

Clase 8 - Procesos



Status

Partes de una computadora

Siempre que queramos que la computadora realice una instrucción, se va a ejecutar un proceso en la CPU. El proceso se va a ejecutar según el plan de acción que gestiona el SO.

Un proceso se trata básicamente de un programa que entra en ejecución. Son una sucesión de instrucciones que buscan realizar una tarea.

Un programa es un algoritmo que genera una secuencia de instrucciones con las que podemos realizar una tarea concreta.

El SO es quien se encarga de organizar el orden en que se ejecutarán los procesos de los programas o software.

EJEMPLO: analogía de tostar el pan por la mañana:

- **El sistema operativo** será nuestra casa o la habitación en donde estemos, la cual proporciona los medios para poder ejecutar la tarea.
- **El programa** será la tarea que queremos realizar, la de tostar el panel
- **Los procesos** serán las distintas tareas que tendremos que hacer hasta llegar a la solución: abrir el pan → enchufar el tostador → colocar el pan → esperar a que se tueste → sacarlo → desconectar el tostador.
- **Nosotros somos el procesador**, los encargados de ejecutar la secuencia de tareas o procesos.

Dentro del proceso tenemos lo que llamamos instrucciones, que corresponden a cada uno de los pasos que debemos hacer para completar esa tarea.

Además, para separar cada proceso, el procesador le asigna un contador de programa, para que cada uno esté separado y bien diferenciado de otro que incluso puede ser igual, por ejemplo, abrir dos veces el explorador. De esta forma cada proceso se guarda en distintos registros, con distintas variables y por supuesto en distinta región de la memoria RAM.



Los procesos no pueden almacenarse en la memoria principal, ya que consumen espacio. Son efímeros, se crean y se terminan.

Creación de un proceso

Las **formas de empezar un proceso informático** serán las siguientes:

- **Que nosotros arranquemos un programa o el propio ordenador:**
 - Petición del usuario para crear un proceso. Al darle click a un icono o boton.
- **Que el sistema llame a los programas o procesos:**
 - Arranque del sistema
 - Una llamada al SO para la creación de un proceso.

Terminación de un proceso

- **Terminar la rutina o el programa:** dando un resultado final que considera correcto.
- **Finalizar de forma repentina por un error:** la rutina puede estar mal programada y no dar el resultado esperado.
- **A partir de otro proceso o interruptor por nosotros mismos:** nosotros mismos podemos ejecutar una tarea que elimine la que se está ejecutando
- **Se puede bloquear:** si espera una terminada respuesta y ésta no llega el proceso permanecerá bloqueado hasta que el sistema detecte que no puede continuar.
- **Por un corte de alimentación**

