otro componente y tenía acceso al sistema sin restricciones. Si bien esto hizo que el sistema operativo fuera muy eficiente, también significaba que los errores eran más difíciles de aislar y había un alto riesgo de daños debido a códigos erróneos o maliciosos.

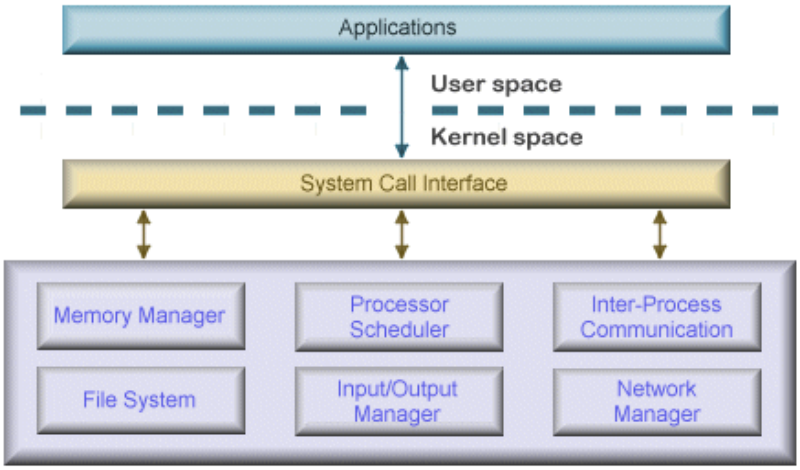


Figura 4. Una arquitectura de SO monolítica.

A medida que los sistemas operativos se volvieron más grandes y complejos, este enfoque se abandonó en gran medida en favor de un enfoque modular que agrupaba componentes con funcionalidad similar en capas para ayudar a los diseñadores de sistemas operativos a gestionar la complejidad del sistema. En este tipo de arquitectura, cada capa se comunica solo con las capas inmediatamente por encima y por debajo de ella, y las capas de nivel inferior brindan servicios a las de nivel superior mediante una interfaz que oculta su implementación.

La modularidad de los sistemas operativos en capas permite modificar la implementación de cada capa sin necesidad de modificar las capas adyacentes. Aunque este enfoque modular impone estructura y consistencia en el sistema operativo, simplificando la depuración y la modificación, una solicitud de servicio de un proceso de usuario puede pasar a través de muchas capas de software del sistema antes de recibir servicio y el rendimiento se compara desfavorablemente con el de un kernel monolítico. Además, debido a que todas las capas aún tienen acceso sin restricciones al sistema, el kernel aún es susceptible a códigos erróneos o maliciosos. Muchos de los sistemas operativos actuales, incluidos Microsoft Windows y Linux, implementan algún nivel de capas.

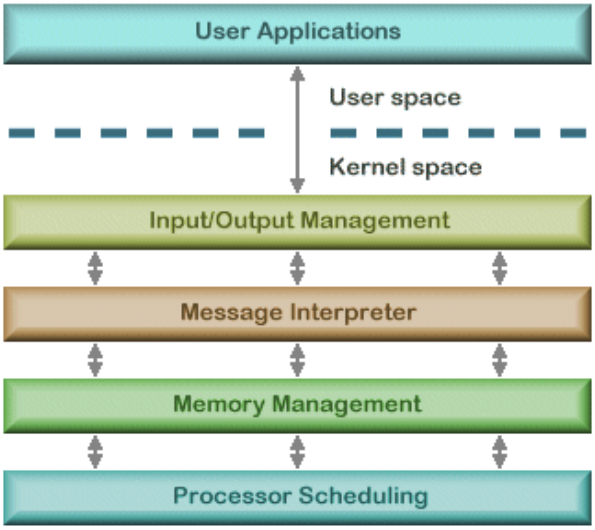


Figura 5. Arquitectura de sistema operativo en capas

Una arquitectura de *microkernel* incluye solo una pequeña cantidad de servicios dentro del kernel en un intento de mantenerlo pequeño y escalable. Los servicios generalmente incluyen administración de memoria de bajo nivel, comunicación entre procesos y sincronización básica de procesos para permitir que los procesos cooperen. En los diseños de microkernel, la mayoría de los componentes del sistema operativo, como la gestión de procesos y la gestión de dispositivos, se ejecutan fuera del kernel con un nivel más bajo de acceso al sistema.

