PROCESOS

Cuando queremos que nuestra computadora realice una instrucción son los **procesos** los que asumen el trabajo, para que, así, la **CPU** ejecute la tarea por medio del plan de acción diseñado por el **sistema operativo**.

Definición
Es la ejecución de un programa o instrucción
Conjunto de operaciones que componen a un programa, los cuales, a la hora de ejecutarse, se reparten la utilización del procesador para realizar su tarea

información del programa ---> memoria ---> procesos

- Todos los software ejecutables se organizan en procesos que quieren utilizar la CPU
- El SO organiza el orden de ejecución de los procesos
- Al cambio de procesos se le llama **cambio de contexto**
- Los procesos se ejecutan uno a la vez
- Los procesos no pueden almacenarse en la memoria principal, ya que consumen espacio y, a la larga, terminarían por llenar la memoria RAM. Por esto, todos los procesos son **efímeros** (se crean y se terminan)

¿Cómo se crean?

- **Proceso de manera interactiva con el usuario**
  - Cuando exportamos un archivo
- **Procesos de llamados al SO**
  - Se crean en segundo plano
  - Cuando un software no puede acceder directamente a un recurso, le solicita al SO que lo gestione

¿Cómo se terminan?

- Los procesos pueden transitar por diferentes estados que indican en qué parte de su ciclo de vida se encuentra y, con base en esto, el SO toma decisiones sobre él (se determina su condición)
  - Nuevo
    - Cuando el proceso se crea
  - Listo
    - Cuando el SO lo carga en la memoria
  - Ejecución
    - Cuando el proceso se empieza a ejecutar

    - Bloqueado
        - Una posible opción es que el proceso sea suspendido cuando hacemos un llamado para leer el archivo
        - Cuando espera que un proceso o recurso pueda ser utilizado
        - Cuando un proceso utiliza un recurso lo bloquea para que ningún otro proceso lo utilice
    - Salida/Terminado
        - Cuando el proceso es ejecutado y cumple con su objetivo
        - En este momento, da lugar a que otro proceso sea ejecutado

Sincronización de procesos

- Con base a los estados del proceso, existe una herramienta para la sincronización de procesos: LOS SEMÁFOROS
    - Mientras un proceso se está ejecutando y aparece una llamada de espera, el proceso entrante pasa a una lista de "bloqueado" y permanece ahí hasta que un proceso diferente le envía la señal de avance y este, que permanecía bloqueado, se coloca en una lista de espera para utilizar el CPU
- La sincronización de procesos permite que mientras un proceso está escribiendo un registro, otro proceso no pueda leer.

Mecanismos de comunicación entre procesos (IPC = Inter Process)
Consta de dos modelos de intercomunicación (IPC = Inter-Process Communication):

- Señales o pasos de mensaje
    - Avisos que puede enviar un proceso a otro, luego, el SO se encarga de que el proceso que recibe la señal tome una acción para gestionarla
        - La comunicación se hace mediante operaciones explícitas de envío y recepción.
    - Los procesos no comparten memoria
    - En los SO, esta tarea de intermediario la lleva el Kernel
    - No existen los errores como exclusión mutua
    - Son compatibles con cualquier tipo de arquitectura de computadora
- Memoria compartida
    - Recursos para que los software puedan intercambiar información
    - Dos procesos pueden estar realizándose con la memoria compartida y, al mismo tiempo, pueden estar intercambiando información
    - Se establece un espacio en memoria que será compartido por los procesos
    - Es generalmente más económica que usar un multiprocesador

Comunicación de procesos
Existen dos tipos de procesos:

- **Procesos independientes**
  - Tiene completa autonomía (no pueden ser afectados ni afectar a otros procesos que se estén ejecutando en el sistema)
  - No se comunican con otros procesos
- **Procesos cooperativos**
  - Se comunican mediante **memoria compartida y pasos de mensajes**
  - Trabaja en función de los recursos y la disponibilidad de otros procesos
    - Cuando se crea una lista de espera, se le llama **Área Crítica**
  - Sí puede afectar y ser afectados
    - Cualquier tipo de proceso que comparta cualquier tipo de datos o recursos con otros procesos es considerado cooperativo
  - Se debe hacer cooperación entre procesos porque:
    - Algunos procesos carecen de información, entonces, deben consultarla para poder ejecutarse
    - El CPU trabaja de una manera más eficiente y veloz, esto da como resultado la **modularidad**, es decir que cuando una tarea contiene varios pasos, el CPU puede ejecutarlos de manera independiente y simultanea
  - Este tipo de comunicación puede traer problemas, siempre y cuando, un proceso ejecute una tarea de forma errónea o no haya planificación
  - Se debe planifica el uso de la CPU y así evitar que la cola de procesos colapse o tenga una inanición, es decir, que funcione de forma tan deficiente ya que le negaría recursos a otros procesos que necesitan ejecutarse, debido a que los recursos necesarios para el procesamiento no están disponibles. El **planificador del procesador** tiene como misión la asignación del mismo a los procesos que están en la cola de procesos preparados
    - La **planificación** son políticas y mecanismos que poseen los SO para realizar la gestión del procesador
    - Los criterios que se deben tener en cuenta a la hora de elegir un algoritmo de planificación son:
      - Tiempo de procesador
        - Es el tiempo que un proceso está utilizando el procesador sin contar el tiempo que se encuentra bloqueado por operaciones de entrada/salida
      - Tiempo de respuesta