Estados de un proceso

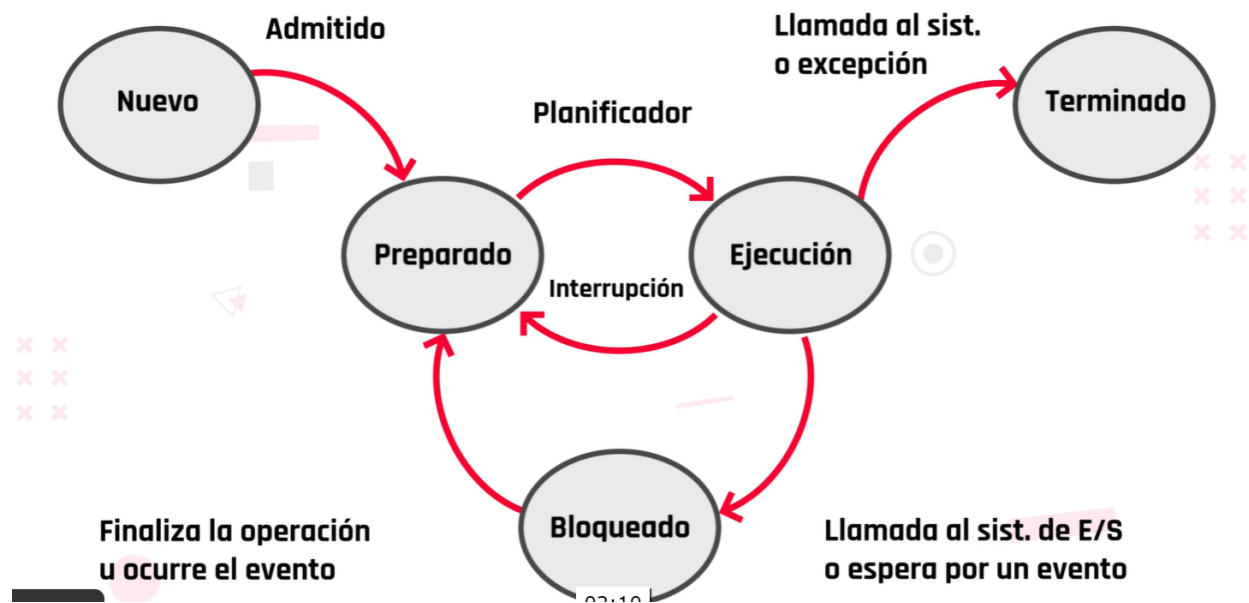
Los procesadores sólo pueden ejecutar un solo proceso a la vez, turnándolos para el uso de este.

Los procesos ocupan el procesador hasta que una interrupción o señal llega al procesador para hacer el cambio de proceso, a esto se le conoce como **cambio**

de contexto.

Los posibles estados son:

- **Nuevo:** el proceso se está creando
- **Listo:** una vez que el proceso es admitido por el SO para estar listo. Está esperando a que se le de tiempo del CPU. Se cargó el proceso en la memoria de los registros.
- **Ejecución:** Al liberarse un procesador, el planificador (scheduler) selecciona el próximo proceso a ejecutar. Cuando pasa a ejecución, las instrucciones del proceso se están ejecutando en el CPU. Cuando hay alguna interrupción, el procesador queda libre para ejecutar otra tarea y el proceso original vuelve a la cola de listo, hasta que termine el time-out. Luego vuelve a ejecución.
- A medida que el proceso ejecuta instrucciones realiza pedidos en distintos componentes (ej.: genera un pedido de E/S). Teniendo en cuenta que el pedido puede demorar, el proceso es puesto en una cola de espera hasta que se complete su pedido. De esta forma, se logra utilizar en forma más eficiente el procesador.
- **Bloqueado:** eventualmente el proceso hace una solicitud o llamada a entrada y salida. Allí queda bloqueado esperando un evento. Este evento es que se complete el pedido de entrada y salida o una señal. El CPU esta libre para ejecutar otros procesos. Una vez que ocurre el evento que el proceso estaba esperando en la cola de espera, el proceso es puesto nuevamente en la cola de procesos listos y luego al estado de ejecución.
- **Salida:** Cuando el proceso ejecuta sus última instrucción pasa al estado terminado. El proceso terminó su ejecución



Tipos de procesos

- **INDEPENDIENTES:** tienen total autonomía, no pueden ser afectados ni afectar otros procesos que se estén ejecutando en el sistema
- **COOPERATIVOS:** pueden afectar y ser afectadas. Comparten datos o recursos.

Porque debe haber comunicación:

1. Información compartida: algunos procesos requieren consultar información que no tienen para poder ejecutarse.
2. Hace que el cpu sea mas eficiente y veloz.
3. Modularidad: cuando una tarea contiene varios pasos el cpu puede ejecutarlos de forma independiente y simultanea.

Desventajas: cuando no hay planificación y un proceso ejecuta una tarea de forma errónea.

Comunicación entre procesos (IPC)

Los procesos pueden interactuar cooperando en pos de un objetivo común. Por esto es necesario que el sistema operativo brinde unas herramientas específicas para la comunicación y sincronización entre los procesos (Inter Process Communication – IPC).

