**Universidad Nacional Autónoma de México**

**Facultad de Ingeniería** portada

División de Ciencias de la Ingeniería

Semestre 2025 – 2

**Sistemas Operativos**

*Tarea 2*

Profesor: Cruz Sergio Aguilar Diaz

Grupo 02

Muñoz San Agustin Victoria Monserrat

Fecha de entrega: 05 de marzo de 2024

Contenido

[**Investigar conceptos de memoria** 2](#_Toc192061225)

[*Definición de memoria.* 2](#_Toc192061226)

[*Clasificación de memoria* 3](#_Toc192061227)

[*Tipos de memoria* 4](#_Toc192061228)

[*Evolución de memoria RAM - Año, nombre e Imagen.* 7](#_Toc192061229)

[*Evolución de memoria secundaria (Almacenamiento)– Año, nombre e imagen.* 9](#_Toc192061230)

[**¿Qué entiendes por memoria de intercambio? ¿Cuál es la función que realiza en los Sistemas Operativos?** 12](#_Toc192061231)

[**Investigar:** 15](#_Toc192061232)

[*¿Qué es un proceso?* 15](#_Toc192061233)

[*¿Qué es un hilo de ejecución?* 16](#_Toc192061234)

[*La diferencia entre hilos y procesos.* 17](#_Toc192061235)

[*Características de los procesos.* 18](#_Toc192061236)

[**Conclusiones** 21](#_Toc192061237)

[**Bibliografía** 22](#_Toc192061238)

# **Investigar conceptos de memoria**

## *Definición de memoria.*

La memoria es el dispositivo que retiene, memoriza o almacena datos informáticos durante algún periodo de tiempo.1​ La memoria proporciona una de las principales funciones de la computación moderna: el almacenamiento de información y conocimiento. Es uno de los componentes fundamentales de la computadora, que interconectada a la unidad central de procesamiento (CPU, por las siglas en inglés de Central Processing Unit) y los dispositivos de entrada/salida, implementan lo fundamental del modelo de computadora de la arquitectura de Von Neumann.

En la actualidad, «memoria» suele referirse a una forma de almacenamiento de estados sólido, conocida como memoria RAM (memoria de acceso aleatorio; RAM por sus siglas en inglés, de random access memory), y otras veces se refiere a otras formas de almacenamiento rápido, pero temporal. De forma similar, se refiere a formas de almacenamiento masivo, como discos ópticos, y tipos de almacenamiento magnético, como discos duros y otros tipos de almacenamiento, más lentos que las memorias RAM, pero de naturaleza más permanente. Estas distinciones contemporáneas son de ayuda, porque son fundamentales para la arquitectura de computadores en general.

Además, se refleja una diferencia técnica importante y significativa entre «memoria» y «dispositivos de almacenamiento masivo», que se ha ido diluyendo por el uso histórico de los términos «almacenamiento primario» (a veces «almacenamiento principal»), para memorias de acceso aleatorio, y «almacenamiento secundario», para dispositivos de almacenamiento masivo. Esto se explica en las siguientes secciones, en las que el término tradicional «almacenamiento» se usa como subtítulo, por conveniencia.

## Memoria RAM vs memoria ROM: diferencias principales*Clasificación de memoria*

La clasificación de la memoria en computadoras se organiza según diferentes criterios:

1. *Según la permanencia de los datos*

* **Memoria Volátil**

Pierde los datos cuando se apaga la computadora.

* **Memoria No Volátil**

Conserva los datos incluso sin energía.

1. *Según su función en el sistema*

* **Memoria Principal (Primaria)**

Accesible directamente por la CPU y esencial para la ejecución de programas.

* **Memoria Secundaria**

Almacenamiento permanente de datos y programas.

* **Memoria Caché**

Memoria de alta velocidad que almacena datos usados frecuentemente por el procesador.

* **Memoria Virtual**

Espacio en el disco duro usado como una extensión de la RAM cuando esta se llena.

