Unidad II (Procesos)

martes, 16 de mayo de 2023 07:16 p. m.

-

CLASE 3:

¿ Cuáles son las características de un proceso?

Un proceso es un programa en ejecución que se puede correr en el procesador, presenta estado actual, secuencia de instrucciones y un conjunto de instrucciones asociadas al proceso.

Elementos:

- Identificador: ID del proceso
- Estado: condición actual del proceso **(Ejecución - Terminado - Bloqueado)**
- Prioridad: prioridad que presenta el proceso con respecto a los demás
- Program Counter: registro que almacena la dirección de la siguiente instrucción a ejecutar
- Memoria de Punteros: área de memoria que contiene punteros para permitir la correcta administración de recursos para el proceso
- Context Data: información sobre el contexto de ejecución del proceso
- Estado de información de E/S: información sobre las operaciones de Entrada y Salida
- Accounting information: datos relacionados en base a la utilización de recurso del proceso

¿Cuáles son los componentes de un proceso?

- Programa Ejecutable
- Datos Asociados al programa como variables y buffers
- Contexto de Ejecución: información que se requiere para administrar el proceso, contenido de registros, PC, registro de datos.

¿Qué problemas podrán causar n procesos ejecutándose al mismo tiempo en una misma CPU?

1. Sincronización Incorrecta: esto se debe al mal manejo de interrupciones o un mal manejo de señales.
2. Funcionamiento no determinista: la salida debe depender de la entrada y no de actividades realizadas.
3. Fallos de Exclusión Mutua: 1 solo proceso por vez tome los recursos necesarios
4. Interbloqueos: dos o más programas suspendidos, abrazo mortal.

¿Qué ventajas cree conveniente que acarrea el hecho de trabajar con memoria virtual?

Mayor capacidad de memoria que la que se utiliza físicamente, además permite una administración eficiente de los recursos de la memoria.
Mayor capacidad de protección y seguridad para los procesos en ejecución.
Mayor flexibilidad y compatibilidad, permite la ejecución de programas que requieren más memoria que la que se encuentra físicamente disponible.

Procesos / Hilos

Diferencia entre HILO y PROCESOS

Un proceso presenta su propio espacio de direcciones y ejecución (registros de CPU, program counter, pila). Cada proceso se ejecuta de manera independiente y aislada de los demás procesos. En cambio, los hilos comparten el mismo espacio de memoria permitiendo una comunicación entre ellos.

Los procesos son creados por el SO y pueden iniciar y finalizar de manera independiente. Los hilos se crean dentro de un proceso existente y se ejecutan de manera concurrente.

Debido a compartir el mismo espacio de memoria se debe tener más precaución en la modificación de datos, en cambio los procesos al tener espacio de memoria propio presentan mayor seguridad.

Diferencia entre MICROKERNEL y MULTIHILO

- **MicroKernel:** busca minimizar el núcleo, dejando las funciones esenciales dentro del núcleo, como gestión de memoria, comunicación y planificación de la CPU.
  Las demás funciones se trasladan a módulos externos servidores /subsistemas

- **MultiHilo:** capacidad del SO de ejecutar multiples hilos de forma concurrente dentro de un proceso. Esto permite un paralelismo y mejora el rendimiento de las respuestas, ya que al presentar una misma memoria se pueden generar una comunicación más rápida.

Multiprocesamiento Simétrico (SMP):

Existen varios procesadores que pueden ejecutar el mismo set de instrucciones

Permite que varios procesadores trabajen juntos de manera simétrica en la ejecución de tareas. Los procesadores comparten la memoria principal y los dispositivos de E/S.

Características:

1. Simetría: procesos idénticos, misma capacidad de ejecución de instrucciones.
2. Memoria Compartida: comparten la misma memoria principal.
3. Tareas concurrentes ejecutadas en distintos procesadores.

Usuario UNIX:

Capas entre el usuario y el hardware:

- Unix comandos y librerias
- Interfaz de llamadas al sistema
- Kernel
- Hardware

¿Por qué razón Ud. cree que los diseñadores han optado por un modelo por capas? ¿Qué ventajas cree Ud. que trae el programar de esta forma? ¿Podemos utilizar esta idea en la construcción de software aplicativo?

