

Historia de los Sistemas Operativos.

Indice

1. Introducción
2. Tipos de Sistemas Operativos
3. Sistemas Operativos por Servicio
4. Sistemas Operativos por su Estructura
5. Sistema Operativo por la Forma de Ofrecer sus Servicios
6. Proceso
7. Dispositivos de Entrada y Salida
8. Principios del Software de E/S
9. Manejo de los Dispositivos de E/S
10. Administración de Archivos

1. Introducción

Para tratar de comprender los requisitos de un Sistema Operativo y el significado de las principales características de un Sistema Operativo contemporáneo, es útil considerar como han ido evolucionando éstos con el tiempo.

A finales de los 40's el uso de computadoras estaba restringido a aquellas empresas o instituciones que podían pagar su alto precio, y no existían los sistemas operativos. En su lugar, el programador debía tener un conocimiento y contacto profundo con el hardware, y en el infortunado caso de que su programa fallara, debía examinar los valores de los registros y paneles de luces indicadoras del estado de la computadora para determinar la causa del fallo y poder corregir su programa, además de enfrentarse nuevamente a los procedimientos de apartar tiempo del sistema y poner a punto los compiladores, ligadores, etc; para volver a correr su programa, es decir, enfrentaba el problema del procesamiento serial (serial processing).

La importancia de los sistemas operativos nace históricamente desde los 50's, cuando se hizo evidente que el operar una computadora por medio de tableros enchufables en la primera generación y luego por medio del trabajo en lote en la segunda generación se podía mejorar notoriamente, pues el operador realizaba siempre una secuencia de pasos repetitivos, lo cual es una de las características contempladas en la definición de lo que es un programa. Es decir, se comenzó a ver que las tareas mismas del operador podían plasmarse en un programa, el cual a través del tiempo y por su enorme complejidad se le llamó "Sistema Operativo". Así, tenemos entre los primeros sistemas operativos al Fortran Monitor System (FMS) e IBSYS.

Posteriormente, en la tercera generación de computadoras nace uno de los primeros sistemas operativos con la filosofía de administrar una familia de computadoras: el OS/360 de IBM. Fue este un proyecto tan novedoso y ambicioso que enfrentó por primera vez una serie de problemas conflictivos debido a que anteriormente las computadoras eran creadas para dos propósitos en general: el comercial y el científico. Así, al tratar de crear un solo sistema operativo para computadoras que podían dedicarse a un propósito, al otro o ambos, puso en evidencia la problemática del trabajo en equipos de análisis, diseño e implantación de sistemas grandes. El resultado fue un sistema del cual uno de sus mismos diseñadores patentizó su opinión en la portada de un libro: una horda de bestias prehistóricas atascadas en un foso de brea.

Surge también en la tercera generación de computadoras el concepto de la multiprogramación, porque debido al alto costo de las computadoras era necesario idear un esquema de trabajo que mantuviese a la unidad central de procesamiento más tiempo ocupada, así como el encolado (spooling) de trabajos para su lectura hacia los lugares libres de memoria o la escritura de resultados. Sin embargo, se puede afirmar que los sistemas durante la tercera generación siguieron siendo básicamente sistemas de lote.

En la cuarta generación la electrónica avanza hacia la integración a gran escala, pudiendo crear circuitos con miles de transistores en un centímetro cuadrado de silicón y ya es posible hablar de las computadoras personales y las estaciones de trabajo. Surgen los conceptos de interfaces amigables intentando así atraer al público en general al uso de las computadoras como herramientas cotidianas.

Se hacen populares el MS-DOS y UNIX en estas máquinas. También es común encontrar clones de computadoras personales y una multitud de empresas pequeñas ensamblándolas por todo el mundo.

Para mediados de los 80's, comienza el auge de las redes de computadoras y la necesidad de sistemas operativos en red y sistemas operativos distribuidos. La red mundial Internet se va haciendo accesible a toda clase de instituciones y se comienzan a dar muchas soluciones (y problemas) al querer hacer convivir recursos residentes en computadoras con sistemas operativos diferentes. Para los 90's el

paradigma de la programación orientada a objetos cobra auge, así como el manejo de objetos desde los sistemas operativos. Las aplicaciones intentan crearse para ser ejecutadas en una plataforma específica y poder ver sus resultados en la pantalla o monitor de otra diferente (por ejemplo, ejecutar una simulación en una máquina con UNIX y ver los resultados en otra con DOS). Los niveles de interacción se van haciendo cada vez más profundos.

¿Qué es un Sistema Operativo?.

Un Sistema Operativo es un programa que actúa como intermediario entre el usuario y el hardware de un computador y su propósito es proporcionar un entorno en el cual el usuario pueda ejecutar programas. El objetivo principal de un Sistema Operativo es, entonces, lograr que el Sistema de computación se use de manera cómoda, y el objetivo secundario es que el hardware del computador se emplee de manera eficiente.

Un Sistema Operativo es una parte importante de cualquier sistema de computación. Un sistema de computación puede dividirse en cuatro componentes: el hardware, el Sistema Operativo, los programas de aplicación y los usuarios. El hardware (Unidad Central de Procesamiento(UCP), memoria y dispositivos de entrada/salida (E/S)) proporciona los recursos de computación básicos. Los programas de aplicación (compiladores, sistemas de bases de datos, juegos de vídeo y programas para negocios) definen la forma en que estos recursos se emplean para resolver los problemas de computación de los usuarios.

Característica de un Sistema Operativo.

En general, se puede decir que un Sistema Operativo tiene las siguientes características:

Conveniencia. Un Sistema Operativo hace más conveniente el uso de una computadora.

