**UNIDAD 2. MICROPROCESADORES Y GESTION DE PROCESOS**

**2.1 MICROPROCESADORES**

**Introducción**

En el mundo tecnológico de hoy, gran parte de lo que hacemos cotidianamente está influenciado por la informática. Desde nuestros smartphones y computadoras hasta los electrodomésticos más avanzados, todos tienen algo en común: el microprocesador.

Este componente, visto como el "corazón" de un sistema informático, es esencial para interpretar y llevar a cabo instrucciones, asegurando que nuestras aplicaciones y sistemas funcionen con eficacia.

Para quienes se aventuran en el mundo del desarrollo web, entender el funcionamiento de los microprocesadores no es solo interesante, sino vital. Este entendimiento posibilita a los creadores afinar sus aplicaciones, asegurando un rendimiento más ágil y brindando una interacción más suave para el usuario.

Entonces, ¿estás preparado para sumergirte en el corazón de la informática y explorar cómo estos pequeños circuitos transformaron nuestra era digital? ¡Vamos allá!

**1. ¿Qué es un microprocesador y cómo funciona?**

Para saber todo lo demás, debemos definir lo que es un microprocesador. El microprocesador es el núcleo de una computadora y está compuesto por un circuito integrado compuesto por millones de transistores encapsulados en un chip de silicio. Se encarga del procesamiento de datos, del control de la mayoría de los dispositivos de la computadora y, lo más importante, de realizar las operaciones lógicas y matemáticas.

Los bits, que son señales de unos y ceros, constituyen todos los datos que nuestra máquina transmite. Las instrucciones y los programas se crean agrupando cada una de estas señales en un conjunto de bits. Todo esto se interpreta por el microprocesador mediante la realización de operaciones básicas como SUM, RESTA, AND, OR, MUC, DIV, OPUESTO e INVERSO. Luego de lo que se encarga el microprocesador es:

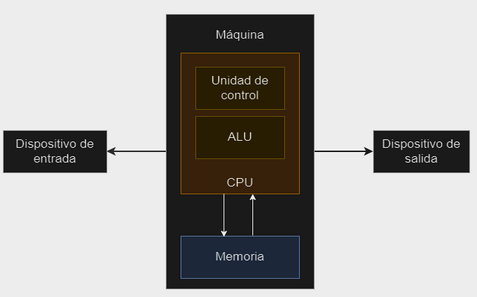
* Decodifica y ejecuta las instrucciones de los programas que se encuentran en la memoria principal del dispositivo.
* Controla y organiza todos los componentes que componen el ordenado, incluidos el mouse, el teclado, la impresora, la pantalla y otros periféricos.

Actualmente, los procesadores son de forma cuadrada o rectangular y se colocan sobre un elemento conocido como zócalo que se fija a la placa base. Esta será responsable de distribuir los datos entre el procesador y los componentes conectados.

**2. Arquitectura (Von Neumann)**

Desde la creación de los microprocesadores hasta la actualidad, estos se basan en una arquitectura que divide el procesador en varios componentes, como veremos más adelante. Esta se llama arquitectura de Von Neumann. Es una estructura creada por el matemático Von Neumann en 1 945 y describe cómo se dividió el diseño de un computador digital en una serie de componentes o componentes.

La mayoría de los procesadores actuales se basan en esta arquitectura básica, aunque lógicamente se han agregado muchos elementos nuevos hasta llegar a los elementos extremadamente completos que tenemos actualmente. Posibilidad de almacenar múltiples números en un mismo chip, elementos de memoria en diferentes niveles, procesador de gráficos integrado, etc.