1. *Según el método de acceso*

* **Memoria de Acceso Aleatorio (RAM)**

Permite acceder a cualquier dato sin un orden específico.

* **Memoria de Acceso Secuencial**

Se accede a los datos en un orden específico, como en cintas magnéticas.

1. *Según su ubicación*

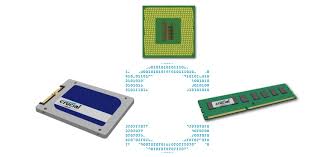
* **Memoria Interna**

Está dentro del procesador o en la placa base.

* **Memoria Externa**

Dispositivos de almacenamiento fuera de la CPU.

## *Tipos de memoria*

Comprender los diversos tipos de memoria de computadora es esencial para comprender la funcionalidad y el rendimiento de los sistemas informáticos. *Según la permanencia de los datos*

* **Memoria Volátil**

Ejemplo: registros.

* + *Memoria de acceso aleatorio (RAM)*

La RAM es un tipo de memoria volátil, lo que significa que pierde sus datos cuando se apaga el sistema. Se utiliza para el almacenamiento temporal de datos, lo que permite que la unidad de procesamiento central (CPU) acceda y procese datos rápidamente.

* + *Memoria caché*

La memoria caché es una memoria volátil de alta velocidad que proporciona acceso de datos de alta velocidad a la CPU y mejora la velocidad y el rendimiento de la computadora.

* **Memoria No Volátil**

Ejemplo: EEPROM, HDD, SSD, USB, Flash, SCM.

* + *Memoria de solo lectura (ROM)*

A diferencia de la RAM, el ROM no es volátil y conserva sus datos incluso cuando está apagado. Almacena la información crítica necesaria para arrancar la computadora.

* + *Memoria flash*

La memoria flash es un tipo de memoria no volátil que se usa para almacenar y transferir datos entre dispositivos digitales y sistemas informáticos.

* + *Memoria de clase de almacenamiento (SCM)*

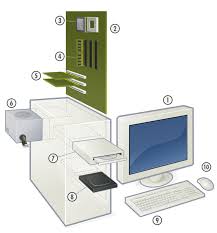
SCM es un tipo de memoria no volátil que proporciona alta velocidad, baja latencia y almacenamiento de datos persistente, lo que reduce la brecha entre RAM y almacenamiento tradicional.

* *Según su función en el sistema*
* **Memoria Principal (Primaria)**

Ejemplo: RAM, ROM.

* **Memoria Secundaria**

Ejemplo: HDD, SSD, USB, CD/DVD.

* **Memoria Caché**

Ejemplo: Caché L1, L2, L3.

* **Memoria Virtual**
* *Según el método de acceso*
* **Memoria de Acceso Aleatorio (RAM)**
  + *Memoria de acceso aleatorio a video (VRAM)*

VRAM es una memoria gráfica dedicada que se utiliza para almacenar datos de imágenes y videos, lo que facilita una renderización más rápida y un mejor rendimiento de los gráficos.

* + *Memoria de acceso aleatorio estático (SRAM)*

SRAM es un tipo de memoria volátil que conserva los datos siempre que se suministre energía, lo que ofrece velocidades de acceso más rápidas en comparación con DRAM, aunque a un costo más alto.

* + *Memoria dinámica de acceso aleatorio (DRAM)*

La DRAM es un tipo de memoria volátil que almacena cada bit de datos en un capacitor separado dentro de una celda de memoria, lo que requiere una actualización periódica para mantener los datos. Es más lento que la SRAM, pero se usa ampliamente debido a su rentabilidad y mayor densidad, lo que la convierte en una opción adecuada para la memoria del sistema principal donde se requieren grandes capacidades.

* **Memoria de Acceso Secuencial**
* *Según su ubicación*
* **Memoria Interna**

Ejemplo: RAM, ROM, caché, registros.

* **Memoria Externa**

Ejemplo: HDD, SSD, USB, CD/DVD.

Cada uno de estos tipos de memoria desempeña un papel fundamental en el funcionamiento y la eficiencia de los sistemas informáticos, lo que contribuye al rendimiento y la funcionalidad general.