Eficiencia. Un Sistema Operativo permite que los recursos de la computadora se usen de la manera más eficiente posible.

Habilidad para evolucionar. Un Sistema Operativo deberá construirse de manera que permita el desarrollo, prueba o introducción efectiva de nuevas funciones del sistema sin interferir con el servicio.

Encargado de administrar el hardware. El Sistema Operativo se encarga de manejar de una mejor manera los recursos de la computadora en cuanto a hardware se refiere, esto es, asignar a cada proceso una parte del procesador para poder compartir los recursos.

Relacionar dispositivos (gestionar a través del kernel). El Sistema Operativo se debe encargar de comunicar a los dispositivos periféricos, cuando el usuario así lo requiera.

Organizar datos para acceso rápido y seguro.

Manejar las comunicaciones en red. El Sistema Operativo permite al usuario manejar con alta facilidad todo lo referente a la instalación y uso de las redes de computadoras.

Procesamiento por bytes de flujo a través del bus de datos.

Facilitar las entradas y salidas. Un Sistema Operativo debe hacerle fácil al usuario el acceso y manejo de los dispositivos de Entrada/Salida de la computadora.

Técnicas de recuperación de errores.

Evita que otros usuarios interfieran. El Sistema Operativo evita que los usuarios se bloqueen entre ellos, informándoles si esa aplicación esta siendo ocupada por otro usuario.

Generación de estadísticas.

Permite que se puedan compartir el hardware y los datos entre los usuarios.

El software de aplicación son programas que se utilizan para diseñar, tal como el procesador de palabras, lenguajes de programación, hojas de cálculo, etc.

El software de base sirve para interactuar el usuario con la máquina, son un conjunto de programas que facilitan el ambiente plataforma, y permite el diseño del mismo.

El Software de base está compuesto por :

Cargadores.

Compiladores.

Ensambladores.

Macros.

2. Tipos de Sistemas Operativos.

Actualmente los sistemas operativos se clasifican en tres clasificaciones: sistemas operativos por su estructura (visión interna), sistemas operativos por los servicios que ofrecen y sistemas operativos por la forma en que ofrecen sus servicios (visión externa).

3. Sistemas Operativos por Servicios(Visión Externa).

Esta clasificación es la más comúnmente usada y conocida desde el punto de vista del usuario final. Esta clasificación se comprende fácilmente con el cuadro sinóptico que a continuación se muestra:

Por Número de Usuarios:

Sistema Operativo Monousuario.

Los sistemas operativos monousuarios son aquéllos que soportan a un usuario a la vez, sin importar el número de procesadores que tenga la computadora o el número de procesos o tareas que el usuario pueda ejecutar en un mismo instante de tiempo. Las computadoras personales típicamente se han clasificado en este renglón.

En otras palabras los sistemas monousuarios son aquellos que nada más puede atender a un solo usuario, gracias a las limitaciones creadas por el hardware, los programas o el tipo de aplicación que se este ejecutando.

Sistema Operativo Multiusuario.

Los sistemas operativos multiusuarios son capaces de dar servicio a más de un usuario a la vez, ya sea por medio de varias terminales conectadas a la computadora o por medio de sesiones remotas en una red de comunicaciones. No importa el número de procesadores en la máquina ni el número de procesos que cada usuario puede ejecutar simultáneamente.

En esta categoría se encuentran todos los sistemas que cumplen simultáneamente las necesidades de dos o más usuarios, que comparten mismos recursos. Este tipo de sistemas se emplean especialmente en redes. En otras palabras consiste en el fraccionamiento del tiempo (timesharing).

Por el Número de Tareas:

Sistema Operativo Monotarea.

Los sistemas monotarea son aquellos que sólo permiten una tarea a la vez por usuario. Puede darse el caso de un sistema multiusuario y monotarea, en el cual se admiten varios usuarios al mismo tiempo pero cada uno de ellos puede estar haciendo solo una tarea a la vez.

Los sistemas operativos monotareas son más primitivos y, solo pueden manejar un proceso en cada momento o que solo puede ejecutar las tareas de una en una.

Sistema Operativo Multitarea.

Un sistema operativo multitarea es aquél que le permite al usuario estar realizando varias labores al mismo tiempo.

Es el modo de funcionamiento disponible en algunos sistemas operativos, mediante el cual una computadora procesa varias tareas al mismo tiempo. Existen varios tipos de multitareas. La conmutación de contextos (context Switching) es un tipo muy simple de multitarea en el que dos o más aplicaciones se cargan al mismo tiempo, pero en el que solo se esta procesando la aplicación que se encuentra en primer plano (la que ve el usuario. En la multitarea cooperativa, la que se utiliza en el sistema operativo Macintosh, las tareas en segundo plano reciben tiempo de procesado durante los tiempos muertos de la tarea que se encuentra en primer plano (por ejemplo, cuando esta aplicación esta esperando información del usuario), y siempre que esta aplicación lo permita. En los sistemas multitarea de tiempo compartido, como OS/2, cada tarea recibe la atención del microprocesador durante una fracción de segundo.

Un sistema operativo multitarea puede estar editando el código fuente de un programa durante su depuración mientras compila otro programa, a la vez que está recibiendo correo electrónico en un proceso en background. Es común encontrar en ellos interfaces gráficas orientadas al uso de menús y el ratón, lo cual permite un rápido intercambio entre las tareas para el usuario, mejorando su productividad.

Un sistema operativo multitarea se distingue por su capacidad para soportar la ejecución concurrente de dos o más procesos activos. La multitarea se implementa generalmente manteniendo el código y los datos de varios procesos simultáneamente en memoria y multiplexando el procesador y los dispositivos de E/S entre ellos.