Los microkernels son altamente modulares, lo que los hace extensibles, portátiles y escalables. Los componentes del sistema operativo fuera del kernel pueden fallar sin que el sistema operativo se caiga. Una vez más, la desventaja es un mayor nivel de comunicación entre módulos que puede degradar el rendimiento del sistema.

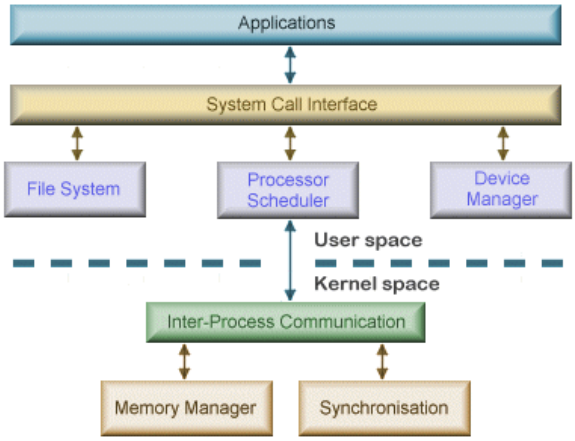


Figura 6. Arquitectura de sistema operativo microkernel

2.2.1. Arquitectura Linux

Linux es una de las versiones populares del sistema operativo UNIX. Es de código abierto ya que su código fuente está disponible gratuitamente. Es de uso gratuito. Linux fue diseñado considerando la compatibilidad con UNIX. Su lista de funciones es bastante similar a la de UNIX.

Componentes del sistema Linux

El sistema operativo Linux tiene principalmente tres componentes

Kernel : el kernel es la parte central de Linux. Es responsable de todas las actividades principales de este sistema operativo. Consta de varios módulos e interactúa directamente con el hardware subyacente. Kernel proporciona la abstracción necesaria para ocultar detalles de hardware de bajo nivel en el sistema o en los programas de aplicación.

Biblioteca del sistema: las bibliotecas del sistema son funciones o programas especiales que utilizan programas de aplicación o utilidades del sistema para acceder a las funciones del Kernel. Estas bibliotecas implementan la mayoría de las funcionalidades del sistema operativo y no requieren derechos de acceso al código del módulo del kernel.

Utilidad del sistema: los programas de utilidad del sistema son responsables de realizar tareas especializadas de nivel individual.

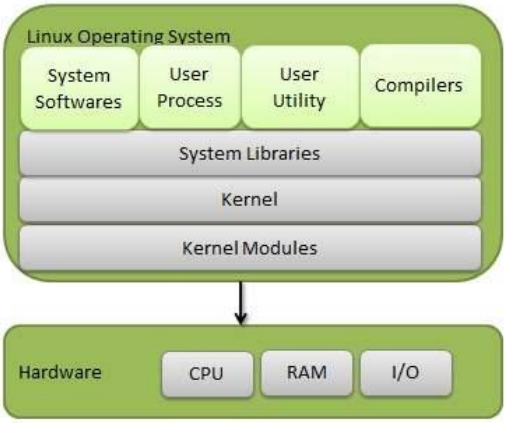


Figura 7. Componentes de Linux

Arquitectura de Linux

La siguiente ilustración muestra la arquitectura de un sistema Linux:

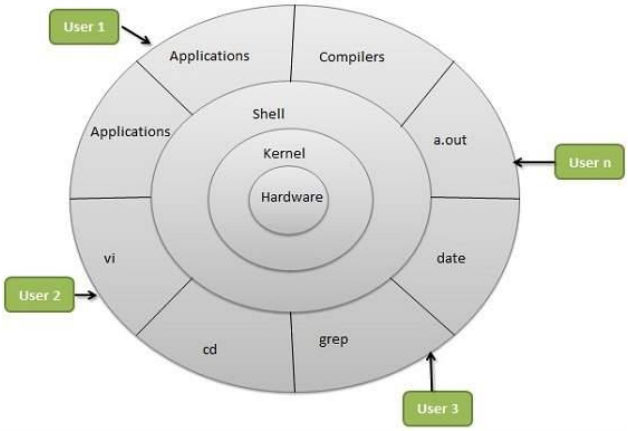


Figura 8. Arquitectura de Linux

La arquitectura de un sistema Linux consta de las siguientes capas:

Capa de hardware : el hardware consta de todos los dispositivos periféricos (RAM / HDD / CPU, etc.).

Kernel : es el componente central del sistema operativo, interactúa directamente con el hardware y proporciona servicios de bajo nivel a los componentes de la capa superior.

Shell : una interfaz para el kernel, que oculta la complejidad de las funciones del kernel a los usuarios. El shell toma los comandos del usuario y ejecuta las funciones del kernel.

Utilidades : programas de utilidad que proporcionan al usuario la mayoría de las funcionalidades de un sistema operativo.

2.2.2. Arquitectura Windows:

Con esta breve descripción general del diseño y empaquetado de Windows, echemos un vistazo a los componentes clave del sistema que conforman su arquitectura. En la Figura 2-1 se muestra una versión simplificada de esta arquitectura. Tenga en cuenta que este diagrama es básico, no muestra todo. (Por ejemplo, los componentes de red y los distintos tipos de capas de controladores de dispositivo no se muestran).

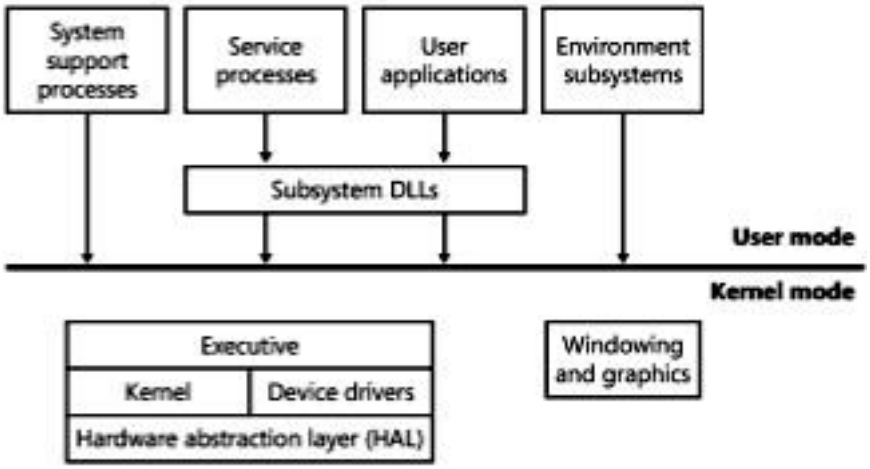


Figura 9. Arquitectura de Windows

En la Figura 9, observe primero la línea que divide las partes del sistema operativo Windows en modo de usuario y modo de núcleo. Los cuadros sobre la línea representan procesos en modo de usuario y los componentes debajo de la línea son servicios del sistema operativo en modo kernel. Los subprocesos en modo de usuario se ejecutan en un espacio de direcciones de proceso protegido (aunque mientras se ejecutan en modo kernel, tienen acceso al espacio del sistema). Por lo tanto, los procesos de soporte del sistema, los procesos de servicio, las aplicaciones de usuario y los subsistemas ambientales tienen cada uno su propio espacio de direcciones de proceso privado.

Los cuatro tipos básicos de procesos en modo usuario se describen a continuación:

• **Procesos de soporte** del sistema fijos (o cableados) , como el proceso de inicio de sesión y el administrador de sesiones, que no son servicios de Windows. (Es decir, no los inicia el administrador de control de control de servicios.