## *Evolución de memoria RAM - Año, nombre e Imagen.*

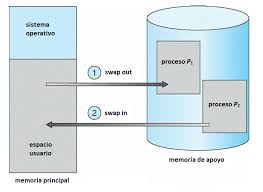
|  |  |  |
| --- | --- | --- |
| Año | Nombre | Imagen |
| 1968 | Memoria de semiconductores | MEMORIA SEMICONDUCTORA |
| 1969 | SRAM o memoria RAM estática | historia tamaño memorias RAM Intel |
| 1970 | DRAM (Dynamic RAM) - Intel 1103 | Intel 1103 primer DRAM |
| 1983 | SIMM | módulo SIMM |
| 1992 | SDRAM y SDR | SDR SDRAM primer módulo |
| 1996 | DDR | Samsung DDR primer modulo |
| 1999 | RDRAM (Rambus DRAM) | memoria RAMBUS RDRAM |
| 2000 | eDRAM | eDRAM integrada tamaño memorias |
| 2001 | DDR2 SDRAM | Samsung DDR2 |
| 2007 | DDR3 SDRAM | DDR3 Samsung tamaño memorias |
| 2014 | DDR4 SDRAM | DDR4 modulos historia |
| 2021 | DDR5 SDRAM | MEMORIA RAM - SODIMM DDR5 - ADATA PREMIER - 8GB - 5600MHZ (PORTATIL) |  Datacont S.A.C |

## *Evolución de memoria secundaria (Almacenamiento)– Año, nombre e imagen.*

|  |  |  |
| --- | --- | --- |
| Año | Nombre | Imagen |
| 1890 | Tarjetas perforadas (utilizadas en máquinas tabuladoras) | ccapitalia.net » Datos, censo y tabuladoras |
| 1928 | Tambor magnético | Tambor magnético - Wikipedia, la enciclopedia libre |
| 1956 | Disco duro (HDD) - IBM 305 RAMAC (5 MB, 50 discos) | 6 curiosidades del primer disco duro que no sabías |
| 1963 | Cinta de casete de audio (usada en almacenamiento de datos) | Cinta de casete estándar vacía 60 minutos de grabación de audio para  reproductor de música de voz : Amazon.com.mx: Electrónicos |
| 1971 | Disquete de 8 pulgadas (IBM) | Disco flexible - UMA Divulga |
| 1976 | Disquete de 5.25 pulgadas | Disquete de 5,25 pulgadas , Representaciones Incluyendo: información y  retro - Envato |
| 1980 | HDD de 3.5 pulgadas (Seagate ST-506) | ST-506 - Wikipedia, la enciclopedia libre |
| 1981 | Disquete de 3.5 pulgadas (Sony) | Disquete 3.5 pulgadas: historia del medio extraíble del siglo XX |
| 1984 | CD-ROM (Compact Disc Read-Only Memory) | Compact Disc Read Only Memory or CD-ROM |
| 1994 | ZIP Drive (Iomega, capacidad de 100 MB y más) | Iomega 100MB Zip Plus Unidad de disco - Dual SCSI y puertos paralelos :  Amazon.com.mx: Electrónicos |
| 1995 | DVD (Digital Versatile Disc) | DVD - Wikipedia, la enciclopedia libre |
| 1999 | Memoria Flash USB (Pendrive) | Memoria USB 16GB Pendrive USB 2.0 10 Piezas - Práctico Pen Drive 16 GB  Económico 10 Unidades Portátil Flash Drives - Giratorio Llave USB 2.0 ... |
| 2003 | HDD SATA (Serial ATA) | Qué es SATA? ¿Cómo funciona SATA? |
| 2006 | Blu-ray Disc (BD) |  |
| 2007 | SSD (Solid State Drive) | Sandisk SSD 128GB : Amazon.com.mx: Electrónicos |
| 2013 | SSD NVMe (Non-Volatile Memory Express) | NVMe (Non-Volatile Memory Express) Definition |

