Arquitectura de Computadoras

Ing. En Sistemas Computacionales – 5mo Semestre agosto – diciembre 2023

JOSE ARTURO BUSTAMANTE LAZCANO

Contenido

[Semana 1 – Unidad 1 - Arquitecturas de cómputo 1](#_Toc145586354)

[Semana 2 – Unidad 1 – Procesamiento 3](#_Toc145586355)

[Semana 3 – Unidad 1 – Tipos de unidades de procesamiento 7](#_Toc145586356)

[Semana 4 – Unidad 1 - Conceptos básicos del manejo de Memoria 16](#_Toc145586357)

# Semana 1 – Unidad 1 - Arquitecturas de cómputo

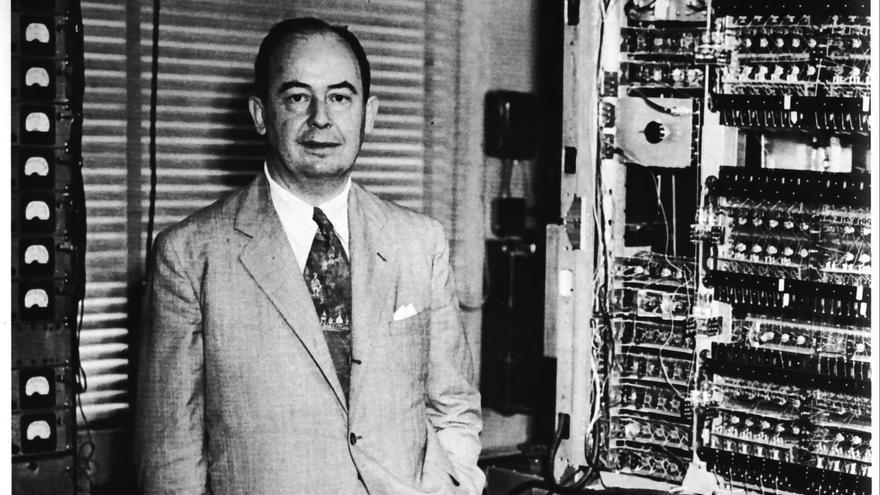
22, 23 y 24 de agosto.

Modelos de arquitecturas de cómputo.

Los modelos de arquitecturas de cómputo son las diferentes formas en que se organizan los componentes de una computadora. La arquitectura de una computadora determina su rendimiento, eficiencia y capacidad.

Los modelos de arquitecturas de cómputo se pueden clasificar en dos categorías principales:

Arquitecturas clásicas: Estas arquitecturas se desarrollaron en las primeras computadoras. La más común es la arquitectura Von Neumann, en la que las instrucciones y los datos se almacenan en la misma memoria. Otra arquitectura clásica es la arquitectura Harvard, en la que las instrucciones y los datos se almacenan en memorias separadas.



Arquitecturas segmentadas: Estas arquitecturas buscan mejorar el rendimiento realizando paralelamente varias etapas del ciclo de instrucción al mismo tiempo. El procesador se divide en varias unidades funcionales independientes y se dividen entre ellas el procesamiento de las instrucciones.

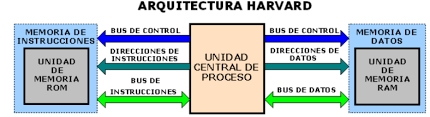
Arquitectura Von Neumann

La arquitectura Von Neumann es la arquitectura de computadora más común. En esta arquitectura, las instrucciones y los datos se almacenan en la misma memoria. El procesador accede a la memoria a través de un bus de datos y un bus de direcciones.

La arquitectura Von Neumann es muy versátil y puede ejecutar una amplia gama de programas. Sin embargo, tiene algunas desventajas, como que puede ser ineficiente para tareas que requieren acceso a datos aleatorios.

Arquitectura Harvard

La arquitectura Harvard es una arquitectura de computadora en la que las instrucciones y los datos se almacenan en memorias separadas. Esto permite que el procesador acceda a las instrucciones y a los datos de forma independiente, lo que puede mejorar el rendimiento para tareas que requieren acceso a datos aleatorios.



La arquitectura Harvard no es tan versátil como la arquitectura Von Neumann, ya que no puede ejecutar todos los programas. Sin embargo, es más eficiente para tareas que requieren acceso a datos aleatorios.

Arquitecturas segmentadas

Las arquitecturas segmentadas buscan mejorar el rendimiento realizando paralelamente varias etapas del ciclo de instrucción al mismo tiempo. El procesador se divide en varias unidades funcionales independientes y se dividen entre ellas el procesamiento de las instrucciones.

Las arquitecturas segmentadas pueden mejorar el rendimiento de tareas que requieren muchas operaciones aritméticas o lógicas. Sin embargo, son más complejas y costosas que las arquitecturas clásicas.

Otros modelos de arquitecturas de cómputo

Además de las arquitecturas clásicas y segmentadas, existen otros modelos de arquitecturas de cómputo, como:

Arquitecturas paralelas: Estas arquitecturas utilizan múltiples procesadores para ejecutar un programa.

Arquitecturas heterogéneas: Estas arquitecturas utilizan procesadores de diferentes tipos, como procesadores de propósito general, procesadores de propósito específico y procesadores gráficos.