• **Procesos de servicio** que alojan servicios de Windows, como los servicios de Programador de tareas y Cola de impresión. Los servicios generalmente tienen el requisito de que se ejecuten independientemente de los inicios de sesión de los usuarios. Muchas aplicaciones de servidor de Windows, como Microsoft SQL Server y Microsoft Exchange Server, también incluyen componentes que se ejecutan como servicios.

• **Aplicaciones de usuario,** que pueden ser de uno de los siguientes tipos: 32 o 64 bits, Windows 3.1 de 16 bits, MS-DOS de 16 bits o POSIX de 32 o 64 bits. Tenga en cuenta que las aplicaciones de 16 bits solo se pueden ejecutar en Windows de 32 bits.

• **Procesos del servidor del subsistema ambiental,** que implementan parte del soporte para el entorno del sistema operativo , o personalidad, presentado al usuario y al programador. Windows NT se envió originalmente con tres subsistemas de entorno: Windows, POSIX y OS / 2. Sin embargo, los subsistemas POSIX y OS / 2 se enviaron por última vez con Windows 2000. Las ediciones Ultimate y Enterprise del cliente Windows, así como todas las versiones del servidor, incluyen soporte para un subsistema POSIX mejorado llamado Subsistema para aplicaciones basadas en Unix (SUA).

En la Figura 9, observe el cuadro "Subsistemas DLL" debajo de los cuadros "Procesos de servicio" y "Aplicaciones de usuario". En Windows, las aplicaciones de usuario no llaman directamente a los servicios del sistema operativo nativo de Windows; más bien, pasan por una o más bibliotecas de vínculos dinámicos (DLL) del subsistema . La función de las DLL del subsistema es traducir una función documentada en las llamadas de servicio del sistema nativo internas (y generalmente no documentadas)

adecuadas. Esta traducción puede implicar o no el envío de un mensaje al proceso del subsistema de entorno que está sirviendo a la aplicación del usuario.

Los componentes en modo kernel de Windows incluyen lo siguiente:

• **El ejecutivo de Windows** contiene los servicios básicos del sistema operativo, como administración de memoria, administración de procesos y subprocesos, seguridad, E / S, redes y comunicación entre procesos.

• **El kernel de Windows** consta de funciones de sistema operativo de bajo nivel, como la programación de subprocesos, el envío de interrupciones y excepciones y la sincronización de multiprocesador. También proporciona un conjunto de rutinas y objetos básicos que el resto del ejecutivo usa para implementar construcciones de nivel superior.

• **Los controladores de dispositivos** incluyen controladores de dispositivos de hardware, que traducen las llamadas de función de E / S del usuario en solicitudes de E / S de dispositivos de hardware específicos, así como controladores de dispositivos que no son de hardware, como controladores de sistema de archivos y de red.

• **La capa de abstracción de hardware (HAL)** es una capa de código que aísla el kernel, los controladores de dispositivos y el resto del ejecutivo de Windows de las diferencias de hardware específicas de la plataforma (como las diferencias entre las placas base).

• **El sistema de ventanas y gráficos** implementa las funciones de la interfaz gráfica de usuario (GUI) (más conocidas como funciones de USUARIO y GDI de Windows), como el manejo de ventanas, los controles de la interfaz de usuario y el dibujo.

2.2.3. Arquitectura Android

Android es un sistema operativo para dispositivos móviles (teléfonos inteligentes y tabletas) y es una plataforma de código abierto construida en el sistema operativo Linux. Fue desarrollado por un conglomerado de empresas de teléfonos móviles como Sony, Samsung, Intel y otras. Esta alianza abierta de teléfonos (OHA) fue liderada por Google y lanza versiones del sistema

operativo Android para ser implementadas en dispositivos móviles. La Arquitectura de Android proporciona un enfoque integrado para que los desarrolladores desarrollen aplicaciones móviles que puedan ejecutarse en cualquier dispositivo con el sistema operativo Android instalado y permite que el componente de las aplicaciones se reutilice y evite la necesidad de un redesarrollo. El código fuente de Android se ofrece en la categoría de licencias de código abierto en varios sitios web. Google aloja la mayor parte bajo la licencia Apache 2.0 y el kernel bajo la licencia pública general 2.0.