La multitarea suele asociarse con soporte hardware y software para protección de memoria con el fin de evitar que procesos corrompan el espacio de direcciones y el comportamiento de otros procesos residentes.

Por el Número de Procesadores:

Sistema Operativo de Uniprocreso.

Un sistema operativo uniprocésor es aquél que es capaz de manejar solamente un procesador de la computadora, de manera que si la computadora tuviese más de uno le sería inútil. El ejemplo más típico de este tipo de sistemas es el DOS y MacOS.

Sistema Operativo de Multiprocésor.

Un sistema operativo multiprocésor se refiere al número de procesadores del sistema, que es más de uno y éste es capaz de usarlos todos para distribuir su carga de trabajo. Generalmente estos sistemas trabajan de dos formas: simétrica o asimétricamente.

Asimétrica.

Cuando se trabaja de manera asimétrica, el sistema operativo selecciona a uno de los procesadores el cual jugará el papel de procesador maestro y servirá como pivote para distribuir la carga a los demás procesadores, que reciben el nombre de esclavos.

Simétrica.

Cuando se trabaja de manera simétrica, los procesos o partes de ellos (threads) son enviados indistintamente a cual quiera de los procesadores disponibles, teniendo, teóricamente, una mejor distribución y equilibrio en la carga de trabajo bajo este esquema.

Se dice que un thread es la parte activa en memoria y corriendo de un proceso, lo cual puede consistir de un área de memoria, un conjunto de registros con valores específicos, la pila y otros valores de contexto.

Un aspecto importante a considerar en estos sistemas es la forma de crear aplicaciones para aprovechar los varios procesadores. Existen aplicaciones que fueron hechas para correr en sistemas monoproceso que no toman ninguna ventaja a menos que el sistema operativo o el compilador detecte secciones de código paralelizables, los cuales son ejecutados al mismo tiempo en procesadores diferentes. Por otro lado, el programador puede modificar sus algoritmos y aprovechar por sí mismo esta facilidad, pero esta última opción las más de las veces es costosa en horas hombre y muy tediosa, obligando al programador a ocupar tanto o más tiempo a la paralelización que a elaborar el algoritmo inicial.

4. Sistemas Operativos por su Estructura (Visión Interna).

Según, se deben observar dos tipos de requisitos cuando se construye un sistema operativo, los cuales son:

Requisitos de usuario: Sistema fácil de usar y de aprender, seguro, rápido y adecuado al uso al que se le quiere destinar.

Requisitos del software: Donde se engloban aspectos como el mantenimiento, forma de operación, restricciones de uso, eficiencia, tolerancia frente a los errores y flexibilidad.

A continuación se describen las distintas estructuras que presentan los actuales sistemas operativos para satisfacer las necesidades que de ellos se quieren obtener.

Estructura Monolítica.

Es la estructura de los primeros sistemas operativos constituidos fundamentalmente por un solo programa compuesto de un conjunto de rutinas entrelazadas de tal forma que cada una puede llamar a cualquier otra. Las características fundamentales de este tipo de estructura son:

- Construcción del programa final a base de módulos compilados separadamente que se unen a través del ligador.
- Buena definición de parámetros de enlace entre las distintas rutinas existentes, que puede provocar mucho acoplamiento.
- Carecen de protecciones y privilegios al entrar a rutinas que manejan diferentes aspectos de los recursos de la computadora, como memoria, disco, etc.

Generalmente están hechos a medida, por lo que son eficientes y rápidos en su ejecución y gestión, pero por lo mismo carecen de flexibilidad para soportar diferentes ambientes de trabajo o tipos de aplicaciones.

Estructura Jerárquica.

A medida que fueron creciendo las necesidades de los usuarios y se perfeccionaron los sistemas, se hizo necesaria una mayor organización del software, del sistema operativo, donde una parte del sistema contenía subpartes y esto organizado en forma de niveles.

Se dividió el sistema operativo en pequeñas partes, de tal forma que cada una de ellas estuviera perfectamente definida y con un claro interface con el resto de elementos.

Se constituyó una estructura jerárquica o de niveles en los sistemas operativos, el primero de los cuales fue denominado THE (Technische Hogeschool, Eindhoven), de Dijkstra, que se utilizó con fines didácticos. Se puede pensar también en estos sistemas como si fueran 'multicapa'. Multics y Unix caen en esa categoría.

En la estructura anterior se basan prácticamente la mayoría de los sistemas operativos actuales. Otra forma de ver este tipo de sistema es la denominada de anillos concéntricos o "rings".

En el sistema de anillos, cada uno tiene una apertura, conocida como puerta o trampa (trap), por donde pueden entrar las llamadas de las capas inferiores. De esta forma, las zonas más internas del sistema operativo o núcleo del sistema estarán más protegidas de accesos indeseados desde las capas más externas. Las capas más internas serán, por tanto, más privilegiadas que las externas.

Máquina Virtual.

Se trata de un tipo de sistemas operativos que presentan una interface a cada proceso, mostrando una máquina que parece idéntica a la máquina real subyacente. Estos sistemas operativos separan dos conceptos que suelen estar unidos en el resto de sistemas: la multiprogramación y la máquina extendida. El objetivo de los sistemas operativos de máquina virtual es el de integrar distintos sistemas operativos dando la sensación de ser varias máquinas diferentes.

El núcleo de estos sistemas operativos se denomina monitor virtual y tiene como misión llevar a cabo la multiprogramación, presentando a los niveles superiores tantas máquinas virtuales como se soliciten. Estas máquinas virtuales no son máquinas extendidas, sino una réplica de la máquina real, de manera que en cada una de ellas se pueda ejecutar un sistema operativo diferente, que será el que ofrezca la máquina extendida al usuario

Cliente-Servidor(Microkernel).