El modelo de arquitectura de cómputo que se utiliza en una computadora depende de las necesidades específicas de la aplicación. Por ejemplo, las supercomputadoras utilizan arquitecturas paralelas para ejecutar aplicaciones científicas y de ingeniería que requieren un gran poder de procesamiento. Las computadoras personales utilizan arquitecturas clásicas para ejecutar aplicaciones de uso general.

# Semana 2 – Unidad 1 – Procesamiento

29, 30 y 31 de Agosto

Segmentadas. La arquitectura de computo segmentada es un modelo de diseño que divide la memoria de un sistema en segmentos o regiones lógicas. Cada segmento tiene un tamaño variable y puede contener datos o código de un programa. Los segmentos se identifican por un número y un desplazamiento, que indican la dirección base y la posición relativa dentro del segmento, respectivamente. La ventaja de esta arquitectura es que permite acceder a más memoria que la que se puede direccionar con un registro de tamaño fijo, y facilita la protección y el intercambio de segmentos entre procesos. La desventaja es que aumenta la complejidad del hardware y del software, y requiere más ciclos de reloj para realizar las operaciones de segmentación.

En este artículo vamos a explicar los conceptos básicos de la arquitectura de computo segmentada, sus aplicaciones y sus desafíos. También vamos a presentar algunos ejemplos de sistemas que utilizan esta arquitectura, como el Intel 8086, el IBM System/360 y el Multics. Finalmente, vamos a analizar las ventajas y desventajas de este modelo frente a otros modelos alternativos, como la arquitectura de computo paginada o la arquitectura de computo plana.

De multiprocesamiento.

La arquitectura de computo de múltiple procesamiento es una forma de operación de una computadora en donde se tiene físicamente más de un procesador. El fin es tener la disposición de ejecutar diferentes partes de un programa a la vez. Estas múltiples unidades centrales de procesamiento (CPU) se mantienen en una estrecha comunicación, compartiendo el bus, la memoria y otros dispositivos periféricos de la computadora.

El multiprocesamiento se refiere más a la cantidad de unidades de CPU, que a la cantidad de procesos en ejecución de forma simultánea. Si el hardware proporciona más de un procesador, entonces eso es multiprocesamiento. Es la capacidad del sistema de aprovechar la potencia informática de múltiples procesadores. El sistema de multiprocesamiento es de gran utilidad cuando se busca tener una velocidad bastante alta para procesar un conjunto elevado de datos. Estos sistemas se usan mayormente en aplicaciones como el de pronóstico del tiempo, control satelital, etc.

Existen diferentes tipos de arquitecturas de computo de múltiple procesamiento, según el grado de acoplamiento entre los procesadores y la forma en que se distribuyen las tareas. Algunas de las más comunes son:

- \*\*Multiprocesamiento simétrico (SMP)\*\*: Es un sistema único con múltiples procesadores estrechamente acoplados que comparten todos los recursos del sistema, como la memoria, el disco y los dispositivos de entrada/salida. Los procesadores pueden ejecutar el mismo o diferentes programas, y el sistema operativo se encarga de asignar las tareas y balancear la carga entre ellos. Esta arquitectura es simple y eficiente para aplicaciones que requieren acceso frecuente a los datos compartidos.

- \*\*Multiprocesamiento asimétrico (AMP)\*\*: Es un sistema único con múltiples procesadores que tienen funciones específicas y no comparten todos los recursos del sistema. Por ejemplo, un procesador puede ser el maestro y coordinar las actividades de los demás, que serían los esclavos. Cada procesador puede tener su propio sistema operativo o ejecutar una parte del mismo. Esta arquitectura es más compleja y adecuada para aplicaciones que requieren una alta especialización o independencia entre los procesadores.

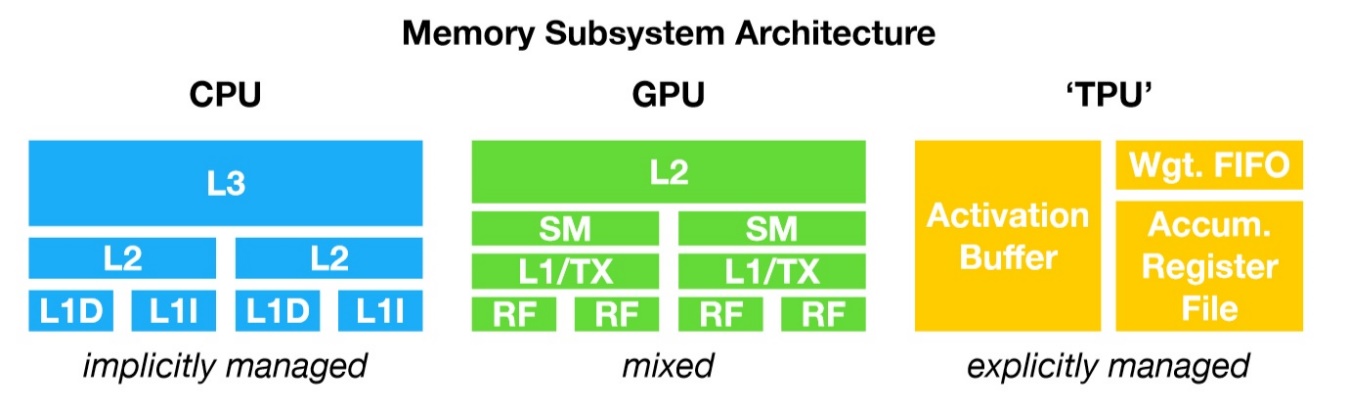
- \*\*Multiprocesamiento masivamente paralelo (MPP)\*\*: Es una matriz de nodos de procesamiento independientes que se comunican a través de un bus de interconexión de alta velocidad. Cada nodo tiene su propio procesador, memoria y disco, y puede ejecutar su propio sistema operativo o una parte del mismo. Los nodos pueden trabajar en paralelo en diferentes partes de un problema o cooperar en una misma tarea. Esta arquitectura es muy escalable y potente para aplicaciones que requieren una gran cantidad de cálculos o datos.

