Respuestas Basadas en Diseño de Sistemas Operativos

# Concurrencia

## 1. Condiciones para la Ejecución Concurrente

Para permitir la ejecución concurrente en un sistema operativo, se deben cumplir ciertas condiciones fundamentales:

- Gestión de procesos.

- Sincronización y exclusión mutua.

- Comunicación entre procesos (IPC).

- Planificación eficiente de CPU.

## 2. Aspectos del Sistema Operativo para la Concurrencia

El sistema operativo debe manejar la planificación, la sincronización, la gestión de memoria, y la comunicación entre procesos, usando mecanismos como semáforos y monitores.

## 3. Problemas Clásicos de Concurrencia

a. Productor-Consumidor: Sincronización entre un proceso que genera datos y otro que los consume.

b. Lectores-Escritores: Controla el acceso simultáneo entre procesos que solo leen y los que escriben.

c. Filósofos que Cenan: Se trata de coordinar múltiples procesos que compiten por recursos limitados.

d. Rendez-vous: Dos procesos deben sincronizarse en un punto exacto antes de continuar.

## 4. Sección Crítica

La sección crítica es una parte del código donde un proceso accede a recursos compartidos, y se deben evitar conflictos de acceso simultáneo. El algoritmo debe garantizar la exclusión mutua, el progreso y la espera limitada.

## 5. Tipos de Interacciones entre Procesos

Las interacciones pueden ser cooperativas o competitivas, y los procesos deben comunicarse de manera segura, ya sea compartiendo memoria o mediante mensajes.

## 6. Tratamiento de la Exclusión Mutua por Hardware y Software

Por hardware, se puede garantizar con instrucciones atómicas como "test-and-set". Por software, se utilizan algoritmos como los de Dekker o Peterson.

## 7. Semáforos

Los semáforos son mecanismos para sincronización. Pueden ser binarios o contadores, y controlan el acceso a recursos compartidos mediante operaciones wait() y signal().

## 8. Monitor

Un monitor encapsula la sincronización y permite exclusión mutua automática, evitando la manipulación directa de los semáforos. Es más seguro pero más complejo que los semáforos.

## 9. Espera Activa vs Bloqueo

La espera activa implica ciclos de CPU desperdiciados, ya que el proceso espera activamente a que se libere un recurso. El bloqueo es más eficiente porque el proceso es suspendido hasta que el recurso esté disponible.

## 10. Esquemas de Comunicación entre Procesos

Existen varios esquemas de comunicación: paso de mensajes, memoria compartida, entre otros. Cada uno tiene sus ventajas dependiendo de los requisitos del sistema.

# Administración de la Información

## 1. Sistema de Archivos en Sentido Amplio y Estricto

En sentido amplio, un sistema de archivos es el conjunto de estructuras que permiten almacenar, organizar y recuperar datos en un sistema de almacenamiento. En sentido estricto, es la implementación que organiza y maneja el almacenamiento en disco.

## 2. Definición de Conceptos

a. Dato: Unidad mínima de información.

b. Campo: Un atributo de un registro de datos.

c. Información: Conjunto de datos organizados que tienen significado.

d. Archivo: Colección de datos relacionados almacenados bajo un nombre.

e. Registro: Conjunto de campos que representan una entidad.

**Administración de la Información**

**1. Sistema de Archivos en Sentido Amplio y Estricto**

* **Sentido Amplio**: Se refiere a todo el conjunto de estructuras y servicios que provee un sistema operativo para almacenar, organizar y recuperar datos en medios de almacenamiento, como discos duros.
* **Sentido Estricto**: Hace referencia a la implementación concreta de cómo se gestionan y organizan los archivos dentro del sistema operativo, lo que incluye el sistema de archivos, las tablas de asignación de archivos, los directorios, etc.

**2. Definición de Conceptos**

* **Dato**: Unidad mínima de información que puede ser procesada.
* **Campo**: Atributo de un registro que contiene una pieza específica de información.
* **Información**: Conjunto de datos que son procesados y organizados para tener un significado útil.
* **Archivo**: Conjunto de datos organizados y almacenados bajo un nombre común.
* **Registro**: Conjunto de campos que forman una estructura lógica de información dentro de un archivo.

**3. Aspectos Relevantes al Definir un File System**

Al diseñar un sistema de archivos, algunos de los aspectos más importantes son:

* **Capacidad**: El tamaño máximo que puede gestionar el sistema de archivos.
* **Rendimiento**: La velocidad con la que se puede acceder a los archivos y escribir datos.
* **Seguridad**: El control de acceso a los archivos para evitar manipulaciones indebidas.
* **Fiabilidad**: La capacidad del sistema para recuperarse de fallos, como la pérdida de datos.