# **¿Qué entiendes por memoria de intercambio? ¿Cuál es la función que realiza en los Sistemas Operativos?**

* **¿Qué es el Swapping?**

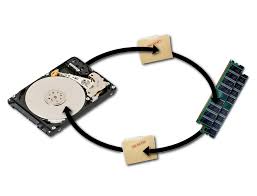
Se entiende como Swapping o espacio swap a una participación del disco o a un archivo. Estos pueden ser creados por los usuarios al momento de la instalación del sistema operativo o cuando lo desee. Estos espacios de intercambio son muy recomendados para personas cuyos sistemas poseen 1 o menos GB de RAM. Pero, también puede aplicarse en equipos con mayor potencia.

Se recomienda usar la siguiente cantidad de memoria swap:

* 1 GB de RAM o menor: misma cantidad de swap.
* De 2 a 4 GB de memoria RAM: la mitad para swap.
* Superior a 4 GB de RAM: 2 GB de memoria swap.

En el sistema operativo Linux, se considera que la partición swap es necesaria. Los desarrolladores de la distribución Ubuntu eliminaron esta necesidad. Reemplazando la misma, desde la distribución 17.04 con el archivo Swapfile.

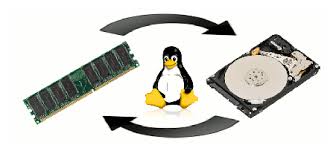
* **Utilidades del Swapping en el sistema operativo**

En un sistema operativo, son diversas las finalidades para las que se lleva a cabo el Swapping. Entre todos, podemos destacar los más importantes:

* + Puede ser utilizado como memoria adyacente. Minimizando las operaciones de entrada y salida en un archivo, para leerlo o escribirlo.
  + Sirve para guardar en un fichero swap todas las aplicaciones que tienen un poco uso o no son tan demandadas. Ayuda al sistema a mantener continuamente una cierta cantidad de memoria RAM libre.
  + La cantidad de memoria RAM que se ha podido liberar por la partición swap, puede utilizarse por el sistema operativo para otras funciones.
* **¿Cómo utiliza Windows el Swapping?**

En el caso del sistema operativo Windows, cuando se utiliza toda la memoria RAM disponible, este activa el Swapping. Windows se encarga de administrar de manera automática, el tamaño del archivo. Permitiendo al sistema operativo tener, de forma temporal, más recursos. En la versión de Windows 10, el tamaño de la memoria swap puede modificarse desde el panel de control. Accediendo a la configuración avanzada del sistema.

* **¿Cómo crear un Swapping desde Linux?**

Se puede comprobar el estado del Swapping desde Linux, utilizando alguna de estas dos órdenes Swapon-s o Free-h.

Se puede crear una partición swap, haciendo uso de cualquier herramienta de gestión de partición en GNU/Linux. La memoria swap es establecida como tipo 82. Esto no impide que pueda ser utilizada cualquier otra partición para el Swapping.

Si deseas crear un área de Swapping en Linux, utiliza la orden mkswap.

La orden mkswap crea por defecto una partición UUID. Si se desea especificar una UUID personalizada, se deberá utilizar la etiqueta –U de la siguiente forma:

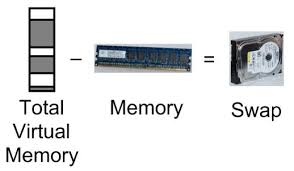
*sudo mkswap -U custom\_UUID /dev/sda2*

Cuando se busca activar el dispositivo para la paginación, se utiliza:

*sudo swapon /dev/sda2*

Si se desea activar la participación swap al arrancar el equipo, se debe ejecutar el siguiente comando:

*sudo echo «/dev/sda2 none swap defaults 0 0» >> /etc/fstab*

* **Activación por systemd**

Systemd activa el Swapping haciendo uso de dos diferentes mecanismos. Estos se ejecutan en /usr/lib/systemd/system-generators. Activándose cuando el sistema operativo arranca, creando unidades nativas de systemd para los montajes. El mecanismo ic|systemd-fstab-generator se encarga de leer el archivo fstab para crear unidades como la swap. El mecanismo systemd-gpt-auto-generator evalúa el disco para crear nuevas unidades.