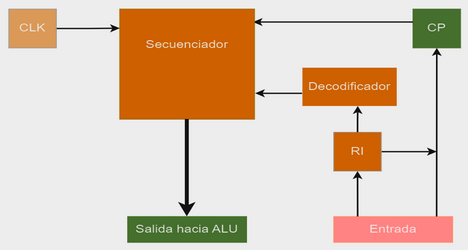
Las siguientes son las **partes fundamentales** de una computadora basada en esta arquitectura:

* La **memoria** es donde se guardan los datos y las instrucciones que ejecuta el computador. El nombre del programa se deriva de estas instrucciones.
* La **unidad central de proceso**, también conocida como CPU, es el componente que ya hemos descrito anteriormente. Es responsable de procesar las instrucciones recibidas de la memoria.
* La **unidad de entrada y salida** permite que los elementos del exterior se comuniquen.
* Las pistas, vías o cables que conectan físicamente los elementos anteriores se conocen como **buses de datos**.

**3. Elementos de un microprocesador**

La **unidad de control (UC)** es la parte que administra las señales de control, como el reloj. El decodificador de instrucciones recibe las instrucciones de la memoria principal y las ejecuta. Partes internas:

* **Reloj:** genera una onda cuadrada para sincronizar las operaciones del procesador.
* **Contador de programas (PC):** contiene la dirección de memoria de la instrucción posterior que se ejecutará.
* **Registro de instrucciones:** almacena la instrucción que actualmente se está ejecutando.
* **Secuenciador:** crea instrucciones básicas para el procesamiento de instrucción.
* **El decodificador de instrucciones (DI):** extrae el código de operación de las instrucciones y las interpreta y ejecuta.



La **unidad aritmética lógica**, también **ALU**, es responsable de realizar cálculos aritméticos, como suma, resta, multiplicación y división, así como operaciones lógicas, como AND, OR, etc. Partes internas:

* **Circuito de operación:** incluye multiplexores y circuitos de operación.
* **Registros de entrada:** antes de que los datos ingresen al circuito operacional, se almacenan.
* **Acumulador:** almacena los resultados de las operaciones
* El **estado de registro (Flag):** almacena condiciones específicas que deben tenerse en cuenta para operaciones futuras.

**Banco de registros y la memoria caché (Caché):** los procesadores actuales cuentan con una memoria volátil que hace de puente desde la memoria RAM hasta la CPU. Esta es mucho más rápida que la memoria RAM y se encarga de acelerar los accesos del microprocesador a la memoria principal.

El **bus frontal (FSB),** también se conoce como **bus de datos**, bus principal o bus de sistema. Es el canal o ruta que conecta el microprocesador a la placa base, en particular al chip conocido como el puente norte o nothbridge. Este controla el bus principal de la CPU, la RAM y los puertos de expansión PCI-Express. Este bus se define como "Interconexión de camino rápido" para Intel y "Hypertransport" para AMD.

El **bus trasero**, también conocido como **BSB**, permite que la memoria caché de nivel 2 (L2) se comunique con el procesador, siempre que no esté incorporada en el núcleo de la CPU. Actualmente, todos los microprocesadores cuentan con memoria caché incorporada en su chip, lo que significa que este bus también está incluido en el mismo chip.

**4. Funcionamiento de un microprocesador**

Un procesador funciona mediante instrucciones, cada una de las cuales es un código binario de una extensión específica que la CPU puede comprender.

Como resultado, un programa es un conjunto de instrucciones que se deben ejecutar de forma secuencial, es decir, ejecutando una de estas instrucciones en cada paso o intervalo de tiempo para poder ejecutarlo.

Hay varias etapas que deben completarse para ejecutar una instrucción:

* **Búsqueda de instrucción:** obtenemos la instrucción desde la memoria y la transferimos al procesador.
* **La decodificación:** consiste en dividir la instrucción en códigos más fáciles de entender por la CPU.
* **Búsqueda de operados:** al cargar una instrucción en la CPU, debe buscar el operador correspondiente.
* **Ejecución:** implica realizar las operaciones lógicas o aritméticas requeridas.
* **Almacenamiento del resultado:** el resultado se almacena en la memoria caché.