El sistema operativo Android se puede personalizar según la necesidad y, por lo tanto, podemos notar que muchos avatares de este sistema operativo se implementan en diferentes dispositivos móviles con múltiples características únicas.

Es compatible con todas las tecnologías de conectividad móvil, a saber, Wi-Fi, CDMA, GSM, NFC, Bluetooth, etc., y funcionalidades básicas como telefonía, SMS y transferencia de datos. Con esta conectividad, los datos se pueden transferir de un dispositivo a otro a través de varias aplicaciones.

Proporciona interfaces (API) que admiten servicios dependientes de la ubicación, como GPS.

La base de datos SQLite proporciona las funcionalidades de almacenamiento que necesita Android. Al ser una base de datos liviana, permite un almacenamiento más simple y una recuperación de datos más rápida.

Admite todas las versiones de archivos multimedia (audio / video) e integra micrófono, cámara, acelerómetro y altavoz para una gestión eficaz de las operaciones de grabación y reproducción.

HTML5 y CSS3 son compatibles para el desarrollo de una pantalla frontal impresionante e intuitiva.

Permite que múltiples ventanas estén activas a la vez, realizando diferentes tareas.

Se admiten gráficos 2D / 3D.

Admite la tecnología NFC que permite la conectividad entre dos dispositivos habilitados para NFC con solo tocar los dispositivos entre sí.

Compatibilidad con varios idiomas, widgets ajustables por el usuario, mensajería en la nube de Google son las otras características.

Arquitectura Android

Consiste en varios módulos de software para apoyar el funcionamiento de dispositivos móviles. Estos módulos de software contienen principalmente kernel y un conjunto de bibliotecas que facilitan el desarrollo de aplicaciones móviles y forman parte del tiempo de ejecución, el marco de la aplicación y la aplicación real.

Los módulos de la aplicación se agrupan en cinco secciones en cuatro capas diferentes.

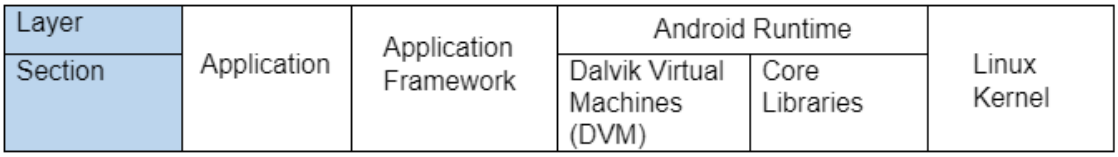


Figura 10. Arquitectura Android

La capa de tiempo de ejecución de Android tiene dos secciones, a saber, DVM y Bibliotecas, y todas las capas tienen solo una sección cada una.

1. Capa de aplicación

La capa de aplicación es la capa superior de la arquitectura y es la interfaz para los usuarios. Las aplicaciones nativas desarrolladas con la arquitectura de Android junto con aplicaciones de terceros se instalan en esta capa. Las aplicaciones de esta capa se ejecutan con la ayuda de la capa de tiempo de ejecución utilizando las clases y los servicios proporcionados por la capa de marco. Ejemplo de aplicación es Correo electrónico, Contactos, Calendario, Cámara, Hora, Música, Galería, Teléfono, SMS, Alarma, Hogar y Reloj.

2. Capa de marco de aplicaciones

La capa del marco de aplicaciones contiene las clases necesarias para desarrollar aplicaciones en la plataforma Android. Permite el acceso al hardware, maneja la interfaz de usuario y administra los recursos para la

aplicación. Los servicios proporcionados por esta capa se ponen a disposición de la capa de aplicación para su desarrollo en forma de clase. Algunos de los componentes de la capa marco son el servicio NFC, el administrador de notificaciones, el administrador de actividades, el servicio de telefonía, el administrador de paquetes y el sistema de visualización, y se utilizan en el desarrollo de aplicaciones según sea necesario.