* **Swapping desde un dispositivo USB**

Linux ofrece excelentes modularidades, donde se pueden tener diversas particiones en dispositivos diferentes. Cuando el disco duro se encuentra saturado, hacer uso de un dispositivo USB puede servir para crear una partición temporal. Sin embargo, existen ciertas desventajas que acompañan a este truco:

El dispositivo USB funcionará de forma más lenta que el disco duro.

Los ciclos de lectura se encuentran limitados en las memorias flash. Haciendo que su uso para Swapping acorte su vida útil.

Si otro dispositivo se encuentra conectado al equipo, no puede ser usado para swapping.

* **Swapping para optimizar el rendimiento del sistema operativo**

La configuración de Swapping puede ser ajustada para mejorar el rendimiento del sistema operativo de la siguiente forma: Swappiness de sysctl constituye la preferencia del núcleo del sistema de no hacer uso del espacio swap. Swappines se mantiene en valores entre 0 y 100, siendo 60 el usado por defecto. Cuando se establece un valor menor, el intercambio desde la RAM se reduce. Mejorando así la capacidad de respuesta del sistema. Cuando el valor es mayor, el núcleo del sistema hará uso agresivamente del swap. Por lo que es recomendable usar valores bajos para mejorar la capacidad.

La comprobación del valor actual de swappiness se realiza con el siguiente comando:

*sudo cat /proc/sys/vm/swappiness*

El ajuste temporal del valor swappiness se consigue con:

*sudo sysctl vm.swappiness=10*

Si se desea establecer de forma permanente el valor swappiness, se debe editar un archivo de ajuste sysctl:

*/etc/sysctl.d/99-sysctl.conf|2=swappiness=10*

* **Prioridad**

Cuando se tienen varias particiones swap, se debe considerar asignarle valores de prioridad a cada área. El sistema hará so de la partición con mayor prioridad, antes que utilizar las de menor valor. Estas pueden asignarse en fstab desde el parámetro 1=pri de la siguiente forma:

/dev/sda1 none swap defaults,pri=100 0 0

/dev/sdb2 none swap defaults,pri=10 0 0

O a través de los parámetros −p (o −−priority) de la orden swapon:

*sudo swapon -p 100 /dev/sda1*

# **Investigar:**

## *¿Qué es un proceso?*

Un programa o mandato que se ejecuta realmente en el sistema recibe la denominación de proceso.

Los procesos existen en jerarquías de padre-hijo. Un proceso iniciado por un programa o mandato es un proceso padre; un proceso hijo es el producto del proceso padre. Un proceso padre puede tener varios procesos hijo, pero un proceso hijo sólo puede tener un padre.

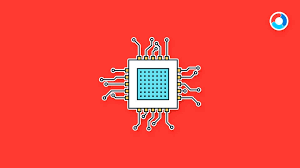
El sistema asigna un número de identificación de proceso (número PID) a cada proceso cuando se inicia. Si inicia el mismo programa varias veces, tendrá un número PID distinto cada vez.

Cuando se inicia un proceso en un sistema, el proceso utiliza una parte de los recursos disponibles en el sistema. Cuando está ejecutándose más de un proceso, un planificador que está incorporado al sistema operativo proporciona a cada proceso su parte de tiempo del sistema, basándose en las prioridades establecidas. Estas prioridades pueden cambiarse mediante la utilización de los mandatos nice o renice.

Nota: Para cambiar una prioridad de proceso a uno superior, debe tener autorización de usuario root. Todos los usuarios pueden reducir las prioridades de un proceso que inician utilizando el mandato nice o de un proceso que ya han iniciado utilizando el mandato renice.

