

Parcial Arq&So

Introducción a los sistemas operativos.
Planificación por prioridad.

Gianfranco Robles, Lucas Peratta, Giuliano Rinaldi, Belén Burgos.

Grupo de trabajo: n°19

Universidad Tecnológica Nacional, Tecnicatura Universitaria en Programación

Arquitectura de Sistemas Operativos

Profesor:
Gustavo Ramoscelli
Ayudante:
Sergio Antozzi

SISTEMAS OPERATIVOS:

Un sistema operativo es un software que administra los recursos de hardware y software de una computadora, proporcionando servicios y soporte para la ejecución de aplicaciones y la gestión de tareas del usuario.

Un sistema operativo está presente en muchos dispositivos, no solo en un ordenador. Todos ellos son esenciales para que puedan funcionar.

Tipos de sistemas operativos:

Dentro de los sistemas operativos hay distintos tipos y categorías, como por ejemplo:

1- Sistemas operativos de mainframe:

Son sistemas operativos diseñados para enormes computadoras utilizadas en centros de datos corporativos. Tienen una gran capacidad de procesamiento y manejan múltiples tareas simultáneamente, como procesamiento por lotes, transacciones y tiempo compartido.

Ejemplos son OS/390 y z/OS.

Características:

- Capacidad de E/S:** Las mainframes manejan grandes volúmenes de datos y operaciones de entrada/salida.
- Procesamiento por Lotes:** Ejecuta trabajos rutinarios sin la necesidad de interacción del usuario, como procesamiento de reclamaciones o reportes de ventas.
- Procesamiento de Transacciones:** Maneja grandes cantidades de pequeñas peticiones rápidamente, como procesamiento de cheques bancarios o reservaciones aéreas.
- Tiempo Compartido:** Permite que varios usuarios remotos ejecuten trabajos simultáneamente, como consultas en grandes bases de datos.

Aplicaciones:

- Servidores Web de alto rendimiento.
- Servidores para sitios de comercio electrónico a gran escala.
- Servidores para transacciones de negocio a negocio.

2- Sistemas operativos de servidores:

Estos sistemas operativos se ejecutan en servidores, que son computadoras dedicadas a proporcionar servicios a través de redes. Los servidores dan servicio a varios usuarios a través de una red, permitiendo compartir recursos de hardware y software. También ofrecen servicios como almacenamiento de archivos, impresión y hospedaje web.

Ejemplos comunes son Windows Server, Linux, Solaris y FreeBSD.

Características:

- Multiusuario:** Permite a varios usuarios acceder y utilizar los recursos del servidor simultáneamente.
- Servicios Comunes:** Proporcionan servicios como impresión, archivos y web.
- Alta Disponibilidad:** Diseñados para operar continuamente y proporcionar servicios sin interrupción.

Aplicaciones:

- Proveedores de Internet utilizan servidores para dar soporte a sus clientes.
- Sitios web utilizan servidores para almacenar páginas web y manejar peticiones entrantes.
- Empresas utilizan servidores para gestionar y compartir recursos corporativos.

3- Sistemas operativos de multiprocesadores:

Estos sistemas operativos están diseñados para computadoras que tienen múltiples CPU, lo que permite una mayor potencia de procesamiento. Son utilizados en sistemas de alta gama, desde servidores hasta estaciones de trabajo avanzadas.

Ejemplos incluyen Windows Server y Linux.

Características:

- Procesamiento Paralelo:** Permiten la ejecución simultánea de múltiples procesos.
- Conectividad y Comunicación:** Necesitan sistemas operativos con características especiales para la comunicación y consistencia.
- Escalabilidad:** Diseñados para manejar un número creciente de núcleos de CPU.

Aplicaciones:

- Sistemas de alto rendimiento en investigaciones científicas.
- Servidores de bases de datos y aplicaciones empresariales que requieren gran capacidad de procesamiento.
- Computación en la nube y servicios de virtualización.

4- Sistemas operativos de computadoras personales:

Estos sistemas operativos son los que usamos en nuestras computadoras de escritorio y portátiles. Son diseñados para proporcionar soporte eficiente a un solo usuario, además soportan la multiprogramación y permiten la ejecución de múltiples programas simultáneamente.

