## Procesos

### 

Un proceso es la ejecución de un programa o instrucción. Es un conjunto de operaciones que componen un programa, los cuales a la hora de ejecutarse se reparten la utilización del procesador para realizar su tarea.

El cambio de proceso consiste en interrumpir la ejecución de un proceso y pasar a ejecutar otro proceso. El cambio de contexto consiste en el cambio de modo que se produce durante un cambio de proceso. Para cambiar el proceso de A por B, deja de ejecutar el proceso en curso (A), pasa de modo usuario a modo supervisor, luego el núcleo estudia si el proceso (B) está listo para ejecución..

Los procesos no se “almacenan” en la memoria principal ya que son efímeros.

La computadora ejecuta las tareas de acuerdo al plan de acción diseñado por el sistema operativo.

Se crean de manera interactiva por el usuario o en segundo plano (llamadas al SO).

Cuando un proceso no puede resolverse inmediatamente, se crean procesos hijos, los cuales realizan subtareas para que el proceso padre cumpla su objetivo. Un proceso padre puede tener varios procesos hijos, pero los procesos hijos tienen un único padre.

Existen 2 tipos de procesos que se ejecutan de manera concurrente:

* **Procesos independientes**: tienen total autonomía, no pueden afectar ni ser afectados por otros procesos en ejecución. Tienen todos los recursos que necesitan para poder terminar. No se comunican con otros procesos.
* **Procesos cooperativos**: pueden afectar y ser afectados por otros procesos. Trabajan en función de la disponibilidad y los recursos de otros procesos. Existen 2 motivos para compartir recursos:
  + Información compartida: algunos procesos carecen de información suficiente, por lo que deben consultar la compartida para poder ejecutarse.
  + CPU eficiente y veloz: gracias a la información compartida, el CPU trabaja de manera más eficiente y veloz. Esto da lugar a la **modularidad**: ejecución independiente y simultánea de varios pasos de una tarea.

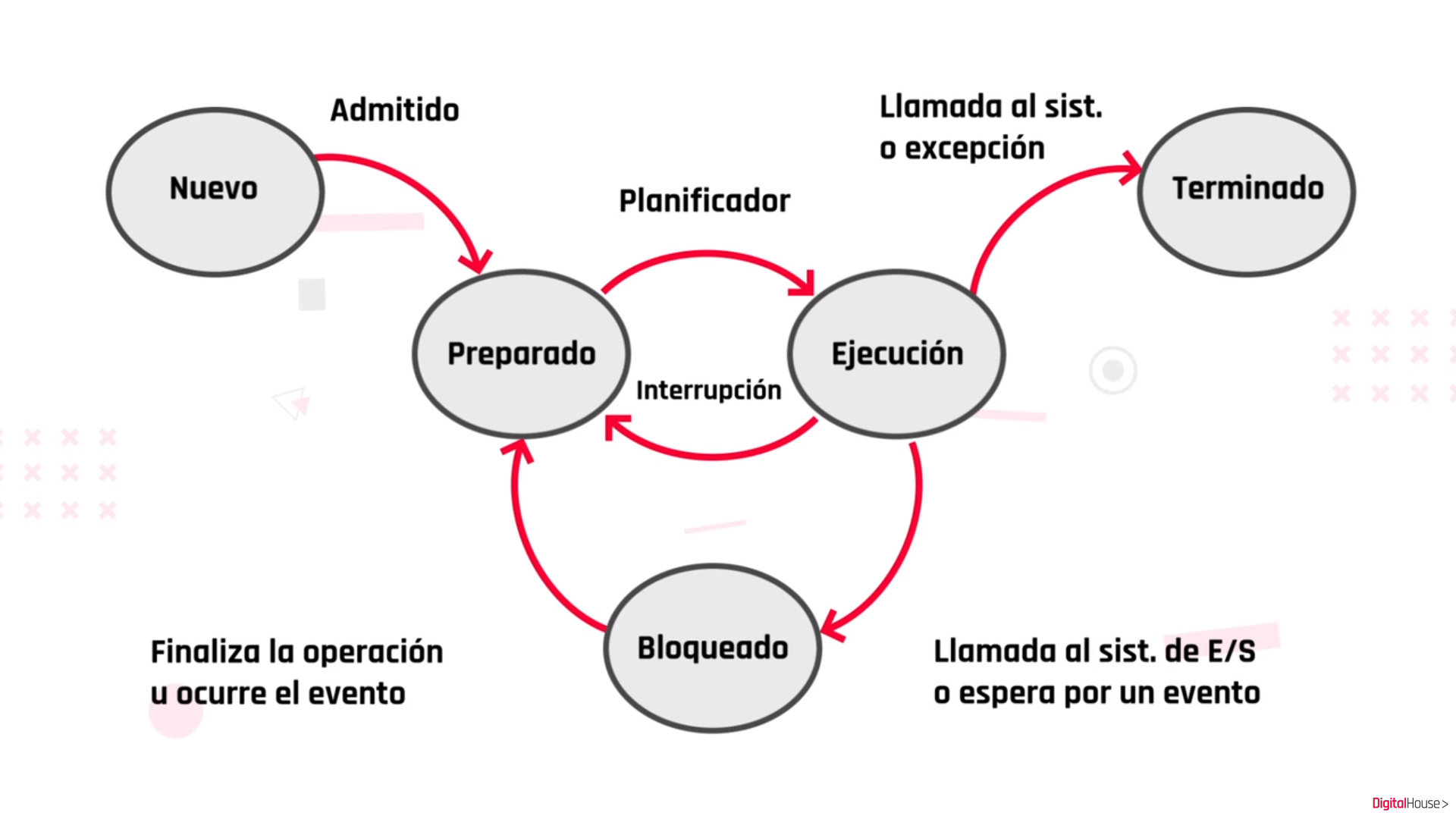
Las tareas de un proceso pueden depender de la finalización de otras tareas. Esto puede ocasionar errores cuando un recurso está bloqueado por la ejecución de otra tarea.

