



Universidad Gerardo Barrios

Unidad I - Sesión II - Sistemas Operativos



Introducción a los Sistemas Operativos

Durante el desarrollo de esta Unidad, se abordarán contenidos indispensables para la formación profesional de cada uno de los estudiantes que en ocasiones desconocen el funcionamiento interno de un sistema operativo , en esta asignatura comprenderá el significado de terminología básica, tipos de sistemas operativos, las capas por las que este compuesto, sus funciones internas y de su rol en la actualidad; Se profundizará en el estudio de los procesos internos detallando así la jerarquía que estos manejan y cómo son tratados por el sistema operativo para el manejo y procesamiento de datos del usuario, se conocerán las formas de comunicación y sincronización de procesos, el trabajo que realiza entre procesos padre e hijos y el rol de hilos según el funcionamiento del CPU; se estudiará la forma básica del manejo de dispositivos de entrada y salida en función de los controladores.

☰ Procesos: concepto, estado, jerarquía

☰ Comunicación, sincronización e hilos

☰ Recursos Complementarios

Procesos: concepto, estado, jerarquía



Concepto de procesos

Un proceso es todo aquel programa que se está ejecutando en un periodo de tiempo dentro una CPU y que con ayuda de la memoria RAM puede almacenar datos para su procesamiento, un proceso al final se puede definir como una unidad más pequeña y primordial que un SO gestiona.

Los procesos tienen características determinadas por una estructura que permite identificar los recursos asignados y controlar su orden de ejecución:

Identificador de proceso

PID como se conoce, es el identificar único por cada proceso, con esto se identifica el número de proceso que este en ejecución el número proceso al cual pertenecen los datos en memoria.

Identificador de usuario

Este es otro ID, pero enfocado en la pertenencia al usuario.

Estado de los registros

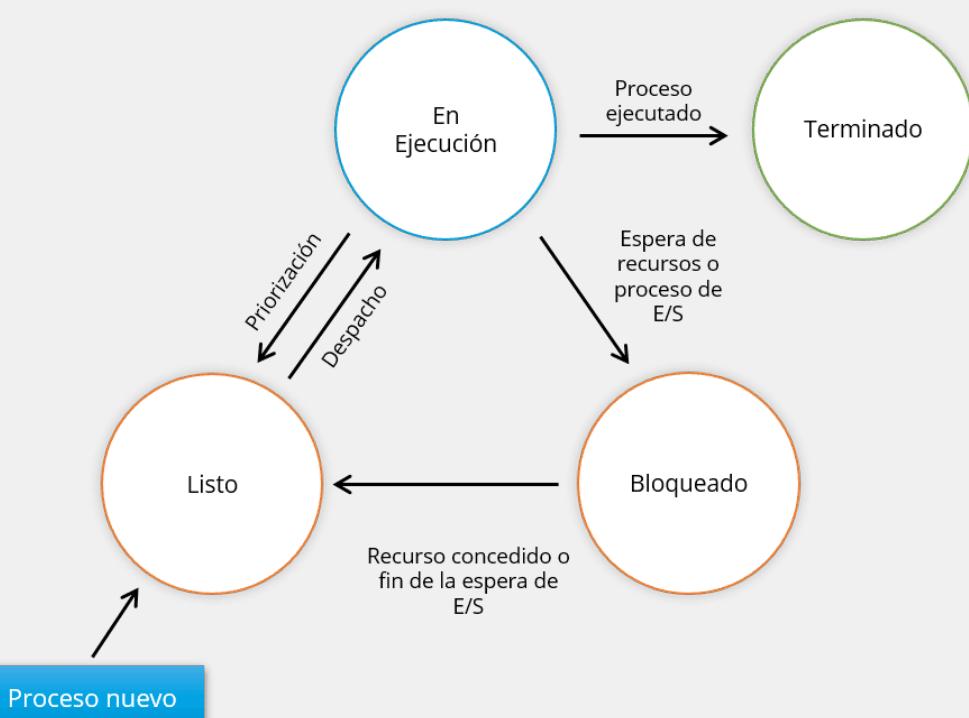
Muestran la situación real de los registros del procesador en función del proceso, es decir, se guarda información en cache sobre el proceso que se ejecutó al momento que este desee recuperar su espacio en procesamiento, vale aclarar que un proceso puede llegar a quedar inactivo, pero necesitará siempre mantener un respaldo de sus últimas acciones y datos generados.