## *¿Qué es un hilo de ejecución?*

Un hilo de ejecución, en los sistemas operativos, es similar a un proceso en que ambos representan una secuencia simple de instrucciones ejecutada en paralelo con otras secuencias. Los hilos permiten dividir un programa en dos o más tareas que corren simultáneamente, por medio de la multiprogramación. En realidad, este método permite incrementar el rendimiento de un procesador de manera considerable. En todos los sistemas de hoy en día los hilos son utilizados para simplificar la estructura de un programa que lleva a cabo diferentes funciones.

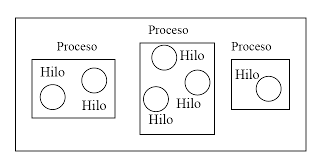
Todos los hilos de un proceso comparten los recursos del proceso. Residen en el mismo espacio de direcciones y tienen acceso a los mismos datos. Cuando un hilo modifica un dato en la memoria, los otros hilos utilizan el resultado cuando acceden al dato. Cada hilo tiene su propio estado, su propio contador, su propia pila y copia de los registros de la CPU. Los valores comunes se guardan en el bloque de control de proceso (PCB), y los valores propios en el bloque de control de hilo (TCB).

Muchos lenguajes de programación (como Java), y otros entornos de desarrollo soportan los llamados hilos o hebras (en inglés, threads).

Un ejemplo de la utilización de hilos es tener un hilo atento a la interfaz gráfica (iconos, botones, ventanas), mientras otro hilo hace una larga operación internamente.

De esta manera el programa responde más ágilmente a la interacción con el usuario.

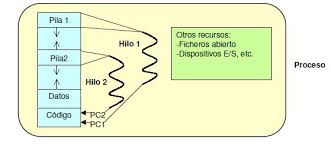
## *La diferencia entre hilos y procesos.*

Los hilos se distinguen de los tradicionales procesos en que los procesos son generalmente independientes, llevan bastante información de estados, e interactúan sólo a través de mecanismos de comunicación dados por el sistema. Por otra parte, muchos hilos generalmente comparten otros recursos directamente. En los sistemas operativos que proveen facilidades para los hilos, es más rápido cambiar de un hilo a otro dentro del mismo proceso, que cambiar de un proceso a otro. Este fenómeno se debe a que los hilos comparten datos y espacios de direcciones, mientras que los procesos al ser independientes no lo hacen. Al cambiar de un proceso a otro el sistema operativo (mediante el dispacher) genera lo que se conoce como overhead, que es tiempo desperdiciado por el procesador para realizar un cambio de modo (mode switch), en este caso pasar del estado de Runnig al estado de Waiting o Bloqueado y colocar el nuevo proceso en Running. En los hilos como pertenecen a un mismo proceso al realizar un cambio de hilo este overhead es casi despreciable.

Sistemas operativos como Windows NT, OS/2 y Linux (2.5 o superiores) han dicho tener hilos "baratos", y procesos "costosos" mientras que en otros sistemas no hay una gran diferencia.

* **Ventajas de los hilos frente a los procesos.**

Si bien los hilos son creados a partir de la creación de un proceso, podemos decir que un proceso es un hilo de ejecución, conocido como monohilo. Pero las ventajas de los hilos se dan cuando hablamos de Multihilos, un proceso tiene múltiples hilos de ejecución los cuales realizan actividades distintas, que pueden o no ser cooperativas entre sí. Los beneficios de los hilos se derivan de las implicaciones de rendimiento y se pueden resumir en los siguientes puntos:

* + Se tarda mucho menos tiempo en crear un hilo nuevo en un proceso existente que en crear un proceso. Algunas investigaciones llevan al resultado que esto es así en un factor de 10.
  + Se tarda mucho menos en terminar un hilo que un proceso, ya que su cuando se elimina un proceso se debe eliminar el PCB de este, mientras que un hilo se elimina su contexto y pila.
  + Se tarda mucho menos tiempo en cambiar entre dos hilos de un mismo proceso.
  + Los hilos aumentan la eficiencia de la comunicación entre programas en ejecución. En la mayoría de los sistemas en la comunicación entre procesos debe intervenir el núcleo para ofrecer protección de los recursos y realizar la comunicación misma. En cambio, entre hilos pueden comunicarse entre sí sin la invocación al núcleo. Por lo tanto, si hay una aplicación que debe implementarse como un conjunto de unidades de ejecución relacionadas, es más eficiente hacerlo con una colección de hilos que con una colección de procesos separados.