Ejemplos son Windows, macOS , FreeBSD y diversas distribuciones de Linux.

Características:

- Interfaz de Usuario:** Proporcionan una interfaz amigable para el usuario.
- Soporte de Aplicaciones:** Soportan una amplia variedad de aplicaciones de uso común como procesamiento de texto, hojas de cálculo y navegación por Internet.
- Actualización Continua:** Reciben actualizaciones regulares para mejorar la funcionalidad y seguridad.

Aplicaciones:

- Uso personal y de oficina para tareas diarias como edición de documentos, gestión de correos electrónicos y navegación web.
- Juegos y aplicaciones multimedia.
- Desarrollo de software y programación.

5- Sistemas operativos de computadoras de bolsillo:

Son sistemas operativos diseñados para dispositivos portátiles como PDAs y teléfonos inteligentes. Ofrecen una variedad de funciones, desde administración de contactos hasta navegación web.

Ejemplos son Android, iOS y antiguamente Palm OS.

Características:

- Portabilidad:** Pequeños y ligeros, diseñados para ser llevados fácilmente.
- Funciones Específicas:** Incluyen libreta de direcciones, bloc de notas, telefonía y fotografía digital.
- Ejecución de Aplicaciones:** Soportan aplicaciones desarrolladas por terceros.

Aplicaciones:

- Gestión de contactos y calendarios.
- Navegación GPS y mapas.
- Aplicaciones de mensajería y redes sociales.

6- Sistemas operativos de sistemas integrados:

Estos sistemas operativos son esenciales para el funcionamiento de muchos dispositivos electrónicos de uso diario. Se encuentran en dispositivos incrustados que controlan funciones específicas, como electrodomésticos, automóviles y dispositivos médicos. Son altamente especializados y tienen un tamaño reducido.

Ejemplos incluyen QNX y VxWorks.

Características:

- Funcionalidad Fija:** Todo el software está preinstalado y no se puede cambiar.

-**Alta Fiabilidad:** No necesitan protección en las aplicaciones, lo cual simplifica su diseño.

-**Bajo Consumo:** Optimizados para funcionar con recursos limitados.

Aplicaciones:

- Electrodomésticos como hornos de microondas y televisores.
- Automóviles y sistemas de control en vehículos.
- Teléfonos celulares y reproductores de sonido.

7- Sistemas operativos de nodos sensores:

Estos sistemas operativos se utilizan en redes de pequeños dispositivos que recopilan datos ambientales, como temperatura y humedad. Son utilizados en aplicaciones como monitoreo de seguridad y gestión ambiental.

Ejemplos incluyen TinyOS y Contiki.

Características:

- Pequeños y Autónomos:** Diseñados para operar en condiciones ambientales rudas y con energía limitada.
- Resiliencia:** La red debe tolerar fallas en nodos individuales.
- Simplicidad:** El sistema operativo es pequeño y responde a eventos externos.

Aplicaciones:

- Monitoreo ambiental y detección de incendios.
- Seguridad y vigilancia.
- Predicción meteorológica y estudios ecológicos.

8- Sistemas operativos en tiempo real:

Estos sistemas operativos están diseñados para ejecutar tareas con tiempos estrictos de respuesta, como sistemas de control industrial y sistemas de navegación. Pueden ser "duros", o "suaves" y son esenciales para aplicaciones que requieren respuestas precisas y rápidas.

Ejemplos son RTLinux y FreeRTOS.

Características:

- Tiempo Real Duro:** Garantiza que las acciones ocurran en un tiempo determinado sin excepciones.
- Tiempo Real Suave:** Permite algunas fallas en el tiempo de respuesta sin afectar gravemente el sistema.
- Bibliotecas Enlazadas:** A veces, el sistema operativo es simplemente una biblioteca enlazada con los programas de aplicación.

Aplicaciones:

- Control de procesos industriales y robótica.
- Aeronáutica y aplicaciones militares.
- Sistemas de audio digital y multimedia.