**4. Principales Atributos de un Archivo**

* Nombre
* Identificador único
* Tipo de archivo
* Ubicación en el disco
* Tamaño del archivo
* Tiempos de creación, modificación y acceso

**5. Diferentes Tipos de Archivos**

* **Archivos regulares**: Contienen datos de usuarios o programas.
* **Archivos de directorio**: Contienen listas de otros archivos.
* **Archivos especiales**: Utilizados para representar dispositivos de hardware.

**6. Operaciones Controlables en un Archivo**

* Creación
* Lectura
* Escritura
* Eliminación
* Renombrado

**7. Diferencia entre Windows y Linux en Extensiones de Archivos**

* **Windows**: Utiliza extensiones de archivo como .exe, .txt para determinar el tipo de archivo.
* **Linux**: No depende de las extensiones para determinar el tipo de archivo, utiliza permisos y metadatos del archivo.

**8. Diferencia entre Métodos de Acceso y Asignación**

* **Método de Acceso**: Se refiere a cómo se accede a los datos de un archivo. Puede ser secuencial o aleatorio.
* **Método de Asignación**: Se refiere a cómo se distribuyen los bloques de datos de un archivo en el sistema de almacenamiento, por ejemplo, asignación continua, enlazada o indexada.

**Procesos**

**1. Necesidad de Scheduling de CPU**

El sistema operativo debe realizar la planificación de la CPU para gestionar eficientemente el tiempo de ejecución de múltiples procesos, optimizando el uso de recursos y garantizando que todos los procesos reciban tiempo de CPU.

**2. Estados de un Proceso**

Los estados posibles de un proceso son:

* **Nuevo**: El proceso está siendo creado.
* **Ejecutándose**: El proceso está ejecutándose en la CPU.
* **Esperando**: El proceso está a la espera de un recurso.
* **Terminado**: El proceso ha finalizado su ejecución.

**3. Elementos de un PCB (Process Control Block)**

El PCB contiene información esencial para gestionar un proceso, incluyendo:

* Identificador del proceso (PID)
* Estado del proceso
* Contador de programa
* Registros de CPU
* Información de memoria
* Información de I/O

**4. Tabla de Procesos**

La tabla de procesos contiene todos los PCBs de los procesos del sistema. Cada entrada en la tabla contiene información relevante para la planificación y gestión del proceso.

**5. Tipos de Schedulers**

* **Short-term scheduler**: Selecciona qué proceso debe ejecutarse a continuación.
* **Long-term scheduler**: Controla cuántos procesos están en la memoria principal.
* **Medium-term scheduler**: Suspende procesos para liberar memoria.

**6. Dispatcher y Loader**

* **Dispatcher**: Encargado de cambiar el contexto de un proceso al ser seleccionado por el scheduler.
* **Loader**: Carga un programa en la memoria para que pueda ejecutarse.

**Gantt para Planificación de Procesos**

Cuando se realiza la planificación de procesos, se suelen aplicar algoritmos como **First-Come, First-Served (FCFS)**, **Shortest Job First (SJF)** o **Round-Robin (RR)**. Estos algoritmos se visualizan mediante diagramas de Gantt, donde se muestran los tiempos de espera y ejecución de los procesos a lo largo del tiempo.

Un ejemplo para la planificación FCFS sería:

* **Proceso P1**: Tiempo de CPU = 10
* **Proceso P2**: Tiempo de CPU = 29
* **Proceso P3**: Tiempo de CPU = 3

Al aplicar FCFS, los procesos se ejecutan en el orden de llegada y se calcula el tiempo de turnaround y el tiempo de espera.

Tienes razón, aún faltan varias preguntas por responder en detalle. A continuación, continuaré proporcionando respuestas completas para las preguntas restantes del archivo:

**Administración de la Información (continuación)**

**9. Métodos de Acceso a un Archivo**

Los principales métodos de acceso a un archivo incluyen:

* **Acceso Secuencial**: Los archivos se leen o escriben secuencialmente desde el principio hasta el final. Es el método más común y eficiente para archivos de texto o archivos grandes donde los datos se procesan en orden.
* **Acceso Directo o Aleatorio**: Permite acceder a cualquier parte del archivo directamente, sin tener que pasar por los datos anteriores. Es útil para bases de datos o archivos donde se necesita acceso rápido a registros específicos.