El tipo más reciente de sistemas operativos es el denominado Cliente-servidor, que puede ser ejecutado en la mayoría de las computadoras, ya sean grandes o pequeñas.

Este sistema sirve para toda clase de aplicaciones por tanto, es de propósito general y cumple con las mismas actividades que los sistemas operativos convencionales.

El núcleo tiene como misión establecer la comunicación entre los clientes y los servidores. Los procesos pueden ser tanto servidores como clientes. Por ejemplo, un programa de aplicación normal es un cliente que llama al servidor correspondiente para acceder a un archivo o realizar una operación de entrada/salida sobre un dispositivo concreto. A su vez, un proceso cliente puede actuar como servidor para otro." [Alcal92]. Este paradigma ofrece gran flexibilidad en cuanto a los servicios posibles en el sistema final, ya que el núcleo provee solamente funciones muy básicas de memoria, entrada/salida, archivos y procesos, dejando a los servidores proveer la mayoría que el usuario final o programador puede usar. Estos servidores deben tener mecanismos de seguridad y protección que, a su vez, serán filtrados por el núcleo que controla el hardware. Actualmente se está trabajando en una versión de UNIX que contempla en su diseño este paradigma.

5. Sistemas Operativos por la Forma de Ofrecer sus Servicios

Esta clasificación también se refiere a una visión externa, que en este caso se refiere a la del usuario, el cómo accesa a los servicios. Bajo esta clasificación se pueden detectar dos tipos principales: sistemas operativos de red y sistemas operativos distribuidos.

Sistema Operativo de Red.

Los sistemas operativos de red se definen como aquellos que tiene la capacidad de interactuar con sistemas operativos en otras computadoras por medio de un medio de transmisión con el objeto de intercambiar información, transferir archivos, ejecutar comandos remotos y un sin fin de otras actividades. El punto crucial de estos sistemas es que el usuario debe saber la sintaxis de un conjunto de comandos o llamadas al sistema para ejecutar estas operaciones, además de la ubicación de los recursos que desee acceder. Por ejemplo, si un usuario en la computadora hidalgo necesita el archivo matriz.pas que se localiza en el directorio /software/codigo en la computadora morelos bajo el sistema operativo UNIX, dicho usuario podría copiarlo a través de la red con los comandos siguientes: hidalgo% hidalgo% rcp morelos:/software/codigo/matriz.pas . hidalgo%. En este caso, el comando rcp que significa "remote copy" trae el archivo indicado de la computadora morelos y lo coloca en el directorio donde se ejecutó el mencionado comando. Lo importante es hacer ver que el usuario puede acceder y compartir muchos recursos.

El primer Sistema Operativo de red estaba enfocado a equipos con un procesador Motorola 68000, pasando posteriormente a procesadores Intel como Novell Netware.

Los Sistemas Operativos de red mas ampliamente usados son: Novell Netware, Personal Netware, LAN Manager, Windows NT Server, UNIX, LANtastic.

Sistemas Operativos Distribuidos.

Los sistemas operativos distribuidos abarcan los servicios de los de red, logrando integrar recursos (impresoras, unidades de respaldo, memoria, procesos, unidades centrales de proceso) en una sola máquina virtual que el usuario accesa en forma transparente. Es decir, ahora el usuario ya no necesita saber la ubicación de los recursos, sino que los conoce por nombre y simplemente los usa como si todos ellos fuesen locales a su lugar de trabajo habitual. Todo lo anterior es el marco teórico de lo que se desearía tener como sistema operativo distribuido, pero en la realidad no se ha conseguido crear uno del todo, por la complejidad que suponen: distribuir los procesos en las varias unidades de procesamiento, reintegrar sub-resultados, resolver problemas de concurrencia y paralelismo, recuperarse de fallas de algunos recursos distribuidos y consolidar la protección y seguridad entre los diferentes componentes del sistema y los usuarios. Los avances tecnológicos en las redes de área local y la creación de microprocesadores de 32 y 64 bits lograron que computadoras mas o menos baratas tuvieran el suficiente poder en forma autónoma para desafiar en cierto grado a los mainframes, y a la vez se dio la posibilidad de intercomunicarlas, sugiriendo la oportunidad de partir procesos muy pesados en cálculo en unidades más pequeñas y distribuirlos en los varios microprocesadores para luego reunir los sub-resultados, creando así una máquina virtual en la red que exceda en poder a un mainframe. El sistema integrador de los microprocesadores que hacer ver a las varias memorias, procesadores, y todos los demás recursos como una sola entidad en forma transparente se le llama sistema operativo distribuido. Las razones para crear o adoptar sistemas distribuidos se dan por dos razones principales: por necesidad (debido a que los problemas a resolver son inherentemente distribuidos) o porque se desea tener más confiabilidad y disponibilidad de recursos. En el primer caso tenemos, por ejemplo, el control de los cajeros automáticos en diferentes estados de la república. Ahí no es posible ni eficiente mantener un control centralizado, es más, no existe capacidad de cómputo y de entrada/salida para dar servicio a los millones de operaciones por minuto. En el segundo caso, supóngase que se tienen en una gran empresa varios grupos de trabajo, cada uno necesita almacenar grandes cantidades de información en disco duro con una alta confiabilidad y disponibilidad. La solución puede ser que para cada grupo de trabajo se asigne una partición de disco duro en servidores diferentes, de manera que si uno de los servidores falla, no se deje dar el servicio a todos, sino sólo a unos cuantos y, más aún, se podría tener un sistema con discos en espejo (mirror) a través de la red, de manera que si un servidor se cae, el servidor en espejo continúa trabajando y el usuario ni cuenta se da de estas fallas, es decir, obtiene acceso a recursos en forma transparente.