Cada procesador utiliza un conjunto de instrucciones específico, el cual ha cambiado a medida que han evolucionado los procesadores. El nombre x86 0 x386 se refiere al conjunto de instrucciones de un procesador.

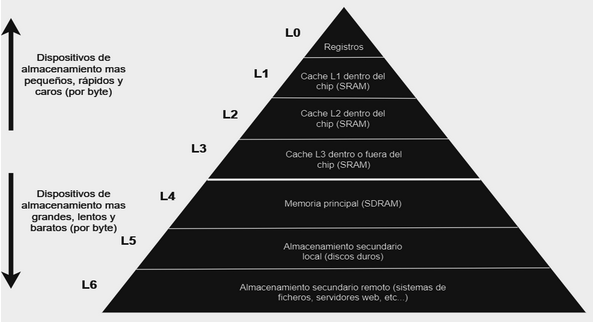
Los procesadores de 32 bits también se conocen como x86 debido a que utilizaron el conjunto de instrucciones del procesador Intel 80386, el primero en implementar la arquitectura de 32 bits.

**5. Memoria caché**

Aunque son mucho más pequeñas que la RAM, estas memorias funcionan mucho más rápido. Su función es almacenar las instrucciones que se van a procesar o las que se han procesado recientemente. La velocidad de transacciones que la CPU puede coger y soltar aumenta con la memoria caché.

Aquí debemos tener en cuenta que todo lo que llega al procesador proviene del disco duro, que es significativamente más lento que la memoria RAM y aún mucho más lento que la memoria caché.

Estas memorias en estado sólido se diseñaron para resolver el gran problema del disco duro. Si nos preguntamos por qué no crean solo memorias cachés de gran tamaño, la respuesta es sencilla: porque son muy costosas.



Cuando miramos un procesador, la velocidad de Internet es casi siempre lo más notable. "El procesador tiene 3,2 GHz", pero, ¿qué es esto? La velocidad del microprocesador es su frecuencia de reloj. Esta velocidad puede realizar más operaciones por unidad de tiempo. La memoria caché se crea para acelerar la toma de datos del procesador y realizar el máximo número de operaciones por unidad de tiempo, lo que da mayor rendimiento.

Una señal de onda cuadrada periódica proporciona esta frecuencia de reloj. La duración máxima de una operación es de un período. La frecuencia es la inversa del período.

La velocidad no es lo único que importa. La velocidad de un procesador está influenciada por una variedad de componentes. Por ejemplo, si tenemos un procesador de cuatro núcleos a 1,8 GHz y otro de un solo núcleo a 4,0 GHz, el procesador de cuatro núcleos es seguro que funciona más rápido.

**6. Velocidad del bus**

La velocidad del bus de datos es igual de importante que la velocidad del procesador.

La frecuencia de reloj de la placa base es mucho menor que la del microprocesador, por lo que será necesario un multiplicador para ajustar esta frecuencia.

Si tenemos una placa base con un bus que funciona en una frecuencia de reloj de 200 MHz, un multiplicador de 10x podría lograr una frecuencia de CPU de 2 GHz.

**7. Refrigeración de componentes**

El paso de corriente genera calor debido a la enorme velocidad de la CPU. Es necesario refrigerar este componente porque una mayor frecuencia y voltaje generará más calor. Hay varias formas de hacerlo:

* La **refrigeración pasiva** se logra a través de disipadores metálicos, como cobre o aluminio, que utilizan aletas para aumentar la superficie de contacto con el aire.
* La **refrigeración activa** utiliza un ventilador junto con el disipador para proporcionar un flujo de aire forzado entre las aletas del elemento pasivo.
* La **refrigeración líquida** se compone de un circuito con una bomba y un radiador aleteado. Un bloque en la CPU circula el agua; el elemento líquido recoge el calor y lo transporta al radiador, que reduce la temperatura del líquido mediante ventilación forzada.