**10. Métodos de Asignación de Espacio para Archivos**

Existen varios métodos para asignar el espacio de almacenamiento a los archivos en un sistema de archivos:

* **Asignación Contigua**: Todos los bloques de un archivo están almacenados de manera contigua en el disco. Proporciona acceso rápido pero puede causar fragmentación externa.
* **Asignación Enlazada**: Los bloques de un archivo están enlazados mediante punteros. Evita la fragmentación externa pero el acceso es más lento.
* **Asignación Indexada**: Utiliza un bloque de índice para mantener una lista de punteros a los bloques de datos. Es más flexible que la asignación contigua o enlazada, pero requiere más espacio para almacenar el índice.

**11. Fragmentación en Asignación de Archivos**

* **Fragmentación Externa**: Ocurre cuando hay suficientes espacios libres en el disco, pero no están contiguos. Esto sucede en la asignación contigua y se puede mitigar mediante la desfragmentación.
* **Fragmentación Interna**: Ocurre cuando se asigna más espacio del necesario a un archivo. Esto es más común en sistemas que utilizan bloques de tamaño fijo para la asignación.

**12. Alternativas para Manejar la Lista de Espacios Libres**

Las principales alternativas para manejar los espacios libres incluyen:

* **Mapa de bits**: Cada bit representa un bloque de almacenamiento. Un bit en "0" indica que el bloque está libre, mientras que "1" indica que está ocupado.
* **Lista enlazada**: Los bloques libres se almacenan en una lista enlazada, donde cada bloque contiene un puntero al siguiente bloque libre.
* **Grupo de bloques libres**: Los bloques libres se agrupan en grupos, lo que permite gestionar grandes áreas de espacio libre de manera más eficiente.

**13. Estructuras de Directorio**

Las estructuras de directorio permiten organizar archivos y directorios dentro del sistema de archivos. Algunas de las estructuras más comunes son:

* **Estructura de Directorio Plano**: Todos los archivos están en un único nivel sin subdirectorios. Es simple pero no escalable.
* **Estructura Jerárquica**: Los archivos están organizados en una estructura de árbol con directorios y subdirectorios. Permite una mejor organización.
* **Directorios en Red (gráficos)**: Permiten que un archivo o directorio pertenezca a múltiples directorios.

**14. Diferencia entre Hard-link y Soft-link**

* **Hard-link**: Es una referencia directa al archivo real en el sistema de archivos. Varios hard-links pueden apuntar al mismo archivo y compartir el mismo inode.
* **Soft-link (enlace simbólico)**: Es un puntero que apunta al nombre de un archivo en lugar del archivo en sí. Si el archivo al que apunta es eliminado, el enlace simbólico queda roto.

**15. Implementación de Directorios en Unix y DOS**

* **Unix**: Utiliza un sistema de archivos jerárquico basado en inodes, donde cada archivo o directorio tiene un inode asociado que contiene metadatos y punteros a los bloques de datos.
* **DOS**: Utiliza una estructura de árbol para organizar archivos y directorios, pero su sistema de archivos es más limitado en cuanto a la gestión de nombres de archivos y subdirectorios.

**16. Tamaño Máximo de Archivo en un Sistema UNIX**

Para calcular el tamaño máximo de archivo, se debe considerar el número de punteros directos, de indirección simple, doble y triple. Con un tamaño de bloque de 4 KB y punteros de 16 bits, se pueden almacenar archivos grandes utilizando el sistema de direcciones de UNIX.

**Procesos (continuación)**

**7. Mecanismos para Interrumpir la Ejecución de un Proceso**

Los mecanismos de interrupción de procesos incluyen:

* **Interrupciones por hardware**: Cuando un dispositivo de entrada/salida finaliza una operación.
* **Interrupciones por software**: Cuando un programa hace una llamada al sistema.
* **Interrupciones por temporizador**: Para garantizar que un proceso no monopolice la CPU.

**8. Efecto Convoy**

El efecto convoy ocurre cuando un proceso lento retiene la CPU durante mucho tiempo, haciendo que otros procesos más rápidos esperen innecesariamente. Este problema es común en la planificación FCFS, donde los procesos de larga duración bloquean a los de corta duración.