Los sistemas distribuidos deben de ser muy confiables, ya que si un componente del sistema se compone otro componente debe de ser capaz de reemplazarlo.

Entre los diferentes Sistemas Operativos distribuidos que existen tenemos los siguientes: Sprite, Solaris-MC, Mach, Chorus, Spring, Amoeba, Taos, etc.

6. Proceso.

Uno de los conceptos mas importantes que gira entorno a un sistema operativo es el de proceso. Un proceso es un programa en ejecución junto con el entorno asociado (registros, variables ,etc.). El corazón de un sistema operativo es el núcleo, un programa de control que reacciona ante cualquier interrupción de eventos externos y que da servicio a los procesos, creándolos, terminándolos y respondiendo a cualquier petición de servicio por parte de los mismos.

Planificación del Procesador.

La planificación del procesador se refiere a la manera o técnicas que se usan para decidir cuánto tiempo de ejecución y cuando se le asignan a cada proceso del sistema. Obviamente, si el sistema es monousuario y monotarea no hay mucho que decidir, pero en el resto de los sistemas esto es crucial para el buen funcionamiento del sistema.

Características a considerar de los Procesos.

No todos los equipos de cómputo procesan el mismo tipo de trabajos, y un algoritmo de planificación que en un sistema funciona excelente puede dar un rendimiento pésimo en otro cuyos procesos tienen características diferentes. Estas características pueden ser:

- Cantidad de Entrada/Salida: Existen procesos que realizan una gran cantidad de operaciones de entrada y salida (aplicaciones de bases de datos, por ejemplo).
 - Cantidad de Uso de CPU: Existen procesos que no realizan muchas operaciones de entrada y salida, sino que usan intensivamente la unidad central de procesamiento. Por ejemplo, operaciones con matrices.
 - Procesos de Lote o Interactivos: Un proceso de lote es más eficiente en cuanto a la lectura de datos, ya que generalmente lo hace de archivos, mientras que un programa interactivo espera mucho tiempo (no es lo mismo el tiempo de lectura de un archivo que la velocidad en que una persona teclea datos) por las respuestas de los usuarios.
 - Procesos en Tiempo Real: Si los procesos deben dar respuesta en tiempo real se requiere que tengan prioridad para los turnos de ejecución.
 - Longevidad de los Procesos: Existen procesos que típicamente requerirán varias horas para finalizar su labor, mientras que existen otros que solo necesitan algunos segundos.
- Un proceso es una actividad que se apoya en datos, recursos, un estado en cada momento y un programa.

El Bloque de Control de Procesos (PCB).

Un proceso se representa desde el punto de vista del sistema operativo, por un conjunto de datos donde se incluyen el estado en cada momento, recursos utilizados, registros, etc., denominado Bloque de Control de Procesos (PCB).

Los objetivos del bloque de control de procesos son los siguientes:

Localización de la información sobre el proceso por parte del sistema operativo.

Mantener registrados los datos del proceso en caso de tener que suspender temporalmente su ejecución o reanudarla.

La información contenida en el bloque de control es la siguiente:

Estado del proceso. Información relativa al contenido del controlador del programa (Program Counter, PC), estado de procesador en cuanto a prioridad del proceso, modo de ejecución, etc., y por último el estado de los registros internos de la computadora.

Estadísticas de tiempo y ocupación de recursos para la gestión de la planificación del procesador.

Ocupación de memoria interna y externa para el intercambio (swapping).

Recursos en uso (normalmente unidades de entrada/salida).

Archivos en uso.

Privilegios.

Estas informaciones se encuentran en memoria principal en disco y se accede a ellas en los momentos en que se hace necesaria su actualización o consulta. Los datos relativos al estado del proceso siempre se encuentran en memoria principal.

Existe un Bloque de Control de Sistema (SCB) con objetivos similares al anterior y entre los que se encuentra el enlazado de los bloques de control de procesos existentes en el sistema.

El cambio de contexto se producirá en caso de ejecución de una instrucción privilegiada, una llamada al sistema operativo o una interrupción, es decir, siempre que se requiera la atención de algún servicio del sistema operativo.

Estado de los Procesos.

Los bloques de control de los procesos se almacenan en colas, cada una de las cuales representa un estado particular de los procesos, existiendo en cada bloque, entre otras informaciones. Los estados de los procesos son internos del sistema operativo y transparentes al usuario.

Los estados de los procesos se pueden dividir en dos tipos: activos e inactivos.

1.- Estados activos: Son aquellos que compiten con el procesador o están en condiciones de hacerlo. Se dividen en:

Ejecución. Estado en el que se encuentra un proceso cuando tiene el control del procesador. En un sistema monoprocesador este estado sólo lo puede tener un proceso.

Preparado. Aquellos procesos que están dispuestos para ser ejecutados, pero no están en ejecución por alguna causa (Interrupción, haber entrado en cola estando otro proceso en ejecución, etc.).

Bloqueado. Son los procesos que no pueden ejecutarse de momento por necesitar algún recurso no disponible (generalmente recursos de entrada/salida).

2.- Estados inactivos: Son aquellos que no pueden competir por el procesador, pero que pueden volver a hacerlo por medio de ciertas operaciones. En estos estados se mantiene el bloque de control de proceso aparcado hasta que vuelva a ser activado. Se trata de procesos que no han terminado su

trabajo que lo han impedido y que pueden volver a activarse desde el punto en que se quedaron sin que tengan que volver a ejecutarse desde el principio.