IPC es una herramienta que permite a los procesos comunicarse y sincronizarse sin compartir el espacio de direccionamiento en memoria.

1. **Paso de mensajes:** son avisos que puede enviar un programa a otro. El programa que recibe la señal toma acción para gestionarla.

Hay un intermediario entre ambos procesos que permite compartir la información, este intermediario es el kernel.

La comunicación se hace mediante operaciones explícitas de envío y recepción.

No existen errores de exclusión mutua.

Compatibles con cualquier arquitectura de computadora.

2. **Memoria compartida:** un recurso compartido para que los software puedan intercambiar información. Dos programas mientras se ejecutan comparten información.

Mas económica.

Cuando hay procesos que no se pueden realizar, por ejemplo porque necesitan una llamada al sistema, se crean procesos hijos. Estos realizan subtareas del proceso padre.

Subprocesos o hilos

Los sistemas actuales permiten ejecutar varios programas de forma simultánea, y en consecuencia, tendremos gran cantidad de procesos activos en el sistema, decimos que son multihilo. Cada proceso entonces se divide en uno o varios hilos de ejecución o subprocesos. Cada hilo, tiene sus propias instrucciones y un estado de ejecución.

Comparten recursos y memoria.

Siguiendo con el **símil de tostar el pan**, podríamos entender esto de la siguiente forma:

- Fijándonos en el proceso de esperar que se tueste el pan, **podemos dividirlos en varios subprocesos o hilos**, por ejemplo, partir el pan dos trozos y aprovechar las dos ranuras del tostador. O también comernos un trozo mientras tostamos el otro.
- Cada trozo estará más o menos tostado, **y ese será su estado de ejecución**, nosotros, el procesador, debemos estar pendientes de que no se queme.

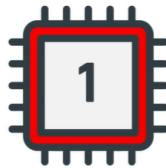
Desventajas: es muy importante la sincronización ya que un hilo puede ocupar un recurso y bloquearlo para el resto de los hilos.

Procesadores

Monolítico: solo pueden trabajar con 1 hilo a la vez.

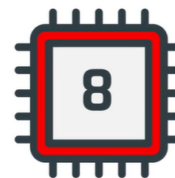
Multihilos: varios hilos de ejecución y alta velocidad de procesamiento. Mas impredecibles.

Monolíticos = un solo núcleo



- ☒ Capacidad de respuesta menor.
- ☒ Comportamiento predecible.
- ☒ No presenta errores.
- ☒ Menores bloqueos de recursos.

Multinucleos = varios núcleos



- ☒ Excelente capacidad de respuesta.
- ☒ Trabajo en paralelo.
- ☒ Sincronización compleja.
- ☒ Puede presentar errores.

Planificación

Son 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.

- La planificación (scheduling) es la base para lograr la multiprogramación.
- Un sistema multiprogramado tendrá varios procesos que requerirán el recurso procesador a la vez.
- Esto sucede cuando los procesos están en estado listo.
Si existe un procesador disponible y existen procesos en estado listo, se debe elegir el que será asignado al recurso para ejecutar.
- El componente del sistema operativo que realiza la elección del proceso es llamada planificador (scheduler)

A la hora de elegir o diseñar un algoritmo de planificación se deben tener en cuenta las siguientes características:

1. Rendimiento
2. Tiempo de respuesta
3. Tiempo de servicio
4. Tiempo de ejecución
5. Tiempo de procesador
6. Eficiencia
7. Tiempo de espera

Algoritmos de planificación de CPU

El planificador del procesador asigna tiempo de CPU a los procesos que están en la cola de procesos listos. Se decide cuál usar al desarrollar el sistema operativo. Puede usar varios para diferentes procesos.

Es el que menos rendimiento ofrece.