**9. Análisis de Quantum en Planificación RR**

El valor del quantum en la planificación **Round-Robin (RR)** tiene un impacto significativo en el rendimiento del sistema:

* **Quantum grande**: El sistema se comporta de manera similar a FCFS, lo que podría llevar al efecto convoy.
* **Quantum pequeño**: El cambio de contexto ocurre con mayor frecuencia, lo que incrementa el overhead.
* **Quantum cercano a E (tiempo de ejecución promedio de un proceso)**: Ofrece un buen balance entre la interactividad y el uso de la CPU.
* **Quantum igual a C (tiempo de cambio de contexto)**: Los cambios de contexto ocurren continuamente, causando una pérdida significativa de tiempo de CPU.

**10. Esquema de Scheduling para Programas Clasificados**

Un esquema de scheduling en sistemas con grupos de programas se puede implementar con **colas multinivel**, donde cada nivel tiene su propia política de planificación. Por ejemplo, UNIX utiliza una cola multinivel con retroalimentación, donde los procesos de menor prioridad pueden ser promovidos a colas de mayor prioridad con el tiempo (aging).

**11. Diagrama de Gantt para FCFS, SJF y RR**

* **FCFS**: Los procesos se ejecutan en el orden en que llegan.
* **SJF (Shortest Job First)**: Los procesos más cortos se ejecutan primero. Esto puede reducir el tiempo de espera promedio, pero requiere conocer de antemano la duración de los procesos.
* **RR (q=10)**: Cada proceso obtiene un tiempo fijo de ejecución (quantum), después del cual pasa a la cola de listos.

**12. Cálculo de Turnaround, Espera y Respuesta**

Para cada algoritmo de planificación (FCFS, SJF, Prioridad, RR), es posible calcular el **tiempo de turnaround** (tiempo total desde que un proceso llega hasta que finaliza), **tiempo de espera** (tiempo que un proceso pasa en la cola de listos) y **tiempo de respuesta** (tiempo desde que un proceso llega hasta que comienza a ejecutarse).

**13. Algoritmo de Prioridad con Aging**

En un sistema que utiliza planificación por prioridad, el **aging** es un mecanismo que aumenta la prioridad de un proceso a medida que pasa más tiempo en la cola de listos. Esto evita la inanición de procesos de baja prioridad.

**14. Planificación en un Sistema Multiprogramado**

En un sistema multiprogramado que utiliza colas multinivel sin retroalimentación, los procesos interactivos tienen la mayor prioridad y se ejecutan con un quantum corto (RR con q=4), mientras que los procesos batch tienen la prioridad más baja y se ejecutan mediante **FCFS**.

**Unidad 1 - Conceptos Elementales**

**1. Definición de “Sistema Operativo” y Objetivos Principales**

Un **sistema operativo** es un software fundamental que actúa como intermediario entre los usuarios y el hardware de una computadora. Los objetivos principales de un sistema operativo son:

* **Facilitar la ejecución de programas de usuario**.
* **Gestionar los recursos del hardware de manera eficiente**.
* **Proporcionar una interfaz cómoda y segura para interactuar con el sistema**.

**2. Generaciones en la Evolución de los Sistemas Operativos**

* **Primera Generación (1940-1950)**: Computadoras sin sistemas operativos, los programas se ejecutaban directamente.
* **Segunda Generación (1950-1960)**: Introducción de sistemas de procesamiento por lotes.
* **Tercera Generación (1960-1980)**: Aparece la multiprogramación, y se mejora la gestión de recursos.
* **Cuarta Generación (1980-presente)**: Sistemas operativos orientados a redes, sistemas distribuidos y sistemas en tiempo real.

**3. Definiciones de Términos Clave**

* **Tarea**: Trabajo o acción específica que un sistema debe realizar.
* **Programa**: Conjunto de instrucciones para realizar una tarea específica.
* **Proceso**: Instancia de un programa en ejecución.
* **Multiprogramación**: Ejecución de múltiples procesos en un mismo sistema, compartiendo la CPU.
* **Multitarea**: Capacidad de un sistema de ejecutar varias tareas o procesos simultáneamente.
* **Multiprocesamiento**: Uso de más de una CPU en un sistema operativo para ejecutar múltiples procesos en paralelo.

**4. Ventajas de la Multitarea**

* **Mejor uso de los recursos del sistema**.
* **Mayor rendimiento general del sistema**.
* **Mayor capacidad de respuesta a las interacciones de usuario**.

**5. Recursos de un Sistema de Computadoras**

Un sistema de computadoras típicamente administra los siguientes recursos:

* **CPU**: Proporciona tiempo de procesamiento para ejecutar instrucciones.
* **Memoria**: Espacio temporal para almacenar datos y programas.
* **Dispositivos de entrada/salida**: Permiten la comunicación entre el usuario y la computadora.
* **Archivos**: Almacenan datos de manera permanente.