Este tipo de desarrollo requiere de un objetivo específico que genera interfaces bien definidas permitiendo la especificación de cada capa y la facilitación en cuanto al trabajo en grupo, distribuyendo cada capa a un desarrollador.

Asimismo pueden corregirse y actualizarse de manera separada sin interferir en el resto de las capas.
En cuanto al desarrollo se puede aplicar a cualquier tipo de software.

Linux no utiliza la arquitectura de micro-núcleo (microkernel), describa brevemente cuál es la idea de su implementación en contraposición con la arquitectura de un UNIX tradicional.

Linux utiliza una arquitectura modular, estos módulos se cargan en tiempo de ejecución y se pueden vincular y desvincular.

Los módulos son apilables de forma jerárquica, permitiendo la implementación de un módulo en n códigos, reduciendo la cantidad de código y en consecuente de memoria. Además genera módulos eficientes y específicos.

CLASE 4:

¿Qué es la traza de un proceso?

¿Por qué razón el modelo de dos estados no es apropiado para describir

El termino se asocia al comportamiento del proceso, secuencia de instrucciones que este mismo ejecuta. La misma depende del flujo del proceso.

PROCESS SPAWNING

Se le denomina al proceso cuando crea a otro proceso a través de un fork(). Mediante esta operación se realiza la creación de procesos pesados.

SWAPPING

Movimiento de un proceso en memoria principal a memoria secundaria. SWAP es el área de intercambio entre ambas memorias.

Diferencia entre BLOCKED y BLOCKED/SUSPEND

BLOCKED: proceso bloqueado en memoria principal.

BLOCKED / SUSPEND: proceso bloqueado en memoria secundaria-

Suspensión de Procesos (Razones)

1. Realización de swapping
2. Liberar memoria principal
3. Interbloqueo dentro del proceso
4. Decisión de usuario
5. Temporización de proceso
6. El proceso padre puede decidir suspenderse mediante un wait esperando la finalización del proceso hijo/s.

lo que realmente pasa en los procesos?

Este modelo no es apropiado ya que asume que los procesos en estado Not Running pueden pasar a estado Running si es seleccionado por el **Dispatcher**. Pero esto no siempre es posible ya que un proceso puede estar en espera frente a la finalización de una operación de E/S.

coordinar y administrar la ejecución de los proceso

TIME OVERRUM

Tiempo determinado que se le da a un proceso para que realice un evento. Una vez que el proceso de espera finaliza el proceso se termina.

Estados de un Proceso

- **Running:** proceso en ejecución
- **Ready**: proceso preparado para ejecutarse
- **Blocked / Waiting:** proceso bloqueado o esperando por la finalización de otro para su ejecución
- **New:** proceso recién creado pero que no se incluyó dentro de la Pool de procesos.
- **Exit**: proceso que se sacó de la pool de procesos ejecutables.

CLASE 5:

¿Por qué es necesario para el SO mantener estructura de control de procesos?

Es necesario para que se pueda realizar una raza de procesos y administrar el estado de cada uno.

Dentro de un bloque de Control de Procesos podemos encontrar información como identificación del proceso, información del estado de la CPU e información del control de proceso.

Imagen de un Proceso

Representación del estado y de los recursos asociados a un proceso en determinado momento.

¿Por qué se realiza un PROCESS SWITCHING?

1. Porque se realiza una interrupción externa al proceso de ejecución.
2. Ocurre una causa interna en el proceso de ejecución
3. Ocurre una llamada al SO

Operación en donde se produce un cambio en la ejecución de un proceso activo a otro proceso. Se guarda el estado del proceso en ejecución y datos del mismo.

¿Por qué se realiza un cambio de Modo?

Se cambia de modo de usuario a kernel debido a la necesidad de ejecutar instrucciones privilegiadas y en caso contrario para ejecutar instrucciones no privilegiadas en modo usuario.

Diferencia entre cambio de Proceso y cambio de Modo

Cambio de Modo: afecta a las instrucciones que se pueden ejecutar pero no cambia el proceso. **(MODO NUCLEO - MODO USUARIO)**.

