

TAREA 5

Sistemas Operativos

Integrantes:

Mora Ayala José Antonio
Ramírez Cotonieto Luis Fernando
Torres Carrillo Josehf Miguel Ángel
Tovar Jacuinde Rodrigo

Profesor:

Cortés Galicia Jorge
2CM17

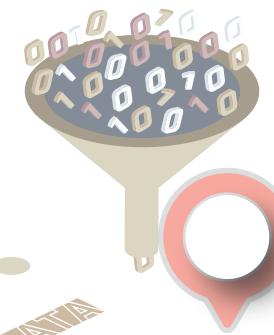
Proceso Independiente

Proceso

Los procesos que se ejecutan concurrentemente pueden ser procesos independientes o procesos cooperativos.

Proceso Cooperativo

Afecta o es afectado por los demás procesos que se ejecutan en el sistema.



Requiere mecanismos de comunicación interprocesos (IPC, interprocess communication) que les permitan intercambiar datos e información

Comunicación interprocesos

Almacenamiento en búfer

Sea la comunicación directa o indirecta/ los mensajes intercambiados por los procesos que se están comunicando residen en una cola temporal.

Sincronización

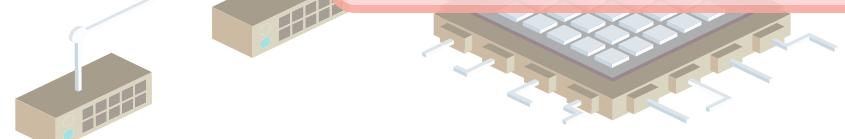
- Envío con bloqueo. El proceso que envía se bloquea hasta que el proceso receptor o el buzón de correo reciben el mensaje.
- Envío sin bloqueo. El proceso transmisor envía el mensaje y continúa operando.
- Recepción con bloqueo. El receptor se bloquea hasta que hay un mensaje disponible.
- Recepción sin bloqueo. El receptor extrae un mensaje válido o un mensaje nulo.

Paso de Mensajes.

La comunicación tiene lugar mediante el intercambio de mensajes entre los procesos cooperativos

Memoria Compartida

La memoria compartida permite una velocidad máxima y una mejor comunicación, ya que puede realizarse a velocidades de memoria cuando se hace en una misma computadora.



Comunicación interprocesos



Los procesos que se ejecutan concurrentemente pueden ser procesos independientes o procesos cooperativos.

Proceso Independiente

Un proceso es independiente si no puede afectar o verse afectado por los restantes procesos que se ejecutan en el sistema. Cualquier proceso que no comparte datos con ningún otro proceso es un proceso independiente.

Útil para intercambiar pequeñas cantidades de datos, ya que no existe la necesidad de evitar conflictos. El paso de mensajes también es más fácil de implementar que el modelo de memoria compartida como mecanismo de comunicación entre computadoras.

Proceso Cooperativo

Un proceso es cooperativo si puede afectar o verse afectado por los demás procesos que se ejecutan en el sistema. Evidentemente, cualquier proceso que comparte datos con otros procesos es un proceso cooperativo.

¿Por qué permitir la cooperación entre procesos?

- Compartir información. Dado que varios usuarios pueden estar interesados en la misma información (por ejemplo, un archivo compartido), debemos proporcionar un entorno que permita el acceso concurrente a dicha información.
- Acelerar los cálculos. Si deseamos que una determinada tarea se ejecute rápidamente, debemos dividirla en subtareas, ejecutándose cada una de ellas en paralelo con las demás. Observe que tal aceleración sólo se puede conseguir si la computadora tiene múltiples elementos de procesamiento, como por ejemplo varias CPU o varios canales de E/S.
- Modularidad. Podemos querer construir el sistema de forma modular, dividiendo las funciones del sistema en diferentes procesos o hebras.
- Conveniencia. Incluso un solo usuario puede querer trabajar en muchas tareas al mismo tiempo. Por ejemplo, un usuario puede estar editando, imprimiendo y compilando en paralelo.