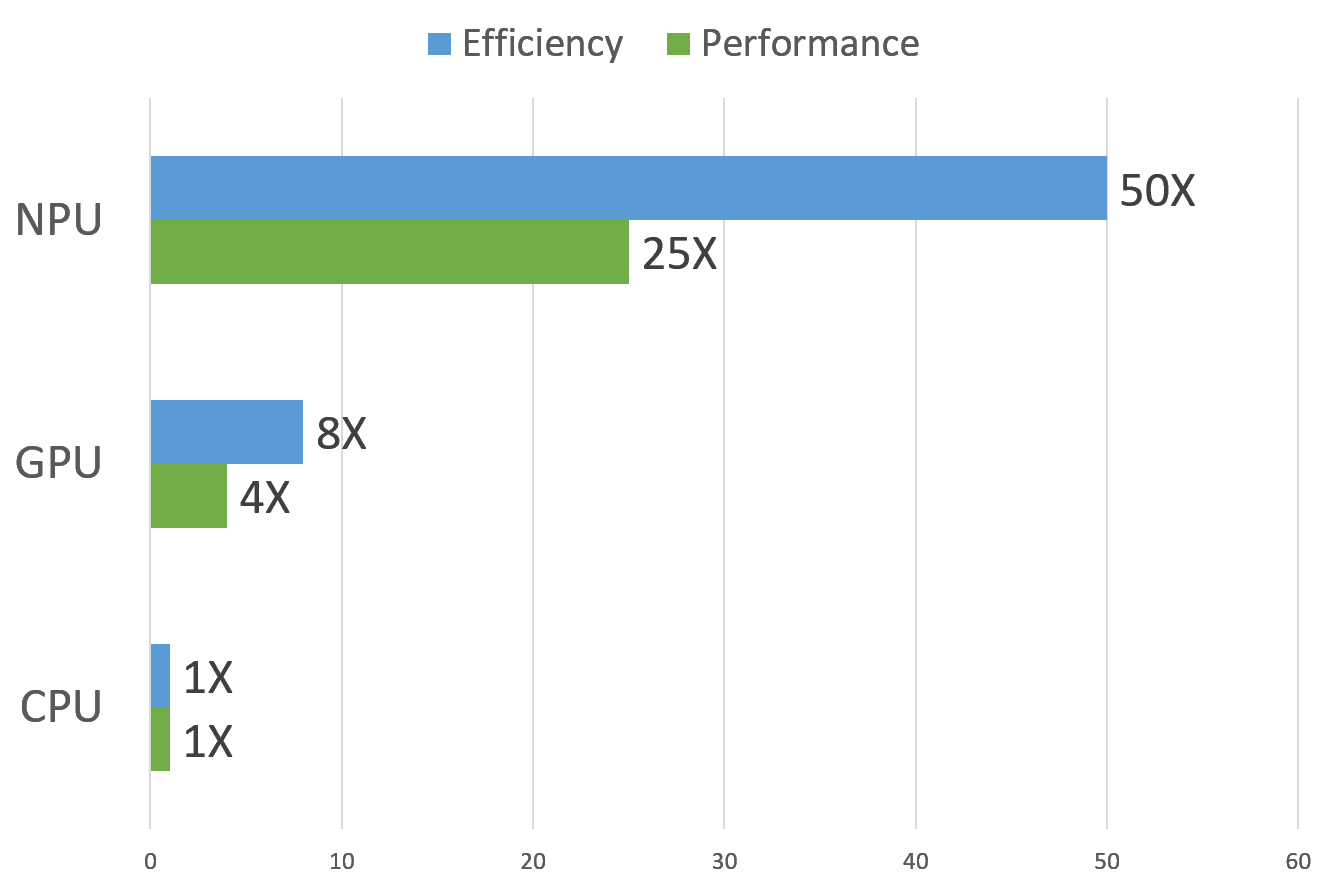
- \*\*Computación distribuida o en malla (Grid)\*\*: Es una red de computadoras heterogéneas que se conectan a través de Internet u otras redes para compartir recursos y datos. Las computadoras pueden estar ubicadas en diferentes lugares geográficos y pertenecer a diferentes organizaciones. Los usuarios pueden acceder a los recursos disponibles según sus necesidades y preferencias, sin importar su ubicación física. Esta arquitectura es muy flexible y dinámica para aplicaciones que requieren una gran diversidad o disponibilidad de recursos.

# Semana 3 – Unidad 1 – Tipos de unidades de procesamiento

05, 06 y 07 de septiembre

Tipos de Arquitectura y Componentes de computo





CPU

La arquitectura de la CPU es el diseño conceptual y la estructura operacional fundamental de un sistema de computadoras. Define la estructura y el funcionamiento de los componentes internos de la CPU, que son los encargados de realizar las operaciones necesarias para ejecutar programas.