**2.2 GESTIÓN DE PROCESOS. ALGORITMOS**

La gestión de procesos, también conocida como planificación de procesos, es un conjunto de mecanismos integrados en el sistema operativo que establecen el orden en que se realizan los diversos procesos. El objetivo principal de una buena planificación es brindar un buen servicio a todos los procesos que existen en un momento dado en el sistema.

**1. Criterios de planificación**

Los criterios que se deben tener en cuenta a la hora de elegir o diseñar un algoritmo de planificación son los siguientes:

* **Tiempo de respuesta:** Velocidad con que el ordenador da respuesta a una petición. Depende mucho de la velocidad de los dispositivos de entrada/salida.
* **Tiempo de servicio:** Es la duración de un proceso, es decir, desde que se solicita la ejecución hasta que se completa. Este tiempo incluye el tiempo de ejecución del procesador, el tiempo de espera en la cola de procesos preparados, el tiempo de carga del programa en memoria y el tiempo consumido en operaciones de entrada/salida.
* **Tiempo de ejecución:** Es el tiempo teórico que necesitaría el proceso para ejecutarse si fuera el único proceso disponible en el sistema, es decir, es igual al tiempo de servicio menos el tiempo de espera en la cola de procesos preparados.
* **Tiempo de procesador:** Es la cantidad de tiempo que un proceso utiliza el procesador sin contar el tiempo bloqueado por las operaciones de entrada/salida.
* **Tiempo de espera:** Es el período de tiempo en el que los procesos están activos pero no se ejecutan, es decir, los tiempos de espera en las diferentes colas.
* **Eficiencia:** Se refiere al uso del procesador, que debe estar ocupado todo el tiempo posible para lograr un alto rendimiento.
* **Rendimiento:** Es la cantidad de trabajos o procesos que se realizan en una unidad de tiempo, que debe ser lo más grande posible.

¿Qué algoritmo de planificación se debe utilizar para un sistema específico? El diseñador del sistema operativo tendrá la tarea de seleccionar los mecanismos adecuados para que la política seleccionada, basándose en los criterios anteriores, sea satisfactoria y brinde un alto rendimiento global.

**2. Parámetros de medida**

Para estudiar el comportamiento de las diversas políticas de planificación, definiremos dos medidas relacionadas entre sí que nos indiquen cómo estamos tratando un proceso específico.

Sea t el tiempo de servicio, que es el tiempo que un proceso P necesita estar en ejecución para completar su trabajo, ti el momento en que el usuario da la orden de ejecución del proceso y tf el momento en que el proceso termina su ejecución. En función de estos datos, evaluaremos cada proceso de la siguiente manera:

* **Tiempo de servicio (T): T = tf-t**
* **Tiempo de espera (E): E = T-t**

Podemos crear una relación a partir de los dos valores anteriores para evaluar la actuación de la política establecida en lo que se conoce como índice de servicio (I), donde:

* **I=t/T**

Este índice muestra el tiempo de ejecución del proceso en comparación con su tiempo de vida en el sistema.

Si solo se ejecuta un proceso en el sistema según el valor del índice de servicio, podemos decir que:

* Cuando **I** sea próximo a la unidad, el proceso está limitado por proceso.
* Si **I** tiene un valor bajo próximo a O, el proceso estará limitado por entrada/salida.

En situaciones típicas en las que hay más de un proceso en el sistema, es imposible realizar las consideraciones mencionadas anteriormente debido a la posibilidad de desvirtuar el comportamiento real del sistema. Por esta razón, se implementan las mismas medidas, pero se utilizan valores medios que se obtienen al tomar en cuenta todos los procesos en curso. Las medidas que se tomarán mostrarán el comportamiento real del sistema. Las medidas que mencionamos son:

* **Tiempo medio de servicio.**
* **Tiempo medio de espera.**
* **Eficiencia** (índice medio de servicio).