Descriptores

Esta sección de la información de un proceso mantiene activa la información de los ficheros que el proceso está utilizando y que puertos de comunicación están abiertos para su uso.

Finalmente, un proceso puede definirse como instrucciones que deben ejecutarse con los recursos del procesador para calcular los datos, la RAM para almacenar los datos a procesar o procesados, así mismo los datos de cada proceso, es decir, el estado de los procesos.

Estado de los procesos



Jerarquía de procesos.

Los estados de los procesos son aquellos por los cuales un programa que se va a ejecutar o parte de un programa necesita para su correcta ejecución, se explican 4 fases o estados en los cuales se entiende el momento en el que un proceso puede estar en ejecución o bloqueada por algún motivo:

En ejecución

En ese momento el CPU está ejecutando instrucciones en el código del proceso.

Bloqueado

El proceso tiene que esperar hasta que se conceda una petición de recursos formulada por aquél, o hasta que acurra un evento específico. No debe asignársele un CPU hasta que su espera esté terminada.

Listo

El proceso decide usar el CPU para continuar su operación; sin embargo, no ha sido planificado.

Terminado

La operación del proceso, es decir, la ejecución del programa representado por aquél se ha terminado normalmente, o el sistema la ha abortado.

Continuar

Jerarquía de procesos

La jerarquía de procesos se define en base a los estados que un proceso puede tener y el orden de ejecución que este tiene según si PID, para revisar los estados del proceso revisar la ilustración anterior.

Listo - en ejecución

El proceso se despacha. El CPU inicia la ejecución de sus instrucciones.

Bloqueado - listo

Se satisface una petición hecha por el proceso u ocurre un evento que estaba esperándose.

En ejecución - bloqueado

El programa que está siendo ejecutado hace una llamada al sistema para indicar que desea esperar hasta que se satisfaga alguna otra petición de recurso hasta que ocurra un evento específico en el sistema. Las cinco principales causas de bloqueo son:

1. El proceso solicita una operación de E/S.
2. El proceso solicita memoria o cualquier otro recurso.
3. El proceso desea esperar durante un intervalo específico de tiempo.
4. El proceso espera el mensaje de otro proceso.
5. El proceso desea esperar alguna otra acción de otro proceso.

(Flores, 2020)

En ejecución - terminado

La ejecución del programa se completa o se termina. Las siguientes son cinco razones principales para la terminación del proceso:

- Auto terminación: el programa que está siendo ejecutado termina su tarea o descubre que no puede ejecutarse en forma significativa y hace una llamada de “termíname” al sistema.
- Terminación por un parent: un proceso hace una llamada de termíname al sistema para terminar el proceso hijo, cuando encuentra que ya no es necesaria o significativa la ejecución del proceso hijo.
- Utilización excesiva de recursos: un sistema operativo puede limitar los recursos que un proceso puede consumir. Un proceso que sobrepase un límite de recursos será terminado por el cerner.

- Condiciones anormales durante la ejecución: el cerner cancela un proceso si surge una condición anormal en el programa que se está ejecutando, como instrucción invalidado desbordamiento.
- La interacción incorrecta con otros procesos: el cerner puede cancelar un proceso por la interacción incorrecta con otros procesos, por ejemplo, si un proceso interviene en un bloqueo mutuo.

De todo esto se desprende algo llamado jerarquía de procesos donde se crea un orden de ejecución.