Son de dos tipos:

- Suspendido bloqueado. Es el proceso que fue suspendido en espera de un evento, sin que hayan desaparecido las causas de su bloqueo.
- Suspendido programado. Es el proceso que han sido suspendido, pero no tiene causa para estar bloqueado.

Operaciones sobre procesos.

Los sistemas operativos actuales poseen una serie de funciones cuyo objetivo es el de la manipulación de los procesos. Las operaciones que se pueden hacer sobre un proceso son las siguientes:

Crear el proceso. Se produce con la orden de ejecución del programa y suele necesitar varios argumentos, como el nombre y la prioridad del proceso. Aparece en este momento el PCB, que será insertado en la cola de procesos preparados.

La creación de un proceso puede ser de dos tipos:

Jerárquica. En ella, cada proceso que se crea es hijo del proceso creador y hereda el entorno de ejecución de su padre. El primer proceso que ejecuta un usuario será hijo del intérprete de comandos con el que interactúa.

No jerárquica. Cada proceso creado por otro proceso se ejecuta independientemente de su creador con un entorno diferente. Es un tipo de creación que no suele darse en los sistemas operativos actuales.

Destruir un proceso. Se trata de la orden de eliminación del proceso con la cual el sistema operativo destruye su PCB.

Suspender un proceso. Es un proceso de alta prioridad que paraliza un proceso que puede ser reanudado posteriormente. Suele utilizarse en ocasiones de mal funcionamiento o sobrecarga del sistema.

Reanudar un proceso. Trata de activar un proceso que a sido previamente suspendido.

Cambiar la prioridad de un proceso.

Temporizar la ejecución de un proceso. Hace que un determinado proceso se ejecute cada cierto tiempo (segundos, minutos, horas...) por etapas de una sola vez, pero transcurrido un periodo de tiempo fijo.

Despertar un proceso. Es una forma de desbloquear un proceso que habrá sido bloqueado previamente por temporización o cualquier otra causa.

Prioridades

Todo proceso por sus características e importancia lleva aparejadas unas determinadas necesidades de ejecución en cuanto a urgencia y asignación de recursos.

Las prioridades según los sistemas operativos se pueden clasificar del siguiente modo:

Asignadas por el sistema operativo. Se trata de prioridades que son asignadas a un proceso en el momento de comenzar su ejecución y dependen fundamentalmente de los privilegios de su propietario y del modo de ejecución.

- Asignadas por el propietario.
- Estáticas.
- Dinámicas.

El Núcleo del Sistema Operativo.

Todas las operaciones en las que participan procesos son controladas por la parte del sistema operativo denominada núcleo (nucleus, core o kernel, en inglés). El núcleo normalmente representa sólo una pequeña parte de lo que por lo general se piensa que es todo el sistema operativo, pero es tal vez el código que más se utiliza. Por esta razón, el núcleo reside por lo regular en la memoria principal, mientras que otras partes del sistema operativo son cargadas en la memoria principal sólo cuando se necesitan.

Los núcleos se diseñan para realizar "el mínimo" posible de procesamiento en cada interrupción y dejar que el resto lo realice el proceso apropiado del sistema, que puede operar mientras el núcleo se habilita para atender otras interrupciones.

El núcleo de un sistema operativo normalmente contiene el código necesario para realizar las siguientes funciones:

Manejo de interrupciones.

Creación y destrucción de procesos.

Cambio de estado de los procesos.

Despacho.

Suspensión y reanudación de procesos.
Sincronización de procesos.
Comunicación entre procesos.
Manipulación de los bloques de control de procesos.
Apoyo para las actividades de entrada/salida.
Apoyo para asignación y liberación de memoria.
Apoyo para el sistema de archivos.
Apoyo para el mecanismo de llamada y retorno de un procedimiento.
Apoyo para ciertas funciones de contabilidad del sistema.

Núcleo o Kernel y niveles de un Sistema Operativo.

El Kernel consiste en la parte principal del código del sistema operativo, el cual se encargan de controlar y administrar los servicios y peticiones de recursos y de hardware con respecto a uno o varios procesos, este se divide en 5 capas:

Nivel 1. Gestión de Memoria: que proporciona las facilidades de bajo nivel para la gestión de memoria secundaria necesaria para la ejecución de procesos.

Nivel 2. Procesador: Se encarga de activar los cuantums de tiempo para cada uno de los procesos, creando interrupciones de hardware cuando no son respetadas.

Nivel 3. Entrada/Salida: Proporciona las facilidades para poder utilizar los dispositivos de E/S requeridos por procesos.

Nivel 4. Información o Aplicación o Interprete de Lenguajes: Facilita la comunicación con los lenguajes y el sistema operativo para aceptar las ordenes en cada una de las aplicaciones. Cuando se solicitan ejecutando un programa el software de este nivel crea el ambiente de trabajo e invoca a los procesos correspondientes.

Nivel 5. Control de Archivos: Proporciona la facilidad para el almacenamiento a largo plazo y manipulación de archivos con nombre, va asignando espacio y acceso de datos en memoria.

El núcleo y los procesos.

El núcleo (Kernel) de un sistema operativo es un conjunto de rutinas cuya misión es la de gestionar el procesador, la memoria, la entrada/salida y el resto de procesos disponibles en la instalación. Toda esta gestión la realiza para atender al funcionamiento y peticiones de los trabajos que se ejecutan en el sistema.

Los procesos relacionados con la entidad básica de los sistemas operativos actuales: Los procesos.

El esquema general del mismo es el siguiente:

Definición y concepto de proceso.

El Bloque de Control de Proceso (PCB) como imagen donde el sistema operativo ve el estado del proceso.