## *Características de los procesos.*

1. **Estructura de un Proceso**

Un proceso consta de varios componentes esenciales:

* + Código del programa: Las instrucciones que el CPU ejecuta.
  + Datos: Variables y estructuras de datos utilizadas por el programa.
  + Pila: Almacena las llamadas a funciones y datos locales.
  + Heap: Memoria dinámica asignada en tiempo de ejecución.
  + Contexto de CPU: Contiene registros y el contador de programa (PC).

1. **Estados de un Proceso**

Un proceso puede encontrarse en diferentes estados durante su ciclo de vida:

* Nuevo: El proceso ha sido creado, pero aún no ha sido ejecutado.
* Listo: Está esperando su turno para ser ejecutado por la CPU.
* En ejecución: Actualmente está utilizando el procesador.
* Esperando: Está en pausa porque necesita un recurso (como entrada/salida).
* Terminado: Ha finalizado su ejecución y liberado sus recursos.

1. **Planificación de Procesos**

Dado que hay más procesos que CPUs disponibles, el sistema operativo usa algoritmos de planificación para decidir cuál se ejecutará en un momento dado. Algunos algoritmos comunes son:

* FIFO (First In, First Out): El proceso más antiguo se ejecuta primero.
* Round Robin: Cada proceso recibe una pequeña cantidad de tiempo de CPU antes de pasar al siguiente.
* Prioridad: Los procesos con mayor prioridad se ejecutan primero.
* SJF (Shortest Job First): Se ejecuta primero el proceso más corto.

1. **Prioridad de los Procesos**

Cada proceso tiene una prioridad asignada, que puede ser estática (definida al inicio) o dinámica (ajustada por el sistema operativo en función del uso de CPU y otros factores). Los procesos de mayor prioridad se ejecutan antes.

1. **Multiprocesamiento y Multitarea**

En sistemas modernos, varios procesos pueden ejecutarse simultáneamente en distintos núcleos de CPU (multiprocesamiento). Incluso en sistemas de un solo núcleo, la multitarea permite que los procesos alternen su ejecución rápidamente, simulando simultaneidad.

1. **Interrupciones**

El sistema operativo puede interrumpir un proceso cuando ocurre un evento importante, como:

* Interrupciones de hardware: Generadas por dispositivos como el teclado, el disco duro o una impresora. Por ejemplo, cuando se presiona una tecla, el sistema operativo interrumpe el proceso en ejecución para procesar la entrada.
* Interrupciones de software: Ocurren cuando un programa solicita un servicio del sistema operativo, como la apertura de un archivo o la asignación de memoria.
* Interrupciones de tiempo: Se utilizan en sistemas multitarea para garantizar que un proceso no monopolice la CPU.

1. **Sincronización de Procesos**

Cuando varios procesos comparten recursos (como archivos o memoria), es fundamental que su ejecución esté coordinada para evitar conflictos. Algunas técnicas de sincronización incluyen:

* Semáforos: Variables especiales que controlan el acceso a los recursos compartidos.
* Exclusión mutua (Mutex): Mecanismo que asegura que solo un proceso pueda acceder a un recurso en un momento dado.
* Monitores: Estructuras de alto nivel que combinan mutex y variables de condición para gestionar la concurrencia.

1. **Comunicación entre Procesos (IPC, Inter-Process Communication)**

Los procesos pueden necesitar intercambiar información entre sí. Para esto, existen varios mecanismos:

* Memoria compartida: Varios procesos acceden a la misma región de memoria.
* Colas de mensajes: Un proceso envía mensajes a otro mediante el sistema operativo.
* Tuberías (Pipes): Permiten la comunicación entre procesos mediante un canal de datos, usado comúnmente en sistemas UNIX/Linux.
* Sockets: Usados en redes para la comunicación entre procesos en diferentes dispositivos.