Existe un proceso de inicio original a partir del cual se crean los demás; se describe el listado de procesos como un árbol en el que un proceso originario da lugar a otros que a su vez crean más. Cuando un proceso A solicita que al sistema operativo la creación de otro proceso B, se dirá entonces que A es padre del proceso B, y que B es hijo del A (y el padre podrá si lo desea matar a su proceso hijo).

Referencias Bibliográficas

- Ramirez, M. (Febrero de 2015). www.scribd.com. Obtenido de <https://www.scribd.com/document/259659680/Semafaro-Ensayo>
- Ramos, B. (23 de abril de 2020). Obtenido de <https://es.scribd.com/document/292150022/Hilos-y-Sincronizacion-de-Procesos>
- Torres, M. (2020). manueltrrs.blogspot.com. Obtenido de <http://manueltrrs.blogspot.com/2015/05/unidad-1-software-software-educativo.html>

Comunicación, sincronización e hilos



Comunicación y sincronización de procesos

Las secciones críticas y la señalización son los elementos clave de la sincronización de procesos. La solución para un problema de sincronización de procesos deberá usar una combinación adecuada de estos elementos. También deberá poseer tres propiedades importantes:

- Corrección: es decir la verificación de flujo de información.
- Máxima concurrencia: gestionar los recursos de manera que no se vean afectados los datos.

- No existencia de esperas activas: evitar que existan demasiados procesos esperando ser ejecutado.

Los criterios de corrección dependen de la naturaleza de un problema. Estos proyectan requerimientos referentes a la sincronización de accesos a datos y a la sincronización de controles de procesos interactivos. La máxima concurrencia dentro de una aplicación es necesaria para acelerar la ejecución y obtener buen tiempo de respuesta. Para lograr esto los procesos deberán ser capaces de ejecutarse libremente, cuando no están bloqueados debido a requerimientos de sincronización.

Es de suma importancia conocer y estudiar los mecanismos de comunicación y sincronización de procesos, ya que este tema ayuda a comprender mejor cada proceso que efectúa un sistema operativo, y ayuda a entender y conocer cada mecanismo por el que el proceso tiene que pasar cuando una instrucción es enviada por el usuario y como es la comunicación entre procesos. En el presente trabajo de investigación se estudiarán los mecanismos de comunicación y sincronización de procesos, aquí se presentará lo que estos hacen posible en los diferentes procesos que ejecuta un sistema operativo, y los principales mecanismos de comunicación, y la funcionalidad de cada uno de ellos.

Sincronización de procesos

En general se habla de sincronización de procesos cuando un proceso p1 requiere que un proceso p2 realice alguna acción para continuar. Si cuando el proceso p1 realiza su comprobación el proceso p2 hubiera realizado la acción, el proceso p1 sigue su ejecución. Si el proceso p2 no hubiera realizado dicha acción, el proceso p1 es bloqueado, para a la situación de espera y es puesto en cola de procesos que esperan dicho acontecimiento, si es que hay algún otro proceso que prestaba ya esperándolo. Cuando el proceso p2 realiza la acción, el proceso p1 pasa a la situación e preparado para ejecución. Se describen dos herramientas para la sincronización de procesos que se denominan semáforos y eventos. (Ramos, 2020)

Semáforos

Esta técnica es de las más generales y utilizadas, es una herramienta de control del tiempo de ejecución para la administración de procesos, funciona como un semáforo real llevando un contador para permitir la ejecución de un proceso y manteniendo en cola una serie de procesos que luego serán ejecutados.

Las operaciones básicas sobre los semáforos son:

ESPERA

SEÑALIZADOR

Esta lógica responde al rojo y verde habitual de los semáforos reales, pero en este caso se trabaja con valores booleanos es decir 0 y 1, cuando un semáforo tiene su variable mayor a cero permitirá la ejecución de los procesos en el caso de que la variable de estado sea cero no permitirá ejecución de ningún proceso de su cola.

ESPERA

SEÑALIZADOR

Consiste en el proceso básico de etiquetar un orden ejecución para cada proceso, formando una cola ordenada. Utiliza un acumulador donde incremente en uno la variable de estado del semáforo cuando alguna operación se realiza sin procesos detenidos.