        - Es la velocidad con que el ordenador da la respuesta a una petición. Depende mucho de la velocidad de los dispositivos de entrada y salida
    - Tiempo de espera
        - Es el tiempo que los procesos están activos, pero sin ser ejecutados, es decir, los tiempos de espera en las distintas colas
    - Tiempo de servicio
        - Es el tiempo que tarda en ejecutarse un proceso, donde se incluye el tiempo de carga del programa en memoria, el tiempo de espera en la cola de procesos separados, el tiempo de ejecución en el procesador y el tiempo consumido en operaciones de entrada/salida
    - Tiempo de ejecución
        - Es idéntico al tiempo de servicio menos el tiempo de espera en la cola de procesos separados; es decir, es el tiempo teórico que necesitaría el proceso para ser ejecutado si fuera el único presente en el sistema.
    - Rendimiento
        - Es el número de trabajos o procesos realizados por unidad de tiempo, que debe ser lo mayor posible.
    - Eficiencia
        - Se refiere a la utilización del recurso más caro en un sistema, el procesador, que debe estar el mayor tiempo posible ocupado para lograr así un gran rendimiento.
- Para evitar esto existen diferentes técnicas de planificación (Algoritmos de la planificación de la CPU):
    - Método **FIFO** (First In, First Out)
        - Se asigna tiempo de ejecución al CPU al primer proceso que lo solicite
        - El procesador ejecuta cada proceso hasta que termina
        -
    - Método **SJF** (Shortest Job First)
        - La prioridad de ejecución está dada no por quien llega primero, sino por quien posee el menor tiempo de ejecución
        - Toma de la cola de procesos preparados el que necesite menos tiempo de ejecución para realizar su trabajo. Para ello, debe saber el tiempo de ejecución que necesita cada proceso, lo cual no es tarea fácil, pero es posible a través

de diversos métodos como puede ser la información suministrada por el propio usuario o por el propio programa, basándose en la historia anterior

- Método **SRTF** (Shortest Remaining Time)
  - Si un proceso largo se está ejecutando y llega un segundo de menor tiempo, se interrumpe el primero y se ejecuta el segundo. Una vez terminado este, comienza nuevamente a ejecutarse el primero en el sector donde fue cortado...a menos que aparezca otro más chico
    - Presenta mayor sobrecarga
    - Puede ser injusta ya que un proceso corto puede echar a uno largo que esté haciendo uso del procesador y que además esté terminando
    - Es muy eficiente
  - Cambia el proceso que está en ejecución cuando se ejecuta un proceso con una exigencia de tiempo de ejecución total menor que el que se está ejecutando en el procesador
- Método **Round Robin** (RR)
  - Existe una porción de tiempo establecida -o quantum de tiempo- en donde los procesos, a medida que van llegando a la sala de espera, se ejecutan en el CPU hasta que el quantum se cumple. Una vez cumplido, se interrumpe el proceso y si aún le falta tiempo de ejecución vuelve a la cola ubicándose al final hasta que es nuevamente su turno. Así se establece que todos los procesos lleven un tiempo de ejecución equitativo y todos los procesos serán ejecutados de la misma manera
  - Se concede a cada proceso de ejecución un determinado periodo de tiempo q (quantum), trascurrido el cual, si el proceso no ha terminado, se le devuelve al final de la cola de procesos preparados , concediéndole el procesador al siguiente proceso cor su correspondiente quatum
- Hay otras planificaciones que combinan de forma híbrida las planificaciones anteriores o presentan algoritmos diferentes, como ser la retroalimentación multinivel o planificación por comportamiento, estableciendo prioridades entre los procesos o cambiando el tiempo de ejecución.

        - Estos procesos vienen en constante innovación gracias al avance de la tecnología
        - Colas múltiples
            - Cuando los procesos que van a ser ejecutados en una computadora se puede agrupar en distintos grupos, podemos asignarlos a diferentes colas, cada una con distinta planificación, para darle a cada una de ellas la que realmente necesite
            - Esta política divide la cola en procesos preparados en varias colas separadas, de manera que los procesos se asignan a una determinada cola según sus necesidades y tipo

Procesos padres & Procesos hijos

- Cuando hay un proceso que no puede resolverse instantaneamente (como cuando ocurre una llamada al SO), se crean otros procesos que se denominan **IPC: Hijos**.
    - Su función es realizar subtareas para lograr que le proceso padre pueda cumplir su objetivo.
- Los procesos padres pueden tener varios hijos, pero los hijos solo pueden tener un proceso padre.

Hilos de ejecución

Un proceso puede dividirse en secuencias de tareas, denominadas hilos, los cuales son porciones de código que pueden ejecutarse de forma simultánea en cooperación con otros subprocesos

- Procesos simultáneos = Más eficiencia
    - Se tarda menos tiempo en crear un hilo nuevo en un proceso ya existente a crear un nuevo proceso
- **Múltiples hilos pueden existir dentro de un proceso**, ejecutándose de forma concurrente, compartiendo recursos y memoria
    - Una diferencia entre los hilos y los procesos en sí, es que estos últimos no comparten recursos entre ellos cuando se ejecutan
    - Al ser tantos hilos trabajando en conjunto es muy importante la sincronización, ya que un subproceso puede bloquear un recurso y negarlo el acceso a otro hilo
- Núcleos
    - Un solo núcleo (monolíticos):
        - Capacidad de respuesta menor
        - Su comportamiento es predecible
        - No presenta errores
        - Menores bloqueos de recursos

- Varios núcleos (multihilos):
  - Excelente capacidad de respuesta
  - Trabajo en paralelo
  - Sincronización compleja
  - Su comportamiento es impredecible
  - Puede presentar errores