1. **Consumo de Recursos**

Cada proceso necesita recursos del sistema, como:

* CPU: Para ejecutar instrucciones.
* Memoria RAM: Para almacenar datos y código en ejecución.
* Disco duro o SSD: Para cargar archivos o guardar información temporal.
* Dispositivos de entrada/salida (E/S): Como teclado, ratón, impresora, etc.

El sistema operativo gestiona estos recursos para evitar sobrecarga y garantizar que cada proceso tenga lo necesario para funcionar.

1. **Finalización de Procesos**

Un proceso puede finalizar de varias maneras:

* Finalización normal: El proceso completa su tarea y termina por sí solo.
* Finalización por error: Se produce cuando ocurre un fallo, como una división por cero o un intento de acceder a memoria no permitida.
* Terminación forzada: El sistema operativo o un usuario finaliza el proceso manualmente (por ejemplo, con el comando kill en Linux o el Administrador de Tareas en Windows).

# **Conclusiones**

Después de investigar sobre la memoria, los procesos y los hilos de ejecución, puedo concluir que estos conceptos son fundamentales para el funcionamiento de cualquier sistema informático. La memoria, en sus diferentes tipos y clasificaciones, es un recurso esencial que permite la ejecución eficiente de programas y el almacenamiento de datos, tanto de manera temporal como permanente. La evolución de la memoria RAM y del almacenamiento secundario ha sido clave en el desarrollo de sistemas más rápidos y eficientes, lo que ha permitido mejorar el rendimiento de computadoras y dispositivos móviles a lo largo del tiempo.

Por otro lado, comprender cómo funcionan los procesos y los hilos de ejecución me ha ayudado a entender la manera en que los sistemas operativos gestionan múltiples tareas al mismo tiempo. Los procesos permiten la ejecución de programas de manera independiente, mientras que los hilos optimizan los recursos al ejecutar múltiples tareas dentro de un mismo proceso. La diferencia entre ambos es clave en la optimización del rendimiento y en el diseño de aplicaciones que requieren alto procesamiento y multitarea.

En general, aprender sobre estos temas me ha hecho valorar la importancia de la optimización de los recursos en la computación moderna. La manera en que los sistemas operativos administran la memoria y ejecutan múltiples tareas es crucial para el buen desempeño de cualquier dispositivo, desde computadoras personales hasta servidores y sistemas embebidos. Conocer estos fundamentos es esencial para cualquier persona interesada en el desarrollo de software o en el funcionamiento interno de los sistemas informáticos.

# **Bibliografía**

AIX. (2024). Procesos. IBM. Recuperado el 04 de marzo de 2024 de https://www.ibm.com/docs/es/aix/7.2?topic=processes-

Aller, A. (2020). Tamaño de las memorias: Historia y como han evolucionado hasta ahora. Profesional review. Recuperado el 04 de marzo de 2025 de https://www.profesionalreview.com/2020/05/17/tamano-memorias-historia/

IT-ges. (s.f.). Qué es Swapping en sistemas operativos. IT-ges.com. Recuperado el 04 de marzo de 2025 de https://www.it-ges.com/que-es-swapping-en-sistemas-operativos/

PureStorage. (2025). ¿Qué es la memoria informática? PureStorage. Recuperado el 04 de marzo de 2025 de https://www.purestorage.com/la/knowledge/what-is-computer-memory.html

s.a. (s.f.) Gestión de hilos de ejecución. [PDF]. Recuperado el 040 de 2025 de https://biblus.us.es/bibing/proyectos/abreproy/11320/fichero/Capitulos%252F13.pdf

Wikipedia. (2024). Memoria (informática). Wikipedia. Recuperado el 04 de marzo de 2025 de https://es.wikipedia.org/wiki/Memoria\_(inform%C3%A1tica)