Julia López - clase 8 - mochila del viajero

Procesos
Los procesos son la ejecución de un programa o instrucción. Es cuando la info del programa se carga en memoria y se pone en ejecución.
Todos los software ejecutables se organizan en procesos que quieren usar la cpu, y es el sistema operativo quien organiza el orden en el que se van ejecutando los procesos. Estas acciones que el sistema operativo realiza para cambiar el proceso A por el B se denominan **cambio de contexto**. Los procesos se ejecutan uno a la vez por más de que parezca que se ejecutan al mismo tiempo. Los mismos no pueden almacenarse en la memoria principal (RAM) porque consumen espacio y la llenarían. Por eso son efímeros, se crean y se terminan.
Pueden crearse:

1. de manera interactiva con el usuario. Por ejemplo cuando exportamos un archivo, estamos generando un proceso.
2. en segundo plano, como llamadas al sistema operativo (SO) por ejemplo cuando un software no puede acceder directamente a un recurso entonces le solicita al SO que lo gestione.

Para que un proceso termine, tiene que pasar por algún estado que determine su condición:

1. Nuevo: cuando el proceso se crea.
2. Listo: cuando está preparado, admitido y en condiciones de ser ejecutado, es decir cuando el sistema operativo los carga en la memoria.
3. Ejecución: cuando se empieza a ejecutar. Aquí puede cambiar el contexto, es decir ser suspendido cuando hacemos un llamado para leer el archivo. Se hace una interrupción.
4. Bloqueado: cuando está esperando que un recurso o proceso pueda ser utilizado.
5. Salida: cuando el proceso es ejecutado, pasa al estado de salida dando lugar a que otro proceso sea ejecutado.

Tipos de procesos:

- Procesos independientes: tienen total y completa autonomía. No pueden ser afectados ni afectar a otros procesos que se estén ejecutando en el sistema.
- Procesos cooperativos: sí pueden afectar y ser afectados. Cualquier tipo de proceso que comparta datos o cualquier recurso con otros procesos es considerado cooperativo. La cooperación se realiza porque algunos procesos carecen de información y deben consultarla para poder ejecutarse. Además, gracias a que la información es compartida, el CPU trabaja de una manera más eficiente y rápida. Esto da como resultado la modularidad: cuando una tarea tiene varios pasos, el CPU puede ejecutarlos de manera independiente y simultánea. Si alguna área tiene errores o no hay comunicación esto puede traer problemas. Un recurso puede estar bloqueado por otro proceso, como cuando mandamos a imprimir y da error.

Métodos de intercomunicación o IPC (inter-process communication): son mecanismos de comunicación entre procesos. Existen dos:

1. La **memoria compartida**. Se establece un espacio de memoria que será compartido por varios procesos. Esta es un recurso compartido a disposición de los software para que puedan intercambiar información. Dos procesos pueden estar realizándose con la memoria compartida y al mismo tiempo pueden estar

intercambiando información. Cada proceso comparte la info con otro, que va haciendo copia y modificando la info pero no sobrescribe. Es más económica que usar un multiprocesador. No existen los errores como exclusión mutua y son compatibles con cualquier tipo de arquitectura de computadora.

2. **Los pasos de mensaje**: un proceso encuentra información que le sirve, pide permiso al sistema para copiarla, el sistema se contacta con el otro proceso y le pide el permiso. La tarea de intermediario la lleva a cabo el kernel.

Las **señales**, las cuales son un aviso que puede enviar un proceso a otro. Luego, el sistema operativo se encarga de que el proceso que reciba la señal tome una acción para gestionarla.
Cuando se da un proceso que no puede resolverse instantáneamente, como cuando ocurre una llamada al sistema se van a crear otros procesos que se denominan **hijos**. Su función es realizar subtareas para lograr que el proceso padre pueda cumplir su objetivo. Los procesos padres pueden tener varios procesos hijos pero no al revés.

Sincronización de procesos
Existe una herramienta para la sincronización de procesos y es un semáforo. Mientras un proceso se está ejecutando y aparece una llamada de espera, pasa a una lista de bloqueados. Se queda ahí hasta que un proceso diferente le envía la señal de avance y el proceso bloqueado se coloca en una fila de espera para utilizar el cpu.

Área crítica
Es importante planificar el uso del cpu para que la cola de procesos no colapse ni llegue a la inanición, es decir que funcione de forma tan deficiente que le negaría recursos a otros procesos. Es la porción de código de un programa de ordenador en la que se accede a un recurso compartido (estructura de datos o dispositivo) que no debe ser accedido por más de un proceso o hilo en ejecución. La sección crítica por lo general termina en un tiempo determinado y el hilo, proceso o tarea sólo tendrá que esperar un período determinado de tiempo para entrar. Se necesita un mecanismo de sincronización en la entrada y salida de la sección crítica para asegurar la utilización en exclusiva del recurso, por ejemplo un semáforo. Existen técnicas para evitar estos problemas:

- FIFO: first in first out. Se asigna tiempo del cpu al primer proceso que lo solicite.
- SJF: shortest job first. El mejor trabajo primero, es decir, se ejecuta primero el proceso que tenga menor tiempo de ejecución.
- SRTF: shortest remaining time first: si se está ejecutando un proceso y llega uno segundo que requiere menos tiempo que lo que le queda al primero, este se interrumpe y se ejecuta el segundo. Es muy eficiente, puede ser injusta ya que un proceso corto puede echar a uno largo que esté haciendo uso del procesador y que además esté terminando, presenta mayor sobrecarga.
- Round Robin: existe una porción de tiempo establecida o cuantum de tiempo en donde los procesos se ejecutan en el CPU a medida que llegan a la fila de espera hasta que el cuantum se cumple. Cuando se termina, se interrumpe el proceso y si aún le quedan cosas por ejecutar, vuelve a la cola ubicándose al final. Esto distribuye los tiempos de manera equitativa.
- Otras planificaciones: tienen formas híbridas o diferentes. Ejemplo: retroalimentación multinivel o planificación por comportamiento.

Hilos

Procesadores: son un conjunto de transistores configurados de cierta forma que realizan operaciones binarias a partir de impulsos el;ecticos. Contienen un nucleo o más de uno. A mayor cantidad de núcleos, más procesos en paralelo se podrán ejecutar.
Proceso: conjunto de operaciones que componen un programa. Comparten el uso del procesador.
Los hilos son divisiones de esos procesos en seuencias de tareas. Son porciones de código que pueden ejecutarse de forma simultánea en cooperación con otros subprocesos. Multiples hilos pueden existir dentro de un proceso ejecutándose de manera concurrente, compartiendo recursos y memoria. Los procesos NO comparten recursos, los hilos SÍ. Es muy importante la sincronozación para que un hilo no bloquee a otro. Hasta los 2000 los procesadores eran monolíticos por lo que solo podían trabajar con un hilo a la vez. Luego aparecieron los multinúcleos que trabajan con varios hilos y aumentan la velocidad de procesamiento.

Los procesadores de un solo núcleo tienen una capacidad de respuesta menor, su comportamiento es más predecible, no presentan los errores que podrían presentar los multihilos y presentan menos errores.
Los procesadores multihilos tienen una exelente capacidad de respuesta, trabajan en paralelo las tareas, la sincronización es compleja y su comportamiento es dificil de predecir, puede presentar errores.