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

* **Memoria compartida**: se establece una porción de memoria que será compartida por diversos procesos para que puedan intercambiar información. Es el kernel el que establece los permisos.
  + Ventaja: es más económica que usar un multiprocesador.
* **Paso de mensajes/señales**: La comunicación entre los procesos se hace mediante operaciones explícitas de envío y recepción, siendo el kernel el intermediario.
  + Ventajas: no existen errores como la exclusión mutua; son compatibles con cualquier arquitectura de computadora.

**Ciclo de vida/Estados de los procesos**:

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



Cuando un proceso está utilizando un recurso lo bloquea de manera que los demás no lo puedan usar. Cuando un proceso se está ejecutando y aparece una llamada de espera, este pasa a una lista de bloqueados y permanece ahí hasta que un proceso diferente le envía la señal de avance. Así, el proceso que estaba bloqueado pasa a una lista de espera para usar la CPU.

**Área o sección crítica**: acceso a un recurso compartido que no debe ser accedido por más de un proceso o hilo en ejecución. 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, monitores, el algoritmo de Dekker y Peterson, los candados.

Se requiere de una buena planificación de procesos para que la cola de procesos no colapse ni se produzca una **inanición** (a un proceso o hilo se le deniega siempre el acceso a un recurso compartido, por lo que la tarea no puede finalizar nunca).

La **planificación** de procesos consiste en las políticas y mecanismos que poseen los sistemas operativos 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.

**Técnicas o políticas de planificación de procesos:**

* **FIFO** (First In, First Out): se asigna la CPU al proceso que la solicita primero.
* **SJF** (Shortest Job First): se asigna la CPU al proceso con el menor tiempo de ejecución hasta que dicho proceso finaliza.
  + Puede provocar inanición.
* **SRTF** (Shortest Remaining Time First): si un proceso se está ejecutando y llega uno más corto que lo que queda del actual, se interrumpe y se ejecuta el nuevo proceso. Una vez que se termina, se retoma la ejecución del proceso anteriormente interrumpido.
  + Es muy eficiente
  + 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.
* **Round Robin**: existe una porción o quantum de tiempo establecido. Se ejecutan los procesos a medida que llegan a la fila de espera hasta que se termina su ejecución o el quantum se cumple (lo que suceda primero). Si el proceso no finalizó, vuelve a la cola de espera.
* Retroalimentación multinivel
* Planificación por comportamiento
* **Colas múltiples**: los procesos se asignan a diferentes colas, cada una con una planificación diferente, de acuerdo a sus necesidades.

**Criterios a tener en cuenta en un algoritmo de planificación:**

* **Eficiencia**: se refiere al porcentaje de tiempo en que el procesador está siendo utilizado, que debe ser el mayor posible.
* **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 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
* **Rendimiento**: es el número de trabajos o procesos realizados por unidad de tiempo, que debe ser lo mayor posible.
* **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 espera**: es el tiempo que los procesos están activos, pero sin ser ejecutados, es decir, los tiempos de espera en las distintas colas.

**Hilos de ejecución**

Los procesos pueden dividirse en secuencias de tareas o hilos, los cuales son porciones de código que pueden ejecutarse de forma simultánea en cooperación con otros subprocesos. Son divisiones lógicas, no físicas. De esta manera, múltiples hilos pueden existir dentro de un proceso, ejecutándose de forma concurrente, compartiendo recursos y memoria, y aumentando la eficiencia.

Se tarda menos tiempo en crear un hilo nuevo en un proceso ya existente a crear un nuevo proceso.

**Los procesos no comparten recursos cuando se ejecutan, los hilos sí**. Los hilos pueden bloquear recursos y negarle el acceso a otros hilos, por lo que se requiere de una buena sincronización.

Procesadores mononúcleo:

* Capacidad de respuesta menor
* Comportamiento predecible
* No presentan errores que podrían presentarse en los multihilos
* Los problemas de bloqueo de recursos se reducen considerablemente

Procesadores multinúcleo:

* Excelente capacidad de respuesta
* Trabajo en paralelo
* Sincronización compleja
* Comportamiento difícil de predecir

1 - ¿Un script puede estar compuesto por un único comando? VERDADERO

2 - Git es una tecnología desarrollada para rastrear y registrar cambios en cualquier tipo de archivo de computadora, especialmente cuando se trata de archivos utilizados por más de una o dos personas. VERDADERO

3 - Un núcleo no puede trabajar con más de un proceso a la vez. VERDADERO (puede ejecutar diferentes subprocesos de un mismo proceso)

4 - Opción 1: El kernel es una aplicación que corre sobre el sistema operativo.

Opción 2: Es una parte del sistema operativo que asigna o quita recursos de hardware. (CORRECTA)

Opción 3: Es un método...

5 - Pueden coexistir múltiples hilos de un proceso.

Opción 1: Sí, comparten recursos y memoria. (CORRECTA)

Opción 2: Sí, comparten...

Opción 3: No, ...

6 - ¿Quién interactúa en la comunicación por memoria compartida? EL KERNEL, memoria primaria, memoria secundaria