**Procesos**

Siempre que queramos que nuestra computadora realice alguna instrucción, los procesos asumen el trabajo, para que la CPU ejecute la tarea por medio del plan diseñado por el Sistema Operativo.

**Procesos** = ejecución de un programa o instrucción.

Todos los softwares ejecutables se organizan en procesos que quiere usar la CPU y el sistema operativo organiza el orden en que se van ejecutando. A este *cambio de proceso* lo llamamos *cambio de contexto*.

Los procesos se ejecutan uno a la vez. No pueden almacenarse en la memoria principal ya que consumen espacio, y a la larga terminarían por llenar la memoria RAM. Por esto, son efímeros (se crean y se terminan).

* **Procesos de manera interactiva con el usuario**: cuando exporto un archivo, estoy generando un proceso.
* **Proceso de llamados al SO (en segundo plano):** cuando un software no puede acceder directo a un recurso le solicita al sistema operativo que lo gestione.

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

* **Nuevo:** cuando el proceso se crea.
* **Listo**: cuando el SO los carga a la memoria.
* **Ejecución**: cuando se empieza a ejecutar.
* **Bloqueado**: cambia de contexto si es suspendido cuando hago un llamado para leer el archivo.
* **Salida**: cuando el proceso es ejecutado y cumple con su objetivo. Dando lugar a que otro proceso sea ejecutado.

**Mecanismos de comunicación entre procesos IPC (Inter Process)**

* **Señales**: avisos que puede enviar un proceso a otro, luego el SO se encarga de que el proceso que reciba la señal tome una acción para gestionarla.
* **Memoria compartida**: un recurso compartido a disposición de los softwares para que puedan intercambiar información. Entonces dos procesos pueden estar realizándose con la memoria compartida y al mismo tiempo, pueden estar intercambiando información.
* **Hijos**: cuando se da un proceso que no puede resolverse instantáneamente como cuando ocurre una llamada al sistema. Función: realizar subtareas para lograr que el proceso padre logre su objetivo. Los padres pueden tener varios hijos, pero los hijos un solo padre.

**Comunicación de procesos**

Dos tipos de procesos se ejecutan de manera concurrente:

* **Procesos independientes**: son autónomos. 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 proceso que comparta cualquier tipo de datos o recursos con otro proceso es considerado cooperativo.
  + Algunos procesos carecen de información, por lo que deben consultarla para poder ejecutarse, por esto la información tiene que ser compartida.
  + Gracias a que la información es compartida el CPU trabaja mas eficiente y veloz, esto da como resultado la **modularidad**: cuando una tarea contiene varios pasos el CPU puede ejecutarlos de manera independiente y en simultaneo.
  + Siempre que un proceso ejecute una tarea de manera errónea, o no haya planificación, nos traerá problemas.
  + Un recurso se puede encontrar bloqueado por otro proceso que se está ejecutando.

Para que los recursos se comuniquen entre sí existen dos métodos de intercomunicación **IPC** (Inter-Process Communication):

* + - **La memoria compartida**: se establece un espacio en memoria compartida por los procesos.
      * Es más económica que usar un multiprocesador
    - **Pasos de mensajes**: por medio de un intermediario que en los SO es el kernel.
      * No existen los errores como exclusión mutua y son compatibles con cualquier tipo de arquitectura de computadora.

**Sincronización de procesos**

Los procesos pueden transitar por distintos estados que indican en qué parte de su ciclo de vida se encuentran y en base a esto, el SO toma decisiones sobre el:

* **Nuevo**: cuando un proceso se crea.
* **Listo**: cuando esta en condiciones de ser ejecutado.
* **Ejecución**: cuando su turno de usar el proceso comenzó.
* **Bloqueado:** cuando está esperando que un proceso o recurso pueda ser utilizado.
* **Terminado**: cuando ha sido ejecutado y su ciclo de vida finaliza.

En base a estos existe una herramienta para la sincronización de procesos: ***los semáforos.***

Mientras un proceso se está ejecutando, y aparece una llamada de espera, pasa a una lista de bloqueados, hasta que un proceso diferente le envía la señal de avance, y el proceso que permanecía bloqueado, se coloca en una fila de espera para usar el CPU.