Eventos

Son mecanismos para sincronizar procesos con dos funciones básicas a disposición para identificar procesos en ejecución y procesos en cola:

Bloqueado

Cuando un proceso queda en estado bloqueado por cualquier motivo, ya sea por la espera de asignación de recursos o espera de datos, es añadido a la cola para que otro pueda ejecutarse, es similar al estado bloqueado visto en la jerarquía de proceso.

Desbloqueado

Es la acción de ejecutar los procesos cuando evento se desencadena este evento puede ser la respuesta a la petición de recursos y espera de datos.



Comunicación de procesos

En general, se habla de comunicación de procesos cuando un proceso requiere intercambiar información con otro proceso. Para que dos procesos puedan intercambiar información deberán sincronizarse para que el proceso receptor espere por el emisor en el caso de que este no haya enviado el mensaje o viceversa.

Las operaciones para la comunicación de procesos que se describen a continuación suponen la existencia de un almacenamiento intermedio (buzón) para almacenar los mensajes enviado y no recibidos. Se describe a continuación la forma de comunicación de sistemas operativos. (Ramos, 2020)

Crear buzón

Todo buzón tiene asociada una cola de procesos detenidos en el buzón y una cola de mensajes pendientes de recibir. En la cola de procesos detenidos en el buzón estarán todos los procesos que hayan intentado extraer mensajes del buzón, cuando el buzón estaba vacío. En la cola de mensajes del buzón estarán todos los mensajes que han llegado al buzón y que aún no han sido recogidos por ningún proceso.

Cuando se realiza una llamada para crear un buzón se crea una estructura de datos asociadas con el buzón rellenando sus campos, se inicializa la cola de mensajes del buzón y la cola de procesos detenidos en el buzón. (Ramos, 2020)

Enviar buzón

Esta operación permite que un proceso envíe un mensaje a un buzón existente en el sistema.

Al enviar un mensaje a un buzón se comprueba si existen procesos detenidos en la cola de procesos del buzón, se selecciona al primero de la lista, es autorizado a continuar y envía el mensaje. El envío de mensajes no puede producir la detención del proceso que realiza la llamada a no ser que no exista espacio en el buzón para colocar mensajes. (Ramos, 2020)

Recibir

Esta operación permite que un proceso retire un mensaje de un buzón y que dicho mensaje sea recibido por el proceso que realizan la llamada. Al realizarse la llamada se comprueba si existe algún mensaje en la cola de mensajes del buzón y en caso afirmativo se copia el mensaje en el proceso llamador y se cede el control a uno de los procesos preparados. Se puede producir un error al intentar recibir un mensaje de un buzón. (Ramos, 2020)

Continuar



Hilos

Los hilos son un método para evitar sobre carga de procesos, creando pequeños subprocessos que necesitan menos tiempo en ejecución y generan datos más pequeños lo que se traduce en una mayor velocidad y menos pérdida de tiempo si se ejecutaran procesos completos.

Al no ejecutar hilos se crearía algo denominado sobrecarga y esta puede ocurrir de dos formas:

Sobrecarga relacionada con la ejecución

Íntimamente relacionado con el tiempo que necesitará la CPU para procesar los datos de un proceso sin hilos.

Sobrecarga relacionada con el uso del recurso

Esta sobrecarga se orienta la utilización de recursos en gran tamaño y que tiene por consecuencia llevar información del estado del proceso también en volúmenes grandes.

Entonces se define un hilo como:

La ejecución de un subprocesso derivado de un proceso principal, siendo este mucho más pequeño en datos a generar y necesidad de recursos, posee al igual que el proceso padre, su propio PID y estado del CPU.

Cuando un hilo es creado, se le pone en el estado listo porque su proceso padre ya tiene los recursos necesarios asignados. Entra en el estado de ejecución cuando es planificado. No entra en el estado de bloqueo por las peticiones de recursos, porque no hace ninguna petición de recursos, sin embargo, puede entrar en el estado bloqueado debido a los requisitos de sincronización de proceso.