**3. Algoritmos de Planificación**

Ya hemos observado que la tarea del planificador del procesador es asignarle el procesador a los procesos que se encuentran en la cola de los procesos preparados. Dos fuentes diferentes de alimento para esta cola:

1. El planificador coloca un proceso en la cola de procesos preparados si el procesador está ocupado cada vez que un usuario lo ejecuta.
2. Un proceso regresa a la cola de los procesos preparados cuando deja de estar en estado de ejecución sin motivo o se bloquea

Por otro lado, el planificador **no** tiene acceso a un proceso cuando termina su ejecución.

Las políticas de planificación se dividen en **dos categorías**:

* **Políticas apropiativas:** Son los que cambian los procesos con cada cambio de contexto; por ejemplo, un proceso que usa el procesador puede ser temporalmente detenido para permitir que otro proceso lo use. Los sistemas operativos que utilizan tiempo compartido y tiempo real los usan.
* **Políticas no son apropiadas:** Son aquellas en las que un proceso permanece en el procesador desde el inicio hasta el final. Los sistemas de proceso por lotes los utilizan.

**\*\* Hay que tener en cuenta que la apropiación se ve desde el punto de vista del proceso que entra en lugar del punto de vista del procesador.\*\***

Para examinar las diversas políticas, utilizaremos la situación de un grupo de procesos que se encuentran en un sistema, cuyos datos y representación gráfica se encuentran en la siguiente tabla.



Se puede ver el instante de entrada del sistema y el tiempo de servicio de cada proceso, que es el tiempo que necesita cada proceso para ejecutarse si fuera el único proceso presente en el sistema. Por otro lado, supongamos que estos procesos no requieren operaciones de entrada/salida para facilitar el estudio, aunque esto sería una utopía.

En el ejemplo, utilizamos una unidad de tiempo abstracta, que puede ser cualquier unidad, como milisegundos, segundos o ciclos de reloj. El índice medio de servicio o eficiencia, que es un valor adimensional y suele expresarse en tanto por ciento, es el valor más representativo del comportamiento de una política de planificación.

Los algoritmos de planificación implementan una variedad de políticas de planificación:

* **FIFO**
* **ROUND ROBIN**
* **SJN**
* **SRT**

**Algoritmo 1. First In, First Out (FIFO)**

A veces conocido como FCFS (First Come, First Served). El primer proceso que llega recibe el servicio. Los procesos se encuentran en una cola de procesos preparados y se llevarán a ejecución en función de su orden de llegada. Cada proceso es ejecutado por el procesador hasta que se completa.

**Algoritmo 2. Round-Robin (RR)**

Esta política, que podría ser traducida como asignación cíclica o planificación en curso, es una mejora de la FIFO. Intenta ser más equitativo en la respuesta de los procesos cortos y largos.

Consiste en otorgar a cada proceso en ejecución un período de tiempo q (quantum), después del cual, si el proceso no ha terminado, se le devuelve al final de la cola de procesos preparados, dando al procesador su correspondiente quantum para el siguiente proceso.

**Algoritmo 3. Siguiente proceso más corto (SJN**)

El método SJN, también conocido como ShortestJob Next, es un enfoque de planificación no apropiado que busca alcanzar los mismos objetivos que el RR.

Esta política elige procesos preparados que requieren menos tiempo de ejecución. Es necesario conocer el tiempo de procesador necesario para cada proceso, lo cual es una tarea difícil pero posible a través de diversos métodos, como la información proporcionada por el propio usuario, el propio programa, basándose en la historia anterior (heurística), etc.

El tiempo de servicio T de esta política beneficia a los procesos cortos, pero perjudica a los largos.

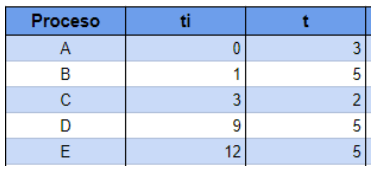
**Algoritmo 4.- Próximo proceso, el de tiempo restante más corto (SRT)**

La política SRT (Shortest Remaining Time) busca obtener las ventajas de ambos métodos. Esta técnica cambia el proceso de ejecución cuando se ejecuta un proceso, pasando del planificador de largo plazo al de corto plazo, lo que reduce el tiempo de ejecución total en comparación con el del procesador. Con respecto a SJN, el tiempo de respuesta promedio de los procesos largos mejora.