**Hay dos tipos de procesos:**

* **Independientes**: trabajan por ellos mismos sin requerir ayuda a nadie.
* **Cooperativos**: trabajan en función de los recursos y la disponibilidad de otros procesos.

Buena planificación del uso del CPU para que la cola de procesos no colapse ni tenga una inanición (que funcione de forma tan deficiente porque le negaría recursos a otros procesos que necesitan ejecutarse debido a que los recursos necesarios para el procesamiento no están disponibles).

Para esto, existen diferentes técnicas de planificación…

**Algoritmos de planificación de CPU**

El planificador del procesador tiene como misión la asignación del mismo a los procesos que están en la cola de procesos preparados.

**Políticas de planificación:**

* **Primero en llegar, primero en salir FIFO:**

FIFO (First In, First Out). El procesador ejecuta cada proceso hasta que termina priorizándolos por orden de llegada, dejando así en una cola de espera a los que vayan llegando hasta sea su turno de ejecución.

* **Round-Robin (RR):**

Concede un periodo de tiempo determinado (quantum), a cada proceso de ejecución. Si este, no ha terminado, se lo devuelve al final de la cola hasta que sea nuevamente su turno.

* **El siguiente proceso, el más corto SJF (Shortest Job First):**

Prioriza a los procesos que requieran de un menor tiempo para completar su ejecución. Para esto, precisa saber el tiempo de ejecución de cada proceso.

* **Próximo proceso, el de tiempo restante más corto SRTF (Shortest Remaning Time First):**

Interrume el proceso que está en ejecución si llega un segundo proceso con menor tiempo de ejecución, para ejecutarlo y una vez terminado, retomar el primer proceso por donde fue cortado.

* **Colas múltiples**:

Cuando los procesos que van a ser ejecutados en una computadora se pueden 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.

**Hilos de ejecución**

**Procesadores** = conjunto de un gran numero de transistores configurados de tal forma que realizan operaciones binarias con impulsos de energía eléctrica. Estos a su vez contienen un ***núcleo***, que puede ser más de uno. A mayor cantidad de núcleos, mayor cantidad de procesos se podrán ejecutar en paralelo

**Procesos**: conjunto de operaciones que componen a un programa, que a la hora de ejecutarse se reparten la utilización del procesador para realizar su tarea.

Pueden dividirse en secuencias de tareas = **hilos** = porciones de código que pueden ejecutarse de forma simultánea en cooperación con otros subprocesos.

Múltiples hilos pueden existir dentro de un proceso ejecutándose de forma concurrente compartiendo recursos y memoria.

Ahí encontramos una gran diferencia con los procesos en sí, ya que estos últimos no comparten recursos cuando se ejecutan.

Al ser tantos hilos trabajando en conjunto, es muy importante la sincronización, ya que en su proceso puede bloquear un recurso y negarle el acceso a otro hilo.

**Sistemas de un solo núcleo:** (pueden trabajar con un solo hilo a la vez)

* Capacidad de respuesta menor.
* Su comportamiento es más predecible
* No presenta los errores que podría presentar los multihilos
* Problemas de bloque de recursos baja considerablemente

**Sistemas multi hilos:**

* Excelente capacidad de respuesta a operaciones
* buen trabajo en paralelo de sus tareas
* Sincronización compleja
* Puede presentar errores pasados por alto en la etapa de prueba y desarrollo

**planificación del CPU**

Políticas y mecanismos que poseen los SO actuales para realizar la gestión del procesador. Su objetivo es dar un buen servicio a todos los procesos que existan en un momento dado en el sistema.

**Criterios que se deben tener en cuenta a la hora de elegir o diseñar un algoritmo de planificación:**

* **Rendimiento**: es el número de trabajos o procesos realizados por unidad de tiempo, que debe ser lo mayor posible.
* **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 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.
* **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 espera**: es el tiempo que los procesos están activos, pero sin ser ejecutados, es decir, los tiempos de espera en las distintas colas.
* **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.