Las ventajas de los hilos

SOBRECARGA BAJA

ACELERACIÓN

COMUNICACIÓN EFICIENTE

Esto se da debido a que el proceso padre ya posee los recursos asignados y el hilo del proceso ya no necesita soltarlos, es decir que el hilo es creado y pase directamente a un estado listo, para ser ejecutado.

SOBRECARGA BAJA

ACELERACIÓN

COMUNICACIÓN EFICIENTE

Debido a que existe baja sobrecarga y que los hilos son procesos más ligeros y con pequeña necesidad de procesamiento la ejecución de un proceso global puede ejecutarse mucho más rápido que ejecutar los grandes bloques de datos.

SOBRECARGA BAJA

ACELERACIÓN

COMUNICACIÓN EFICIENTE

Los recursos asignados al proceso padre y la cache ya disponible para el proceso global es aprovechada por los hilos cuando necesitan compartir datos con otros hilos, siendo así que la comunicación se vuelve mucho más eficiente.

Implementación de hilos

Los hilos en el nivel del Kernel

Estos son hilos que al igual que un proceso tienen su propio estado de ejecución, pero con datos mucho más pequeños, lo que para los programadores es muy conveniente porque se pueden ejecutar como procesos normales y hacer el llamado a ellos con toda naturalidad, tal cual se tratara de un proceso padre, por lo que pueden ser planificados en qué momento serán ejecutados.

El manejo de hilos en este nivel tiene una pequeña limitante y es que el Kernel se saturara de eventos que ejecutaran cada hilo y cada proceso padre, esto se conoce como sobrecarga de eventos lo que resultaría en una difícil administración para sincronizar y comunicar hilos y procesos.

Hilos en el nivel de usuario

Esta implementación de hilos utiliza una biblioteca de hilos enlazada a un PID o código de ejecución de un proceso padre, en esta biblioteca se establecen los procesos de sincronización y comunicación de cada hilo, evitando así una sobrecarga alta de eventos y administración de procesos, por esto un proceso podrá ejecutarse según su naturaleza y sus datos estableciendo las prioridades según el tipo de dato y aplicación que se está ejecutando.

Una de las desventajas es que el kernel no sabe la diferencia entre hilo y un proceso, por lo que si un hilo se bloqueara es una llamada de sistema, el kernel bloquearía su proceso padre. Los hilos no deben hacer llamadas de sistema que pueden inducir al bloqueo.

Modelo de hilos híbridos

Este modelo tiene ambos tipos de hilos tanto a nivel de usuario como de kernel, y un método y asociar unos hilos con otros. Los métodos diferentes para realizar lo anterior proveen combinaciones diferentes de la sobrecarga de baja comutación de los hilos en el nivel del usuario y el paralelismo y la concurrencia altas de los hilos en el nivel del kernel.

Referencias Bibliográficas

- Ramirez, M. (Febrero de 2015). www.scribd.com. Obtenido de <https://www.scribd.com/document/259659680/Semafaro-Ensayo>
- Ramos, B. (23 de abril de 2020). Obtenido de <https://es.scribd.com/document/292150022/Hilos-y-Sincronizacion-de-Procesos>
- Torres, M. (19 de abril de 2020). Sistemas Operativos. Obtenido de <http://manueltrs.blogspot.com/2015/05/unidad-1-software-software-educativo.html>

Recursos Complementarios



Universidad Gerardo Barrios

| Recurso | Título | Cita Referencial |
|-----------|--|--|
| PDF | Sistemas Operativos | (Núcleo Alberto Adriani, 2020) |
| PDF | Importancia de los Sistemas Operativos | (Gonzalez, 2020) |
| Sitio WEB | Hilos y procesos | (Álvarez, 2020) |
| PDF | Sincronización de procesos. | (Ramirez, 2015) |