Las características principales de la arquitectura de la CPU son:

Tipo de instrucciones: Los procesadores pueden utilizar dos tipos de instrucciones: CISC (Complex Instruction Set Computer) y RISC (Reduced Instruction Set Computer). Los procesadores CISC tienen un conjunto de instrucciones más amplio, lo que les permite realizar operaciones más complejas en una sola instrucción. Los procesadores RISC, por su parte, tienen un conjunto de instrucciones más reducido, lo que los hace más eficientes en el uso de los recursos del sistema.

Tamaño de palabra: El tamaño de palabra es la cantidad de bits que puede procesar la CPU a la vez. Los procesadores de 32 bits pueden procesar 32 bits a la vez, mientras que los procesadores de 64 bits pueden procesar 64 bits a la vez. Un tamaño de palabra más grande permite procesar datos más grandes en una sola operación, lo que puede mejorar el rendimiento de la CPU.

Frecuencia de reloj: La frecuencia de reloj es la velocidad a la que la CPU funciona. Se mide en hercios (Hz), y un hercio equivale a un ciclo por segundo. Una frecuencia de reloj más alta permite que la CPU ejecute más instrucciones por segundo, lo que puede mejorar el rendimiento general del sistema.

Número de núcleos: Los procesadores modernos pueden tener uno o más núcleos. Un núcleo es un procesador independiente que puede ejecutar instrucciones de forma simultánea. Un mayor número de núcleos permite que la CPU ejecute más tareas a la vez, lo que puede mejorar el rendimiento multitarea.

Las funciones principales de la CPU son:

Ejecutar instrucciones: La CPU es responsable de ejecutar las instrucciones del programa que está en ejecución. Para ello, la CPU decodifica las instrucciones, obtiene los datos necesarios de la memoria y realiza las operaciones necesarias.

Gestionar la memoria: La CPU también es responsable de gestionar la memoria del sistema. Esto incluye asignar memoria a los programas, liberar memoria cuando ya no es necesaria y evitar que los programas accedan a la memoria de otros programas.

Gestionar los periféricos: La CPU también es responsable de gestionar los periféricos del sistema, como el teclado, el mouse y la pantalla. Esto incluye enviar datos a los periféricos y recibir datos de ellos.

Las partes principales que componen la arquitectura de la CPU son:

Unidad de control: La unidad de control es el cerebro de la CPU. Es responsable de decodificar las instrucciones, asignar recursos y coordinar las operaciones de la CPU.

Unidad aritmético-lógica (ALU): La ALU es la unidad que realiza las operaciones aritméticas y lógicas. Esto incluye operaciones como suma, resta, multiplicación, división, AND, OR y NOT.

Registros: Los registros son memorias de alta velocidad que se utilizan para almacenar datos temporales. La CPU utiliza los registros para almacenar datos que necesita para ejecutar instrucciones.

Memoria caché: La memoria caché es una memoria de alta velocidad que se utiliza para almacenar datos que la CPU ha accedido recientemente. La memoria caché ayuda a mejorar el rendimiento de la CPU al reducir la necesidad de acceder a la memoria principal, que es más lenta.

Bus: El bus es la interfaz que conecta la CPU con el resto del sistema. Utilizado para transferir datos entre la CPU y la memoria, los dispositivos periféricos y otros componentes de la computadora.

La arquitectura de la CPU es un tema complejo que ha evolucionado a lo largo de los años. Los avances en la arquitectura de la CPU han permitido que las computadoras se vuelvan más rápidas, eficientes y potentes.

GPU

Las características principales de la arquitectura GPU son:

Paralelismo masivo: Las GPU están diseñadas para procesar grandes cantidades de datos de forma paralela. Esto se logra mediante la agrupación de miles o millones de núcleos de procesamiento en una sola unidad.

Ejecución de instrucciones vectoriales: Las GPU pueden ejecutar instrucciones vectoriales, que son instrucciones que operan en múltiples datos al mismo tiempo. Esto las hace ideales para tareas que involucran matrices de datos, como el procesamiento de imágenes y el aprendizaje automático.

Memoria dedicada: Las GPU tienen su propia memoria dedicada, que está separada de la memoria de la CPU. Esto permite que las GPU accedan a los datos de forma más rápida y eficiente.

Las funciones de la arquitectura GPU incluyen:

Procesamiento de gráficos: Las GPU fueron originalmente diseñadas para el procesamiento de gráficos 2D y 3D. Sin embargo, con el tiempo, se han vuelto más versátiles y ahora se utilizan para una amplia gama de tareas, como el aprendizaje automático, el análisis de datos y la simulación.

Computación acelerada: Las GPU pueden utilizarse para acelerar el rendimiento de las aplicaciones que realizan cálculos intensivos. Esto se logra mediante el uso de la paralelización y la ejecución de instrucciones vectoriales.

Las partes que componen la arquitectura GPU incluyen:

Núcleos de procesamiento: Los núcleos de procesamiento son los componentes básicos de una GPU. Son responsables de ejecutar las instrucciones y realizar los cálculos.

Memoria de la GPU: La memoria de la GPU es donde se almacenan los datos que están siendo procesados por la GPU.

Controlador de memoria: El controlador de memoria es responsable de gestionar el acceso a la memoria de la GPU.

Unidad de control: La unidad de control es responsable de coordinar las operaciones de la GPU.

Las GPU se han convertido en una parte integral de las computadoras modernas. Su capacidad de paralelizar el procesamiento de datos y ejecutar instrucciones vectoriales las ha hecho ideales para una amplia gama de tareas, incluyendo el procesamiento de gráficos, el aprendizaje automático, el análisis de datos y la simulación.