Estados por los que pasa un proceso a lo largo de su existencia en la computadora.

Operaciones que se pueden realizar sobre un proceso.

Clasificación de los procesos según su forma de ejecución, de carga, etc.

7. Dispositivos de Entrada y Salida.

El código destinado a manejar la entrada y salida de los diferentes periféricos en un sistema operativo es de una extensión considerable y sumamente complejo. Resuelve la necesidades de sincronizar, atrapar interrupciones y ofrecer llamadas al sistema para los programadores.

Los dispositivos de entrada salida se dividen, en general, en dos tipos: dispositivos orientados a bloques y dispositivos orientados a caracteres.

Orientados a Bloques.

Los dispositivos orientados a bloques tienen la propiedad de que se pueden direccionar, esto es, el programador puede escribir o leer cualquier bloque del dispositivo realizando primero una operación de posicionamiento sobre el dispositivo. Los dispositivos más comunes orientados a bloques son los discos duros, la memoria, discos compactos y, posiblemente, unidades de cinta.

Orientados a Caracteres.

Los dispositivos orientados a caracteres son aquellos que trabajan con secuencias de bytes sin importar su longitud ni ninguna agrupación en especial. No son dispositivos direccionables. Ejemplos de estos dispositivos son el teclado, la pantalla o display y las impresoras.

La clasificación anterior no es perfecta, porque existen varios dispositivos que generan entrada o salida que no pueden englobarse en esas categorías. Por ejemplo, un reloj que genera pulsos. Sin embargo, aunque existan algunos periféricos que no se puedan categorizar, todos están administrados por el sistema operativo por medio de una parte electrónica - mecánica y una parte de software.

8. Principios del Software de Entrada y Salida.

Los principios de software en la entrada - salida se resumen en cuatro puntos: el software debe ofrecer manejadores de interrupciones, manejadores de dispositivos, software que sea independiente de los dispositivos y software para usuarios.

Manejadores de Interrupciones.

El primer objetivo referente a los manejadores de interrupciones consiste en que el programador o el usuario no debe darse cuenta de los manejos de bajo nivel para los casos en que el dispositivo está ocupado y se debe suspender el proceso o sincronizar algunas tareas. Desde el punto de vista del proceso o usuario, el sistema simplemente se tardó más o menos en responder a su petición.

Manejadores de Dispositivos.

El sistema debe proveer los manejadores de dispositivos necesarios para los periféricos, así como ocultar las peculiaridades del manejo interno de cada uno de ellos, tales como el formato de la información, los medios mecánicos, los niveles de voltaje y otros. Por ejemplo, si el sistema tiene varios tipos diferentes de discos duros, para el usuario o programador las diferencias técnicas entre ellos no le deben importar, y los manejadores le deben ofrecer el mismo conjunto de rutinas para leer y escribir datos.

Software que sea independiente de los dispositivos.

Este es un nivel superior de independencia que el ofrecido por los manejadores de dispositivos. Aquí el sistema operativo debe ser capaz, en lo más posible, de ofrecer un conjunto de utilerías para accesar periféricos o programarlos de una manera consistente. Por ejemplo, que para todos los dispositivos orientados a bloques se tenga una llamada para decidir si se desea usar 'buffers' o no, o para posicionarse en ellos.

Software para Usuarios.

La mayoría de las rutinas de entrada - salida trabajan en modo privilegiado, o son llamadas al sistema que se ligan a los programas del usuario formando parte de sus aplicaciones y que no le dejan ninguna flexibilidad al usuario en cuanto a la apariencia de los datos. Existen otras librerías en donde el usuario si tiene poder de decisión (por ejemplo la llamada a "printf" en el lenguaje "C"). Otra facilidad ofrecida son las áreas de trabajos encolados (spooling areas), tales como las de impresión y correo electrónico.

9. Manejo de los Dispositivos de E/S.

En el manejo de los dispositivos de E/S es necesario, introducir dos nuevos términos:

Buffering (uso de memoria intermedia).

El buffering trata de mantener ocupados tanto la CPU como los dispositivos de E/S. La idea es sencilla, los datos se leen y se almacenan en un buffer, una vez que los datos se han leído y la CPU va a iniciar inmediatamente la operación con ellos, el dispositivo de entrada es instruido para iniciar inmediatamente la siguiente lectura. La CPU y el dispositivo de entrada permanecen ocupados. Cuando la CPU esté libre para el siguiente grupo de datos, el dispositivo de entrada habrá terminado de leerlos. La CPU podrá empezar el proceso de los últimos datos leídos, mientras el dispositivo de entrada iniciará la lectura de los datos siguientes.

Para la salida, el proceso es análogo. En este caso los datos de salida se descargan en otro buffer hasta que el dispositivo de salida pueda procesarlos.

Este sistema soluciona en forma parcial el problema de mantener ocupados todo el tiempo la CPU y los dispositivos de E/S. Ya que todo depende del tamaño del buffer y de la velocidad de procesamiento tanto de la CPU como de los dispositivos de E/S.

El manejo de buffer es complicado. Uno de los principales problemas reside en determinar tan pronto como sea posible que un dispositivo de E/S a finalizado una operación. Este problema se resuelve mediante las interrupciones. Tan pronto como un dispositivo de E/S acaba con una operación interrumpe a la CPU, en ese momento la CPU detiene lo que está haciendo e inmediatamente transfiere el control a una posición determinada. Normalmente las instrucciones que existen en esta posición corresponden a

una rutina de servicio de interrupciones. La rutina de servicio de interrupción comprueba si el buffer no está lleno o no está vacío y entonces inicia la siguiente petición de E/S. La CPU puede continuar entonces el proceso interrumpido.