9- Sistemas operativos de tarjetas inteligente:

Son sistemas operativos diseñados para tarjetas inteligentes, que se utilizan en aplicaciones como tarjetas de crédito y tarjetas de acceso. Tienen recursos limitados y están diseñados para realizar funciones específicas, como autenticación y procesamiento de pagos.

Ejemplos son tarjetas inteligentes que manejan múltiples applets de Java.

Características:

- Funcionalidad Limitada:** Realizan funciones específicas como pagos electrónicos.
- Energía Limitada:** Algunas tarjetas se energizan mediante inducción, limitando sus capacidades.
- Soporte para Java:** Algunas tarjetas inteligentes funcionan con Java, ejecutando applets mediante un intérprete de la JVM.

Aplicaciones:

- Pagos electrónicos y sistemas de identificación.
- Acceso seguro a sistemas y redes.
- Almacenamiento seguro de información personal.

Algoritmo de Planificación por prioridad:

Planificación por Prioridad

La planificación por prioridad es una estrategia de gestión de procesos que asigna prioridades a cada proceso, de manera que el proceso con la prioridad

más alta es el que se ejecuta primero. A diferencia de la planificación por turno circular, que asume que todos los procesos tienen igual importancia, la planificación por prioridad reconoce que diferentes procesos pueden tener diferentes niveles de importancia.

Conceptos Clave

Asignación de Prioridades:

Estática: Las prioridades se asignan una vez y no cambian durante la ejecución del proceso. Ejemplos incluyen sistemas militares donde los procesos iniciados por oficiales de mayor rango tienen mayores prioridades.

Dinámica: Las prioridades pueden cambiar durante la ejecución del proceso para cumplir con ciertos objetivos, como mejorar el rendimiento del sistema.

Mecanismo de Ejecución:

Ejecución Basada en Prioridad: El proceso con la prioridad más alta se ejecuta primero. Si varios procesos tienen la misma prioridad, se pueden utilizar otras estrategias como el turno circular.

Reducción de Prioridad: Para evitar que los procesos de alta prioridad se ejecuten indefinidamente, la prioridad del proceso actual puede reducirse en cada pulso del reloj. Esto puede llevar a una conmutación de procesos si otro proceso obtiene una prioridad más alta.

Manejo del Cuántum de Tiempo:

Asignación de Cuántum: Cada proceso recibe un cuántum de tiempo máximo para ejecutarse. Una vez que se agota su cuántum, el siguiente proceso con la prioridad más alta recibe la CPU.

Ajuste de Prioridades:

Si no se ajustan las prioridades ocasionalmente, los procesos de menor prioridad pueden quedar sin tiempo de CPU ("muere de hambre").

Clases de prioridad:

Estructura de Clases: Los procesos pueden agruparse en clases de prioridad. Se utiliza la planificación por prioridad entre las clases y turno circular dentro de cada clase.

Ejemplo de Clases: Un sistema puede tener cuatro clases de prioridad, donde la clase 4 tiene la mayor prioridad y la clase 1 la menor. Mientras haya procesos en la clase 4, sólo esos se ejecutan. Si la clase 4 está vacía, se pasa a la clase 3, y así sucesivamente.

Ejemplos y Aplicaciones:

Universidades: Procesos iniciados por decanos tienen mayor prioridad que los de profesores, secretarías, conserjes y estudiantes.

Computadoras Militares: Procesos iniciados por generales tienen la prioridad más alta, seguidos por los de coroneles, mayores, capitanes, tenientes, etc.

Centros Computacionales Comerciales: Los trabajos con mayor prioridad pueden tener un costo más alto, reflejando la importancia del trabajo.

Resumen:

La planificación por prioridad es esencial para sistemas donde ciertos procesos deben tener preferencia sobre otros debido a su importancia. Sin embargo, debe gestionarse cuidadosamente para evitar la **inanición** de procesos de menor prioridad y asegurar un uso eficiente de los recursos del sistema.

Inanición:

La inanición, en sistemas operativos, es cuando un proceso nunca obtiene los recursos necesarios para ejecutarse porque otros procesos de mayor prioridad siempre ocupan esos recursos. Esto ocurre cuando las prioridades se asignan de manera fija y no cambian, o cuando no hay mecanismos para **ajustar las prioridades**(explicado arriba).