GPU

Las características principales de la arquitectura GPU son:

Paralelismo masivo: Las GPU están diseñadas para procesar grandes cantidades de datos de forma paralela. Esto se logra mediante la agrupación de miles o millones de núcleos de procesamiento en una sola unidad.

Ejecución de instrucciones vectoriales: Las GPU pueden ejecutar instrucciones vectoriales, que son instrucciones que operan en múltiples datos al mismo tiempo. Esto las hace ideales para tareas que involucran matrices de datos, como el procesamiento de imágenes y el aprendizaje automático.

Memoria dedicada: Las GPU tienen su propia memoria dedicada, que está separada de la memoria de la CPU. Esto permite que las GPU accedan a los datos de forma más rápida y eficiente.

Las funciones de la arquitectura GPU incluyen:

Procesamiento de gráficos: Las GPU fueron originalmente diseñadas para el procesamiento de gráficos 2D y 3D. Sin embargo, con el tiempo, se han vuelto más versátiles y ahora se utilizan para una amplia gama de tareas, como el aprendizaje automático, el análisis de datos y la simulación.

Computación acelerada: Las GPU pueden utilizarse para acelerar el rendimiento de las aplicaciones que realizan cálculos intensivos. Esto se logra mediante el uso de la paralelización y la ejecución de instrucciones vectoriales.

Las partes que componen la arquitectura GPU incluyen:

Núcleos de procesamiento: Los núcleos de procesamiento son los componentes básicos de una GPU. Son responsables de ejecutar las instrucciones y realizar los cálculos.

Memoria de la GPU: La memoria de la GPU es donde se almacenan los datos que están siendo procesados por la GPU.

Controlador de memoria: El controlador de memoria es responsable de gestionar el acceso a la memoria de la GPU.

Unidad de control: La unidad de control es responsable de coordinar las operaciones de la GPU.

Las GPU se han convertido en una parte integral de las computadoras modernas. Su capacidad de paralelizar el procesamiento de datos y ejecutar instrucciones vectoriales las ha hecho ideales para una amplia gama de tareas, incluyendo el procesamiento de gráficos, el aprendizaje automático, el análisis de datos y la simulación.

NPU

Las características principales de la arquitectura NPU son:

Especialización para el procesamiento de redes neuronales: Las NPU están diseñadas específicamente para procesar redes neuronales, que son un tipo de algoritmo de aprendizaje automático. Esto se logra mediante el uso de arquitecturas y circuitos especializados que están optimizados para el funcionamiento de las redes neuronales.

Mayor eficiencia energética: Las NPU suelen ser más eficientes energéticamente que las CPU y GPU tradicionales. Esto se debe a que las NPU están diseñadas para realizar tareas específicas de aprendizaje automático, que a menudo requieren menos energía que otras tareas, como el procesamiento de gráficos o el cálculo científico.

Mayor rendimiento: Las NPU pueden ofrecer un mayor rendimiento que las CPU y GPU tradicionales para tareas de aprendizaje automático. Esto se debe a que las NPU están diseñadas específicamente para estas tareas, que a menudo requieren una gran cantidad de cálculos.

Las funciones de la arquitectura NPU incluyen:

Procesamiento de redes neuronales: Las NPU se utilizan para procesar redes neuronales, que son un tipo de algoritmo de aprendizaje automático. Las redes neuronales se utilizan en una amplia gama de aplicaciones, como la visión artificial, el reconocimiento de voz y el aprendizaje automático.

Aceleración de aplicaciones de aprendizaje automático: Las NPU se utilizan para acelerar las aplicaciones de aprendizaje automático. Esto puede ayudar a mejorar el rendimiento y reducir el consumo de energía de las aplicaciones de aprendizaje automático.

Las partes que componen la arquitectura NPU incluyen:

Núcleos de procesamiento neuronal: Los núcleos de procesamiento neuronal son los componentes básicos de una NPU. Son responsables de ejecutar las operaciones de las redes neuronales.

Memoria de la NPU: La memoria de la NPU es donde se almacenan los datos que están siendo procesados por la NPU.

Controlador de memoria: El controlador de memoria es responsable de gestionar el acceso a la memoria de la NPU.

Unidad de control: La unidad de control es responsable de coordinar las operaciones de la NPU.

Las NPU se han convertido en una parte integral de las computadoras modernas. Su capacidad de procesar redes neuronales de forma eficiente y rápida ha hecho que sean ideales para una amplia gama de aplicaciones de aprendizaje automático.

Aquí hay algunos ejemplos específicos de cómo se utilizan las NPU:

En los smartphones, las NPU se utilizan para mejorar el rendimiento de las cámaras, el reconocimiento facial y otras funciones de inteligencia artificial.

En los centros de datos, las NPU se utilizan para entrenar modelos de aprendizaje automático de gran tamaño.

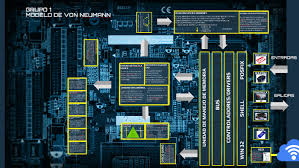
En los automóviles autónomos, las NPU se utilizan para procesar imágenes de las cámaras y otras entradas sensoriales.

A medida que la tecnología de aprendizaje automático continúa desarrollándose, las NPU se volverán aún más importantes.