3. Capa de tiempo de ejecución de Android

La capa de Android Runtime es una parte vital de este sistema operativo y contiene secciones como Dalvik Virtual Machine (DVM) y bibliotecas Core. Este entorno proporciona potencia básica a las aplicaciones con la ayuda de bibliotecas. La máquina virtual Dalvik explota el poder inherente básico del lenguaje Java en la gestión de la memoria y las opciones de subprocesos múltiples para proporcionar múltiples instancias al sistema operativo Android y garantizar que se ejecute de manera eficaz. Se apoya en Kernel para subprocesos y funcionalidades de nivel de SO. Esta capa proporciona los servicios de Zygote para manejar la bifurcación de nuevos procesos, el puente de depuración de Android, etc. Las bibliotecas centrales proporcionan características del lenguaje Java para el desarrollo de aplicaciones en el sistema operativo Android.

4. Capa de núcleo

La capa del núcleo es la capa más inferior e interactúa con las funcionalidades básicas del hardware con el resto de las capas del sistema operativo descritas anteriormente. Se ocupa de los controladores de unidades de visualización, cámara, Bluetooth, unidades de memoria, dispositivos de audio / video y garantiza el buen funcionamiento del dispositivo Android. Realiza una gestión centralizada de la memoria, la energía, la asignación / desasignación de recursos para el dispositivo.

Marco de la arquitectura de Android

El marco de aplicación proporciona clases de Java para el desarrollo de aplicaciones. Los desarrolladores utilizan estas clases de Java durante la codificación. Este componente proporciona los siguientes servicios.

Administrador de actividades: administra el ciclo de vida de la aplicación y realiza un seguimiento de todas las actividades.

Proveedor de contenido: facilita el intercambio de datos con aplicaciones externas.

Administrador de recursos: permite que las aplicaciones utilicen otros recursos, como configuraciones de color, interacciones del usuario y cadenas.

Administrador de notificaciones: administra alertas y notificaciones a los usuarios sobre el estado de ejecución de la aplicación.

Sistema de visualización: proporciona varias opciones de visualización para crear interacción con el usuario.

2.2.4. Arquitectura MAC OS

La arquitectura de macOS describe las capas del sistema operativo que es la culminación del proceso de investigación y desarrollo de una década de Apple Inc. para reemplazar el Mac OS clásico .

Después de los fracasos de sus intentos anteriores, Pink, que comenzó como un proyecto de Apple pero se convirtió en una empresa conjunta con IBM llamada Taligent , y Copland , que comenzó en 1994 y se canceló dos años después, Apple comenzó a desarrollar Mac OS X con la adquisición de NeXT 's NeXTSTEP en 1997.

Tenga en cuenta que Mac OS X pasó a llamarse OS X en 2012 y luego nuevamente a macOS en 2016.

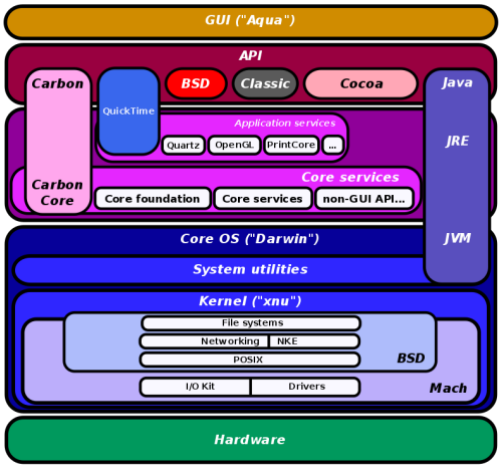


Figura 10. Arquitectura de Mac OS X

En la Conferencia Mundial de Desarrolladores de 1998 (WWDC), Apple anunció una medida que pretendía ser una respuesta a las quejas de los desarrolladores de software Macintosh que no estaban contentos con las dos opciones (Yellow Box y Blue Box) disponibles en Rhapsody. Mac OS X agregaría otra API de desarrollador a las existentes en Rhapsody. Las API clave de Macintosh Toolbox se implementarían en Mac OS X para ejecutarse directamente en las capas BSD del sistema operativo en lugar de en la capa Macintosh emulada. Esta interfaz modificada, llamada Carbon , eliminaría aproximadamente 2000 llamadas API problemáticas (de aproximadamente 8000 en total) y las reemplazaría con llamadas compatibles con un sistema operativo moderno.