Cada diseño de computadora tiene su propio mecanismo de interrupción, pero hay varias funciones comunes que todos contemplan.

El buffering puede ser de gran ayuda pero pocas veces es suficiente.

Spooling.

El problema con los sistemas de cintas es que una lectora de tarjetas no podía escribir sobre un extremo mientras la CPU leía el otro. Los sistemas de disco eliminaron esa dificultad, moviendo la cabeza de un área del disco a otra.

En un sistema de discos, las tarjetas se leen directamente desde la lectora sobre el disco. La posición de las imágenes de las tarjetas se registran en una tabla mantenida por el sistema operativo. En la tabla se anota cada trabajo una vez leído. Cuando se ejecuta un trabajo sus peticiones de entrada desde la tarjeta se satisfacen leyendo el disco. Cuando el trabajo solicita la salida, ésta se copia en el buffer del sistema y se escribe en el disco. Cuando la tarea se ha completado se escribe en la salida realmente. Esta forma de procesamiento se denomina spooling, utiliza el disco como un buffer muy grande para leer tan por delante como sea posible de los dispositivos de entrada y para almacenar los ficheros hasta que los dispositivos de salida sean capaces de aceptarlos.

La ventaja sobre el buffering es que el spooling solapa la E/S de un trabajo con la computación de otro. Es una característica utilizada en la mayoría de los sistemas operativos.

Afecta directamente a las prestaciones. Por el costo de algo de espacio en disco y algunas tablas, la CPU puede simultanear la computación de un trabajo con la E/S de otros. De esta manera, puede mantener tanto a la CPU como a los dispositivos de E/S trabajando con un rendimiento mucho mayor. Además mantiene una estructura de datos llama job spooling, que hace que los trabajos ya leídos permanezcan en el disco y el sistema operativo puede seleccionar cual ejecutar, por lo tanto se hace posible la planificación de trabajos.

10. Administración de Archivos.

Un archivo es un conjunto de información, que se encuentra almacenada o guardada en la memoria principal del computador, en el disco duro, en el disquete flexible o en los discos compactos (Cd-Rom).

Antes de que un archivo pueda leerse o escribirse en él, debe abrirse, momento en el cual se verifican los permisos. Estos archivos se abren especificando en el computador la ruta de acceso al archivo desde el directorio raíz, que es la unidad principal del disco del computador, este puede ser un disco duro o disco flexible. Entonces el sistema operativo visualiza el entorno al abrir un archivo.

Uno de los problemas mas frecuentes en el manejo de archivos son los deadlock, un deadlock es una situación no deseada de espera indefinida y se da cuando en un grupo de procesos, dos o más procesos de ese grupo esperan por llevar a cabo una tarea que será ejecutada por otro proceso del mismo grupo, entonces se produce el bloqueo. Los bloqueos se dan tanto en los sistemas operativos tradicionales como en los distribuidos, solo que en estos últimos es más difícil de prevenirlos, evitarlos e incluso detectarlos, y si se los logra detectar es muy complicado solucionarlos ya que la información se encuentra dispersa por todo el sistema.

Una vez que un deadlock se detecta, es obvio que el sistema está en problemas y lo único que resta por hacer es una de dos cosas: tener algún mecanismo de suspensión o reanudación que permita copiar todo el contexto de un proceso incluyendo valores de memoria y aspecto de los periféricos que esté usando para reanudarlo otro día, o simplemente eliminar un proceso o arrebatarle el recurso, causando para ese proceso la pérdida de datos y tiempo.

Seguridad de un Sistema Operativo.

En los sistemas operativos se requiere tener una buena seguridad informática, tanto del hardware, programas y datos, previamente haciendo un balance de los requerimientos y mecanismos necesarios. Con el fin de asegurar la integridad de la información contenida.

Dependiendo de los mecanismos utilizados y de su grado de efectividad, se puede hablar de sistemas seguros e inseguros. En primer lugar, deben imponerse ciertas características en el entorno donde se encuentra la instalación de los equipos, con el fin de impedir el acceso a personas no autorizadas, mantener un buen estado y uso del material y equipos, así como eliminar los riesgos de causas de fuerza mayor, que puedan destruir la instalación y la información contenida.

En la actualidad son muchas las violaciones que se producen en los sistemas informáticos, en general por acceso de personas no autorizadas que obtienen información confidencial pudiendo incluso manipularla. En ocasiones este tipo de incidencias resulta grave por la naturaleza de los datos; por ejemplo si se trata de datos bancarios, datos oficiales que puedan afectar a la seguridad de los estados, etc.

El software mal intencionado que se produce por diversas causas, es decir pequeños programas que poseen gran facilidad para reproducirse y ejecutarse, cuyos efectos son destructivos nos estamos refiriendo a los virus informáticos.

Para esto, se analizan cuestiones de seguridad desde dos perspectivas diferentes la seguridad externa y la seguridad interna.

Todos los mecanismos dirigidos a asegurar el sistema informático sin que el propio sistema intervenga en el mismo se engloban en lo que podemos denominar seguridad externa.

La seguridad externa puede dividirse en dos grandes grupos:

Seguridad física. Engloba aquellos mecanismos que impiden a los agentes físicos la destrucción de la información existente en el sistema; entre ellos podemos citar el fuego, el humo, inundaciones descargas eléctricas, campos magnéticos, acceso físico de personas con no muy buena intención, entre otros.

Seguridad de administración. Engloba los mecanismos más usuales para impedir el acceso lógico de personas físicas al sistema.

Todos los mecanismos dirigidos a asegurar el sistema informático, siendo el propio sistema el que controla dichos mecanismos, se engloban en lo que podemos denominar seguridad interna.