# Semana 4 – Unidad 1 - Conceptos básicos del manejo de Memoria

12, 13 y 14 de septiembre

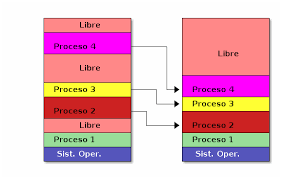
El manejo de memoria es un concepto fundamental en programación y se refiere a cómo un programa gestiona y utiliza la memoria de la computadora para almacenar y acceder a datos. Aquí tienes algunos conceptos básicos relacionados con el manejo de memoria:



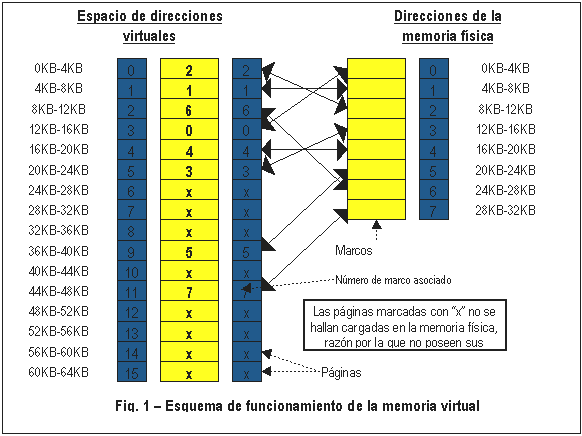
1. \*\*Memoria RAM (Random Access Memory)\*\*: La memoria RAM es un tipo de memoria volátil que se utiliza para almacenar datos y programas mientras se ejecutan en la computadora. Los datos en la RAM se pueden leer y escribir rápidamente, pero se borran cuando se apaga la computadora.



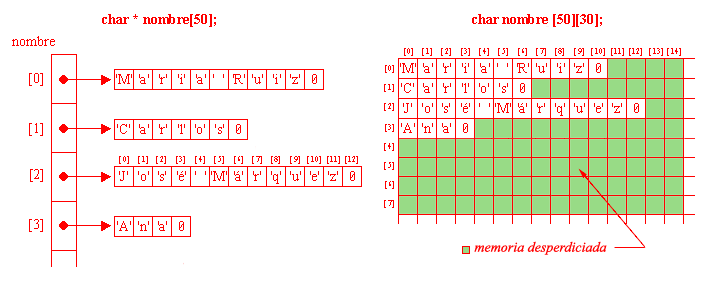
2. \*\*Asignación de memoria\*\*: Es el proceso de reservar un bloque de memoria para su uso en un programa. Esto se hace para almacenar variables, estructuras de datos, objetos u otros tipos de datos.



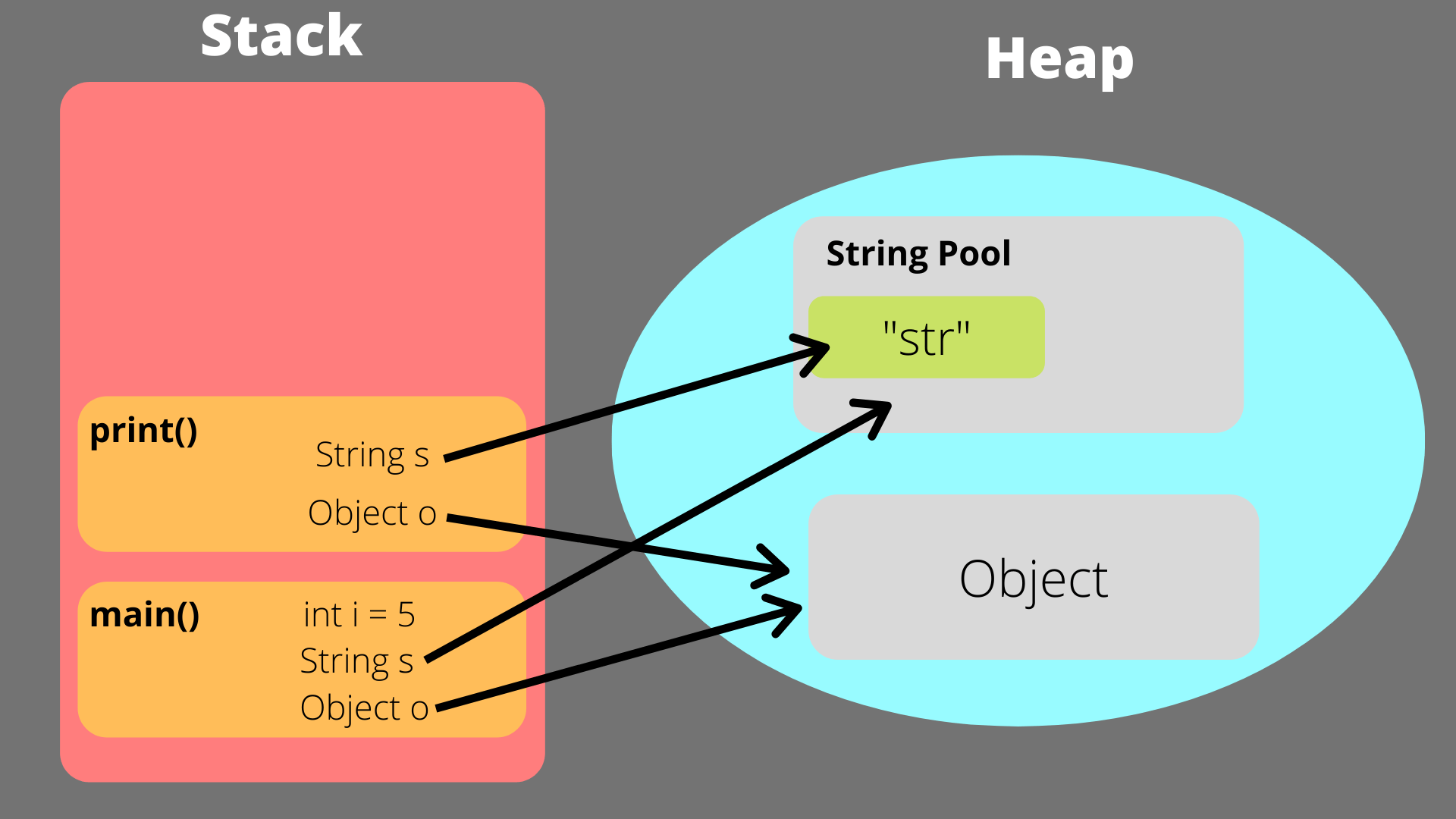
3. \*\*Desasignación de memoria\*\*: También conocida como "liberación de memoria", es el proceso de devolver la memoria previamente asignada a un sistema de gestión de memoria para su reutilización. Esto es importante para evitar fugas de memoria (memory leaks) y garantizar que los recursos se utilicen eficientemente.