En la misma conferencia, Apple anunció que el lado Mach del kernel se había actualizado con fuentes de OSFMK 7.3 (Open Source Foundation Mach Kernel) [3] y que el lado BSD del kernel se había actualizado con fuentes de FreeBSD , NetBSD. y proyectos OpenBSD . También anunciaron un nuevo modelo de controlador llamado I / O Kit, destinado a reemplazar el Driver Kit utilizado en NeXTSTEP citando la falta de capacidad de administración de energía y hot-swap del Driver Kit y su falta de capacidad de configuración automática.

En la WWDC de 1999, Apple reveló Quartz , un nuevo sistema de ventanas basado en formato de documento portátil (PDF) para el sistema operativo que no estaba gravado con tarifas de licencia para Adobe como el sistema de ventanas Display PostScript de NeXTSTEP. Apple también anunció que la capa Yellow Box había sido rebautizada como Cocoa y comenzó a alejarse de su compromiso de proporcionar Yellow Box en Windows. En esta WWDC, Apple también mostró Mac OS X arrancando desde una unidad formateada HFS Plus por primera vez.

La primera versión pública de Mac OS X lanzada a los consumidores fue una versión beta pública lanzada el 13 de septiembre de 2000.

2.3. Servicios de los Sistemas Operativos

A continuación se detallan los cinco servicios que ofrecen los sistemas operativos para la comodidad de los usuarios.

1. Ejecución de programas: El objetivo de los sistemas informáticos es permitir al usuario ejecutar programas. Por lo tanto, el sistema operativo proporciona un entorno en el que el usuario puede ejecutar programas cómodamente. La ejecución de un programa implica la asignación y desasignación de memoria, la programación de la CPU en caso de multiprocesamiento.

2. Operaciones de E/S: Cada programa requiere una entrada y produce una salida. Esto implica el uso de E/S. Así que los sistemas operativos están proporcionando E/S hace que sea conveniente para los usuarios ejecutar programas.

3. La salida de un programa puede necesitar ser escrita en nuevos archivos o la entrada tomada de algunos archivos. El sistema operativo proporciona este servicio.

4. Comunicaciones: Los procesos necesitan comunicarse entre sí para intercambiar información durante la ejecución. Puede ser entre procesos que se ejecutan en el mismo ordenador o que se ejecutan en ordenadores diferentes. Las comunicaciones pueden producirse de dos maneras: (i) memoria compartida o (ii) paso de mensajes.

5. Detección de errores: Un error en una parte del sistema puede provocar el mal funcionamiento de todo el sistema. Para evitar esta situación, el sistema operativo supervisa constantemente el sistema para detectar los

errores. Esto libera al usuario de la preocupación de que los errores se propaguen a varias partes del sistema y causen un mal funcionamiento. A continuación se enumeran los tres servicios que ofrecen los sistemas operativos para garantizar el funcionamiento eficaz del propio sistema.

- **Asignación de recursos:** Cuando varios usuarios se registran en el sistema o varios trabajos se ejecutan al mismo tiempo, es necesario asignar recursos a cada uno de ellos. El sistema operativo gestiona muchos tipos diferentes de recursos.
- **Contabilidad:** Los sistemas operativos llevan la cuenta de qué usuarios utilizan cuántos y qué tipos de recursos informáticos.Este registro puede utilizarse para la contabilidad (para poder facturar a los usuarios) o simplemente para acumular estadísticas de uso.
- **Protección:** Cuando varios procesos desarticulados se ejecutan simultáneamente, no debe ser posible que un proceso interfiera con los demás, o con el propio sistema operativo. La protección implica asegurar que todo el acceso a los recursos del sistema esté controlado. También es importante la seguridad del sistema frente a personas ajenas a él. Dicha seguridad comienza con que cada usuario tiene que autenticarse en el sistema, normalmente mediante una contraseña, para que se le permita el acceso a los recursos.