Cambio de Proceso: implica la transferencia del uso de CPU de un proceso en ejecución a otro.

Diferencia entre TRAP e INTERRUPCIÓN

Ambos permiten al SO manejar eventos especificas

TRAP: excepciones internas generadas por el programa en ejecución que se dan para solicitar servicio o actuar sobre una condición de error

INTERRUPCIÓN: eventos externos generados por dispositivos de hardware, se hacen para notificar al procesador sobre la ocurrencia de un evento o atención.

Describir:

- **Proceso Swaper / Proceso 0:** Estructura de datos que se carga en memoria cuando el sistema arranca.
- **Proceso Init / Proceso 1:** se considera que es el padre de todos los procesos, genera luego del proceso Swaper.

¿Qué es un proceso Zombie?

Un proceso zombie es un proceso que se terminó de manera normal o anormal, antes de que el padre realice el waitpid para tomar la información del proceso. La información del proceso queda en el sistema y hasta que el padre la recupera el proceso continua en estado zombie. Solo cuando el padre obtiene la información de finalización del proceso, este dejará de ser zombie.

mecanismo que permite a los prog. de usuario solicitar servicios o realizar operaciones

¿Para qué sirve la llamada al sistema (system call) fork()? ¿Cómo funciona?

La función fork() se utiliza para crear un proceso

En la creación del proceso se realiza las siguientes funciones:

- Solicita nueva entrada en la tabla de procesos
- Asigna ID al proceso
- Copia imagen del proceso padre
- Incrementa el contador de ficheros, ya que el hijo puede acceder a los ficheros
- [illegible]ser ejecutado
- Devuelve el id al proceso padre y cero al hijo

CLASE 6:

Diferencias entre "MonoHilo" y "MultiHilo"

MonoHilo: refiere a un único hilo, ejemplo: MS-DOS

MultiHilo: refiere a que el SO da soporte a n hilos.

Definir que es un Hilo

Un hilo es una unidad de ejecución consume CPU y tiene PC propio. Es una secuencia de instrucciones que se ejecuta de forma independiente dentro de un proceso.

¿Por qué un cambio de contexto entre procesos es más "pesado o lento" que un cambio entre hilos?

Para cambiar entre procesos se necesita la intervención del SO para que guarde el contexto de ejecución y ponga a ejecutar el nuevo proceso a ejecutar. En cambio un hilo comparte parte del contexto del proceso, entonces podemos intercambiar entre hilos sin necesidad de la intervención del SO.

Es más rápido crear un hilo que hacer un fork() por esto mismo en una circunstancia de atención al cliente nos conviene hacer hilos.

¿Qué sucede con los hilos del proceso X cuando el SO suspende al proceso X?

El SO no hace gestión en el nivel de hilos, la planificación se realiza a nivel de procesos. Si se suspende el proceso, todos los hilos del proceso se suspenden.

¿En un ambiente **ULT (User LEVEL THREAD)**, quien está a cargo de la planificación de los hilos?

La programación está a cargo el usuario ya que se utilizan librerías o bibliotecas para la creación de hilos. *Ejemplo: uso de biblioteca pthread.*

¿En un ambiente **KLT (Kernel LEVEL THREAD)**, quien está a cargo de la planificación de los hilos?

En este caso el que está a cargo de la planificación de hilos es el sistema operativo.

Relación entre hilos y procesos

1:M en donde un hilo puede migrar de un entorno de proceso de otro, esto permite que los hilos puedan moverse en los distintos sistemas. *Ejemplo: Esmerald.*

¿Por qué se dice que ULT no saca provecho de la multiprogramación **(varios programas cargados en memoria al mismo tiempo)**

Esto se debe a que muchas llamadas al sistema son bloqueantes, si un hilo hace una llamada bloqueante, todo el proceso queda bloqueado. Además como ULT es a nivel proceso y no de hilo el SO no puede asignar el mismo proceso a más de un CPU y de esta forma no se estaría cumpliendo el aprovechamiento de Multiprogramación.

Para poder solucionar la respuesta anterior se puede utilizar el concepto de **Jacketing,** la cual determina un envoltorio de una llamada bloqueante para transformarla en no bloqueante.