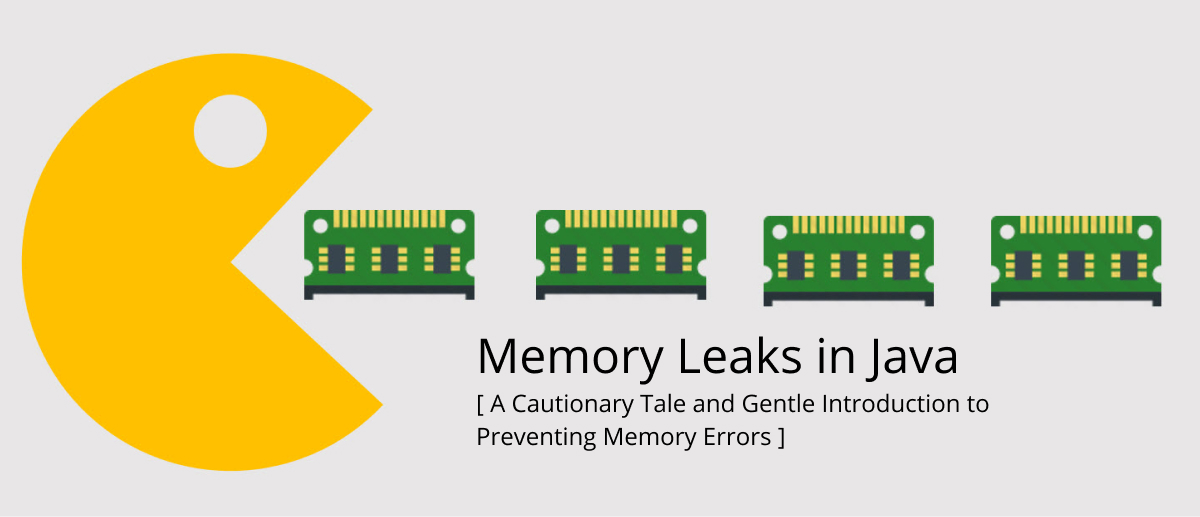
4. \*\*Punteros\*\*: Un puntero es una variable que almacena la dirección de memoria de otro objeto. Se utilizan para acceder y manipular datos en la memoria.



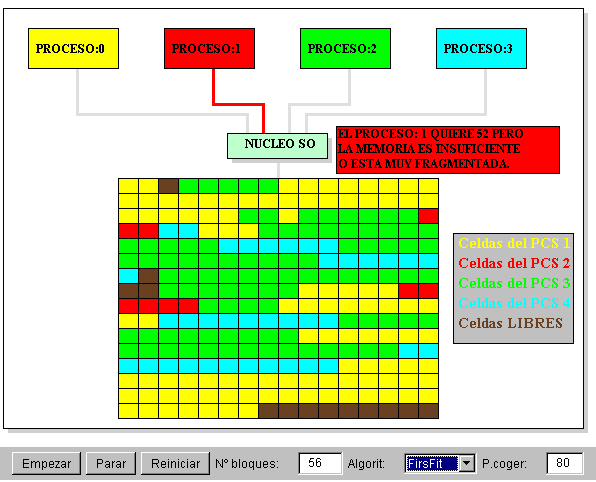
5. \*\*Heap y Stack\*\*: Son dos regiones de memoria utilizadas para diferentes propósitos. El heap se utiliza para la asignación dinámica de memoria, donde los objetos se crean y destruyen explícitamente por el programador. El stack se utiliza para almacenar variables locales y controlar el flujo de ejecución del programa, y su gestión es más automática.



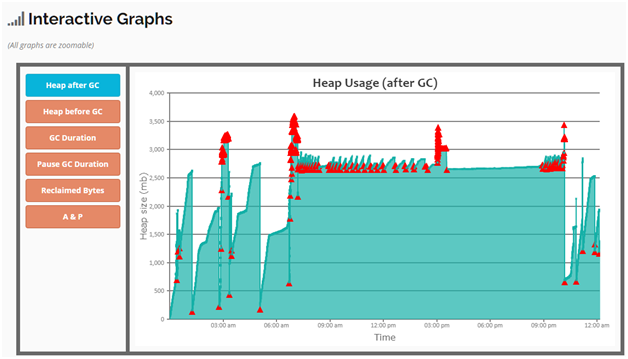
6. \*\*Fuga de memoria (Memory Leak)\*\*: Ocurre cuando un programa asigna memoria dinámicamente pero no la libera adecuadamente cuando ya no se necesita. Esto puede provocar un agotamiento gradual de la memoria disponible y afectar el rendimiento del sistema.