Requiere mecanismos de comunicación interprocesos (IPC, interprocess communication) que les permitan intercambiar datos e información

Memoria Compartida

Se establece una región de la memoria para que sea compartida por los procesos cooperativos. De este modo, los procesos pueden intercambiar información leyendo y escribiendo datos en la zona compartida.

La memoria compartida permite una velocidad máxima y una mejor comunicación, ya que puede realizarse a velocidades de memoria cuando se hace en una misma computadora. La memoria compartida es más rápida que el paso de mensajes, ya que este último método se implementa normalmente usando llamadas al sistema y, por tanto, requiere que intervenga el kernel.

Proceso A

Proceso B

Kernel

Paso de Mensajes.

La comunicación tiene lugar mediante el intercambio de mensajes entre los procesos cooperativos

Útil para intercambiar pequeñas cantidades de datos, ya que no existe la necesidad de evitar conflictos. El paso de mensajes también es más fácil de implementar que el modelo de memoria compartida como mecanismo de comunicación entre computadoras.

Una facilidad de paso de mensajes proporciona al menos dos operaciones: envío de mensajes (send) y recepción de mensajes (receive). Los mensajes enviados por un proceso pueden tener un tamaño fijo o variable. Si sólo se pueden enviar mensajes de tamaño fijo, la implementación en el nivel de sistema es directa. Sin embargo, esta restricción hace que la tarea de programación sea más complicada. Por el contrario, los mensajes de tamaño variable requieren una implementación más compleja en el nivel de sistema, pero la tarea de programación es más sencilla

Desventajas Esquemas Simétricos y Asimétrico

La desventaja de estos dos esquemas (simétrico y asimétrico) es la limitada modularidad de las definiciones de procesos resultantes. Cambiar el identificador de un proceso puede requerir que se modifiquen todas las restantes definiciones de procesos. Deben localizarse todas las referencias al identificador antiguo, para poder sustituirlos por el nuevo identificador.

Sincronización

La comunicación entre procesos tiene lugar a través de llamadas a las primitivas send() y receive(). Existen diferentes opciones de diseño para implementar cada primitiva. El paso de mensajes puede ser con bloqueo o sin bloqueo, mecanismos también conocidos como síncrono y asíncrono.

• Envío con bloqueo. El proceso que envía se bloquea hasta que el proceso receptor o el buzón de correo reciben el mensaje.

• Envío sin bloqueo. El proceso transmisor envía el mensaje y continúa operando.

• Recepción con bloqueo. El receptor se bloquea hasta que hay un mensaje disponible.

• Recepción sin bloqueo. El receptor extrae un mensaje válido o un mensaje nulo.

Almacenamiento en búfer

Sea la comunicación directa o indirecta/ los mensajes intercambiados por los procesos que se están comunicando residen en una cola temporal.



• Capacidad cero.

La cola tiene una longitud máxima de cero; por tanto, no puede haber ningún mensaje esperando en el enlace. En este caso, el transmisor debe bloquearse hasta que el receptor reciba el mensaje.

• Capacidad limitada.

La cola tiene una longitud finita n; por tanto, puede haber en ella n mensajes como máximo. Si la cola no está llena cuando se envía un mensaje, el mensaje se introduce en la cola (se copia el mensaje o se almacena un puntero al mismo), y el transmisor puede continuar la ejecución sin esperar. Sin embargo, la capacidad del enlace es finita. Si el enlace está lleno, el transmisor debe bloquearse hasta que haya espacio.

• Capacidad ilimitada.

La longitud de la cola es potencialmente infinita; por tanto, puede haber cualquier cantidad de mensajes esperando en ella. El transmisor nunca se bloquea.

Almacenamiento Secundario

Intercambio/SWAP (L)

HD extraible, Memorias Flash NAND, Discos Ópticos (Blu Ray, DVD)

Almacenamiento en Red (SAN, NAS)

Cintas Magnéticas

Almacenamiento Secundario