2.4. Tipos de Sistemas Operativos

A continuación se muestran los tipos populares de sistema operativo:

- Sistema operativo por lotes
- Sistema operativo multitarea / tiempo compartido
- SO multiprocesamiento
- SO en tiempo real
- SO distribuido
- SO de red
- SO móvil

2.4.1. Sistema operativo por lotes

Algunos procesos informáticos son muy largos y consumen mucho tiempo. Para acelerar el mismo proceso, un trabajo con un tipo similar de necesidades se agrupa y se ejecuta como un grupo.

El usuario de un sistema operativo por lotes nunca interactúa directamente con la computadora. En este tipo de sistema operativo, cada usuario prepara su trabajo en un dispositivo fuera de línea como una tarjeta perforada y lo envía al operador de la computadora.

2.4.2. Sistemas operativos multitarea / tiempo compartido

El sistema operativo de tiempo compartido permite a las personas ubicadas en una terminal (shell) diferente utilizar un solo sistema informático al mismo tiempo. El tiempo de procesador (CPU) que se comparte entre varios usuarios se denomina tiempo compartido.

2.4.3. SO en tiempo real

El intervalo de tiempo del sistema operativo en tiempo real para procesar y responder a las entradas es muy pequeño. Ejemplos: los sistemas de software militar, los sistemas de software espacial son el ejemplo de SO en tiempo real.

Sistema operativo distribuido

Los sistemas distribuidos utilizan muchos procesadores ubicados en diferentes máquinas para proporcionar cálculos muy rápidos a sus usuarios.

2.4.4. Sistema operativo de red

El sistema operativo de red se ejecuta en un servidor. Proporciona la capacidad de servir para administrar datos, usuarios, grupos, seguridad, aplicaciones y otras funciones de red.

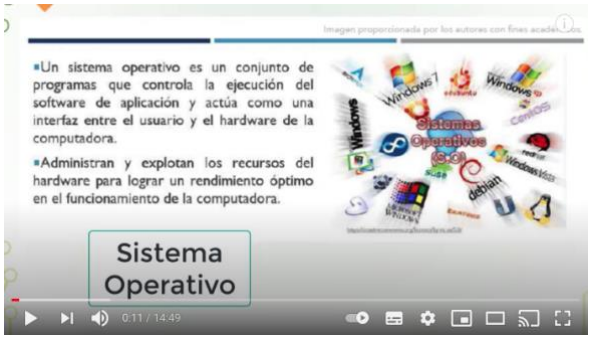
2.4.5. SO móvil

Los sistemas operativos móviles son aquellos sistemas operativos que están especialmente diseñados para alimentar teléfonos inteligentes, tabletas y dispositivos portátiles.

Algunos de los sistemas operativos móviles más famosos son Android e iOS, pero otros incluyen BlackBerry, Web y watchOS

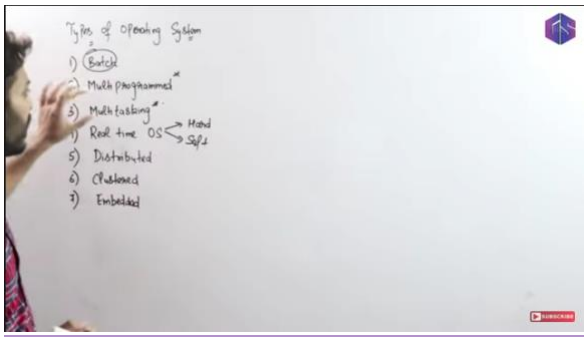
Recursos complementarios

- Video sobre tipos de sistemas operativos



https://www.youtube.com/watch?v=1C8x3nn-u_w

https://www.youtube.com/watch?v=54FGgvyKl5Y



https://www.youtube.com/watch?v=YQZbIT9FcUk

Bibliografía

Tanenbaum, Andrew S. Sistemas operativos modernos. Pearson Educación, 2003.

Stallings, W., Aguilar, L. J., Dodero, J. M., Torres, E., & Mora, M. K. (1997). Sistemas operativos (Vol. 732). Prentice Hall.