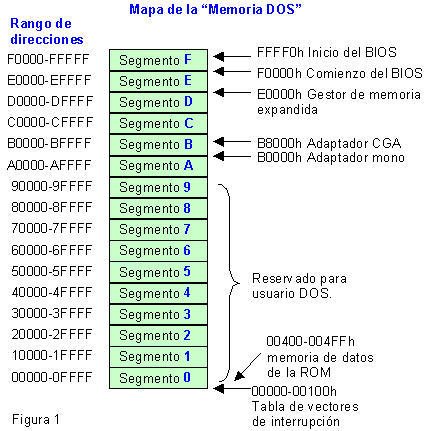
7. \*\*Fragmentación de memoria\*\*: Puede ocurrir en el heap cuando se asigna y libera memoria repetidamente, lo que lleva a la fragmentación de la memoria en pequeños fragmentos no contiguos. Esto puede hacer que sea más difícil asignar bloques grandes de memoria.



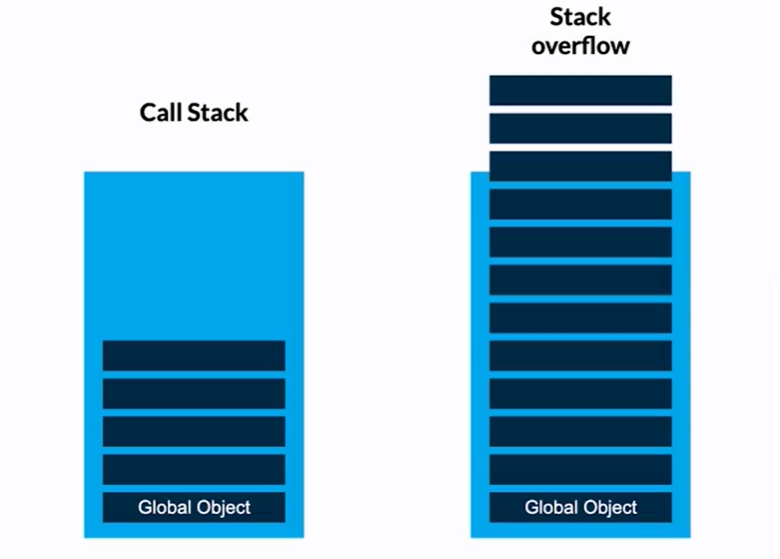
8. \*\*Garbage Collection (Recolección de Basura)\*\*: Es un proceso automatizado en algunos lenguajes de programación (como Java y C#) que busca y libera automáticamente la memoria que ya no está en uso, evitando así fugas de memoria.



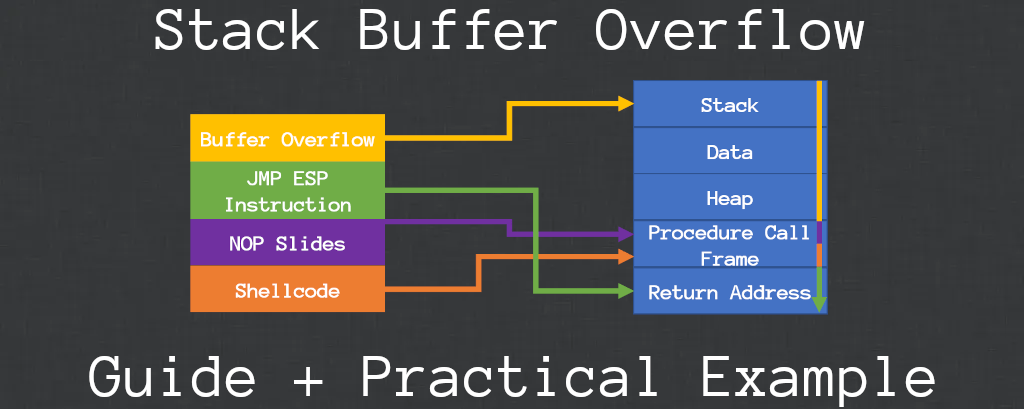
9. \*\*Dirección de memoria\*\*: Cada byte en la memoria tiene una dirección única que se utiliza para acceder a ese byte específico. Las direcciones de memoria suelen ser números hexadecimales.



10. \*\*Stack Overflow\*\*: Ocurre cuando la pila (stack) se llena con llamadas recursivas o funciones anidadas excesivas, lo que provoca un desbordamiento y un error en tiempo de ejecución.



11. \*\*Buffer Overflow\*\*: Es un tipo de error de programación en el que se escriben más datos en un buffer de memoria de lo que puede contener, lo que puede llevar a problemas de seguridad y vulnerabilidades.



El manejo de memoria es una parte crítica de la programación, y entender estos conceptos es esencial para escribir programas eficientes y sin errores relacionados con la memoria. Diferentes lenguajes de programación tienen enfoques diferentes para el manejo de memoria, por lo que es importante estar familiarizado con las características y las mejores prácticas específicas de cada lenguaje que utilices.