CLASE 7:

¿Cuáles son los 4 sistemas que existen según taxonomía de Flynn?

SISD (MONOPROCESADOR)

MISD, SIMD, MIMD : Sistemas que soportan procesamiento en paralelo

¿Un SMP es un MIMD con memoria compartida en el cual el núcleo se ejecuta en CPU determinada?

No, un SMP es un MIMD fuertemente acoplado pero el núcleo se ejecuta en cualquier CPU.

Diferencia entre SMP y Cluster:

- **SMP**: Tiene memoria compartida o fuertemente acoplada, es compartida por n CPU's que se encuentran dentro de un mismo bus y el mismo computados.
- **Cluster:** Tiene memoria débilmente acoplada ya que los computadores son distintos y cada uno presenta CPU's distintas y memorias propias.

El problema de **coherencia de cachés** debe ser resuelto por el hardware.

¿En un sistema SMP, si falla una CPU entonces falla todo el sistema?

Esto no ocurriría ya que el Kernel debería ejecutarse en cualquier CPU, el sistema debe tolerar este tipo de fallos y volver a ejecutarse.

¿Por qué se dice que un núcleo SMP es "reentrante"?

Esto se debe a que un miso código pueda estar ejecutándose al mismo tiempo en más de una CPU.

Algoritmo para la sincronización de un SMP:

El problema principal que se encuentra es que existen n CPU'S accedieron al mismo tiempo a un mismo conjunto de recursos.

=> hilo de nivel usuario (administrado desde una biblioteca o librería)

Los algoritmos que se pueden utilizar son los de Cerrojos, Exclusión Mutua y Candados.

¿Micronúcleo implica SO pequeño?

Los sitemas operativos

¿SO por capas es igual que monolítico?

No, son distintos la principal diferencia radica en la organización interna. Un SO por capas es estructurado y se divide en capas lógicas definidas donde cada una presenta funcionalidades específicas.

Las capas se construyen en base a la capa inferior y las capas inferiores utilizan servicios de las capas inferiores.
En cambio, un sistema monolítico es un enfoque donde las funcionalidades y servicios se encuentran integrado en un mismo bloque de código. Comparten el mismo espacio de memoria.

¿El Micronúcleo corre en modo kernel o núcleo, al igual que el resto del SO?

Solo el micronúcleo corre en modo kernel el resto del sistema operativo lo hace en modo usuario.

¿La implementación de IPC (inter process communication) está bajo la responsabilidad del micronúcleo o es algo externo a él?

IPC se encuentra bajo la responsabilidad del micronúcleo pero solo en cuanto a la forma elemental de comunicación entre procesos.

¿Qué relación hay entre la arquitectura client-server y el micronúcleo?

Un micronúcleo es un conjunto de procesos servidores que dan inicio a un conjunto de procesos clientes, la relación es directa. Los procesos servidores validan los mensajes recibidos de los clientes y los direccionan.

Por qué razón se dice que hay un potencial problema de rendimiento en la implementación de un SO con micronúcleo? ¿Cuál sería una posible solución?

Por el envío, recepción y decodificación de mensajes entre procesos es más lento que realizar una llamada al sistema como se podría hacer en un SO con arquitectura no micronúcleo

Más pequeño el micronúcleo menos tardará el traslado de mensajes y su decodificación.

Explique por qué un SO con micronúcleo facilita la implementación de Sistemas Operativos Distribuidos

Porque si cada mensaje enviado al SO incluye un ID esto permitirá diferenciar entre cada uno y obtener una imagen única del servicio a nivel micronúcleo.

¿Qué ventajas obtengo cuando los hilos de usuarios están asociados con hilos del núcleo y todos comparten el mismo id de grupo de proceso?

La asignación de recursos se hace por grupo de procesos, si un hilo a nivel usuario se asocia con un hilo a nivel kernel ambos pueden tener acceso a los mismos recursos, evitando un cambio de contexto.

SISTEMAS DISTRIBUIDOS

Los sistemas operativos distribuidos son sistemas operativos diseñados para ejecutarse en entornos distribuidos, donde múltiples computadoras o nodos están interconectados y trabajan juntos para realizar tareas y proporcionar servicios