El índice de servicio (I) es excelente y el tiempo de espera (E) es bastante corto para la mayoría de los procesos. SRT hace que la lista de procesos preparados sea lo más corta posible, lo que resulta en una buena eficiencia.

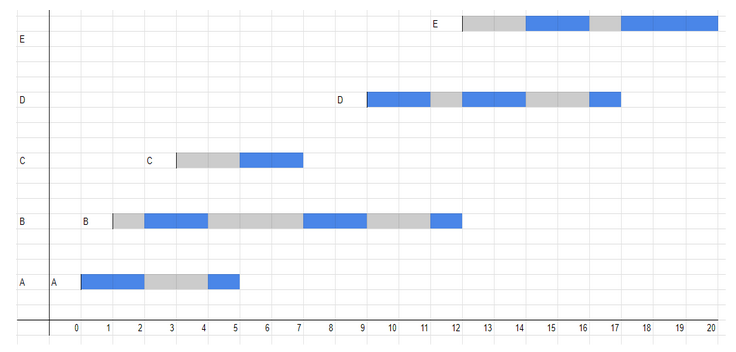
**EJERCICIOS**

1. Tenemos 5 peticiones de CPU cuyos tiempos de EJECUCIÓN (t) y tiempos de llegada (t i) se muestran en la siguiente tabla:

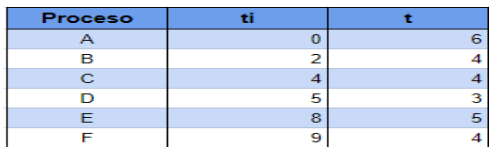


Representa gráficamente cómo el algoritmo RR (quantum = 2) atendería dichos procesos.

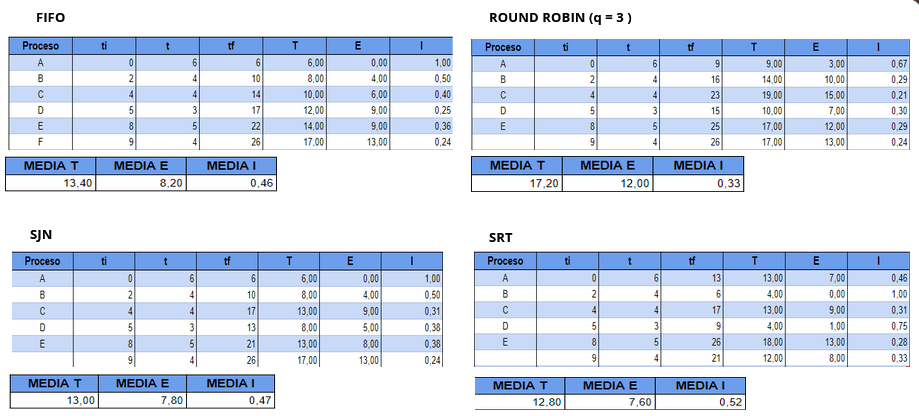
\*\* SOLUCION \*\*



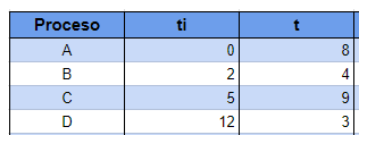
2. Atendiendo a los datos de la siguiente tabla, calcule la eficiencia de los siguientes algoritmos: FIFO, RR (quantum=3), SJN y SRT.



\*\* SOLUCION \*\*

****

3. Atendiendo a los datos de la siguiente tabla:



Representa gráficamente cómo atendería el algoritmo SRT estos procesos.

\*\* SOLUCION \*\*