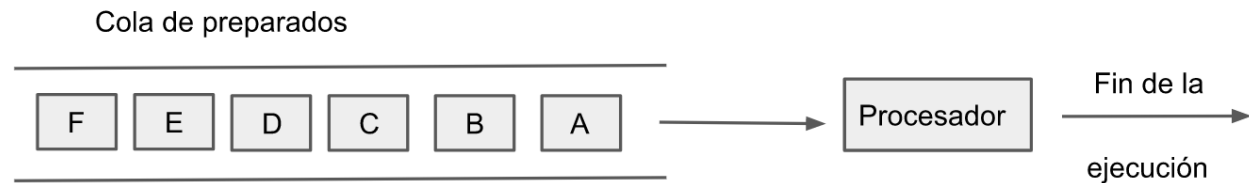
Sencillo y simple

PRIMERO EN SALIR - FIFO (first in, first out)

Los procesos son ejecutados en el orden que llegan a la cola de procesos listos.

Es un algoritmo no expropiativo: una vez que el procesador le es asignado a un proceso este lo mantiene hasta que termina o se bloquea (por ejemplo al generar un pedido de E/S).

El tiempo de espera promedio por lo general es alto.



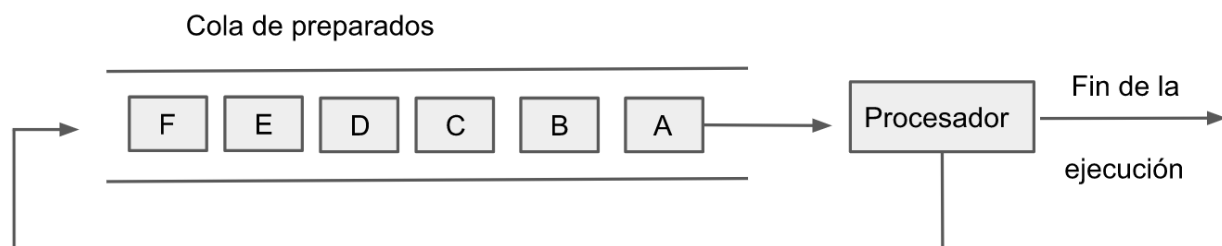
Procesos	Llegada	Tiempo uso CPU
P1	0	11
P2	2	3
P3	3	3
P4	4	3

ROUND-ROBIN - RR

A cada proceso se le brinda un intervalo de tiempo para el uso del procesador (time quantum).

Al finalizar el tiempo, el procesador le es expropiado y vuelve al estado listo al final de la cola.

Es fácil de implementar ya que solamente es necesario una cola de procesos listos. Cuando un proceso consume su quantum es puesto al final de la cola.



SHORTEST JOB FIRST - SJF (el siguiente mas corto)

Este algoritmo toma de la cola de procesos listos aquel que necesite menos tiempo de ejecución.

Cuando el procesador queda disponible se le asigna al proceso que tenga el menor tiempo de ejecución.

Si dos procesos tiene el mismo tiempo de ejecución se desempata de alguna forma.

Su funcionamiento depende de conocer los tiempos de ejecución lo cual en la mayoría de los casos no sucede.

SHORTEST REMAINING TIME FIRST - SRTF

El proceso en CPU es desalojado si llega a la cola un proceso con duración más corta.

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.

Es muy eficiente

MULTILEVEL QUEUE (Colas múltiples)

Si los procesos se pueden clasificar según sus cualidades, es posible dividir la lista de procesos listos (ready queue) en varias colas (una para cada clasificación).

Los procesos son asignados permanentemente a una de las colas.

Cada cola tendrá su propio algoritmo de planificación propio.

Además, se debe tener una estrategia de planificación entre las diferentes colas.

Por ejemplo, una cola tendrá prioridad sobre otra.

Los procesos no comparten recursos, es decir si un proceso esta usando la impresora otro no la va a poder usar hasta que este se libera.