**6. Modo Dual de Operación: Kernel y Usuario**

* **Modo Kernel (supervisor)**: En este modo, el sistema operativo tiene acceso total a todo el hardware y puede ejecutar instrucciones privilegiadas.
* **Modo Usuario**: En este modo, los programas de los usuarios tienen acceso limitado al hardware, y deben hacer llamadas al sistema para acceder a servicios restringidos.

**7. Instrucción Privilegiada**

Las **instrucciones privilegiadas** solo pueden ejecutarse en el modo kernel. Ejemplos incluyen:

* Manipulación de la tabla de páginas.
* Encender o apagar interrupciones.
* Cambiar el modo de usuario a modo kernel.
* Operaciones de I/O.

**8. Llamadas al Sistema y Programas del Sistema**

* **Llamada al Sistema**: Es un mecanismo que permite que un programa solicite un servicio del sistema operativo, como abrir un archivo o asignar memoria.
* **Programa del Sistema**: Un conjunto de utilidades que proporcionan funcionalidades a los usuarios, como comandos de consola.

**9. Interrupciones**

Una **interrupción** es un evento que detiene temporalmente la ejecución de un proceso para atender una tarea urgente. Pueden ser:

* **Interrupciones de hardware**: Originadas por dispositivos de E/S.
* **Interrupciones de software**: Como excepciones y llamadas al sistema.

**10. Relación entre Instrucción Privilegiada, Llamada al Sistema e Interrupción**

* **Instrucción Privilegiada**: Solo se ejecuta en modo kernel.
* **Llamada al Sistema**: Provoca una interrupción de software para solicitar servicios del sistema operativo.
* **Interrupción**: El sistema operativo responde a la interrupción y ejecuta el código necesario.

**11. Secuencia de Eventos Ante una Interrupción**

1. El hardware detiene la ejecución del proceso.
2. El sistema guarda el estado del proceso interrumpido.
3. El sistema ejecuta el controlador de interrupciones.
4. Tras atender la interrupción, el sistema restaura el estado del proceso y retoma su ejecución.

**12. Diferencia entre Buffering y Spooling**

* **Buffering**: Almacena temporalmente datos durante la transferencia entre dos dispositivos.
* **Spooling**: Gestiona la cola de trabajos de E/S, permitiendo que varios trabajos se almacenen en cola para ser procesados secuencialmente.

**Unidad 2 - Procesos y Scheduling**

**1. Necesidad del Scheduling de CPU**

El scheduling de CPU es necesario para distribuir el tiempo de CPU de manera justa y eficiente entre los procesos en ejecución. El objetivo es maximizar el rendimiento del sistema y minimizar el tiempo de espera de los procesos.

**2. Diagrama de Estados de un Proceso**

Un proceso puede estar en uno de los siguientes estados:

* **Nuevo**: El proceso está siendo creado.
* **Listo**: El proceso está esperando ser ejecutado.
* **Ejecutándose**: El proceso está utilizando la CPU.
* **Bloqueado**: El proceso está esperando un evento externo.
* **Terminado**: El proceso ha finalizado su ejecución.

**3. Elementos Típicos de un PCB**

El PCB (Process Control Block) contiene:

* Identificador del proceso (PID).
* Estado del proceso.
* Contador de programa.
* Punteros a la pila y registros.
* Información de administración de memoria.
* Información de E/S.

**4. Estructura de una Tabla de Procesos**

Una tabla de procesos contiene los PCB de todos los procesos activos en el sistema. Cada PCB incluye información sobre el estado del proceso, sus recursos asignados y su ubicación en memoria.

**5. Tipos de Schedulers**

* **Short-term Scheduler**: Selecciona qué proceso debe ejecutarse a continuación.
* **Long-term Scheduler**: Decide qué procesos se cargan en la memoria principal.
* **Medium-term Scheduler**: Decide cuándo suspender o reanudar un proceso.

**6. Dispatcher y Loader**

* **Dispatcher**: Cambia el contexto del proceso cuando el scheduler selecciona uno nuevo.
* **Loader**: Carga el código y los datos de un programa en la memoria para su ejecución.

**7. Mecanismos para Interrumpir Procesos**

Las interrupciones son mecanismos que permiten al sistema operativo suspender temporalmente un proceso para realizar otras tareas. Esto puede ocurrir por una solicitud de E/S, interrupciones por hardware, excepciones, etc.