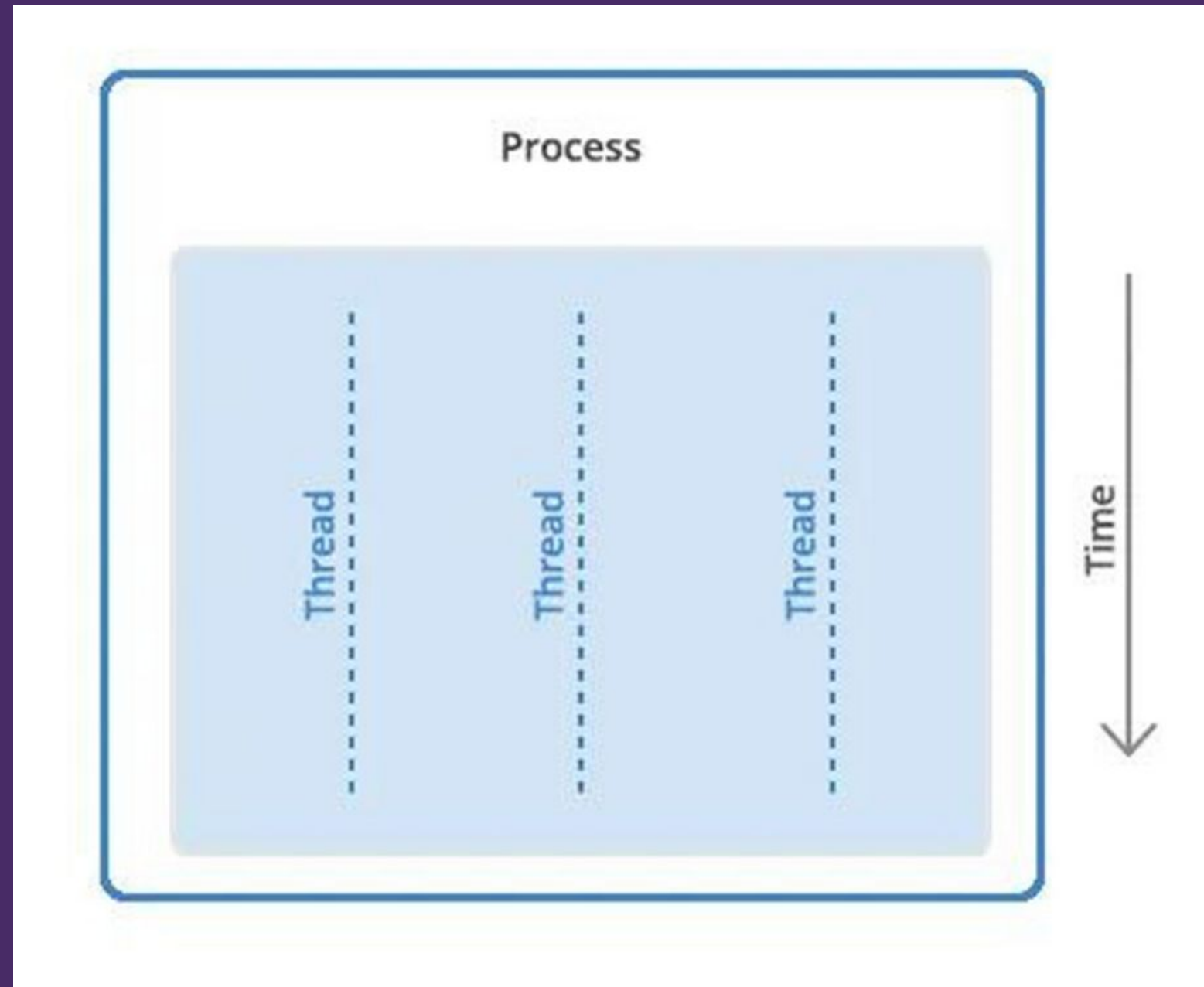




LABORATORIO SISTEMAS OPERATIVOS 2

HILOS, CONCURRENCIA Y PARALELISMO

HILOS (THREADS)



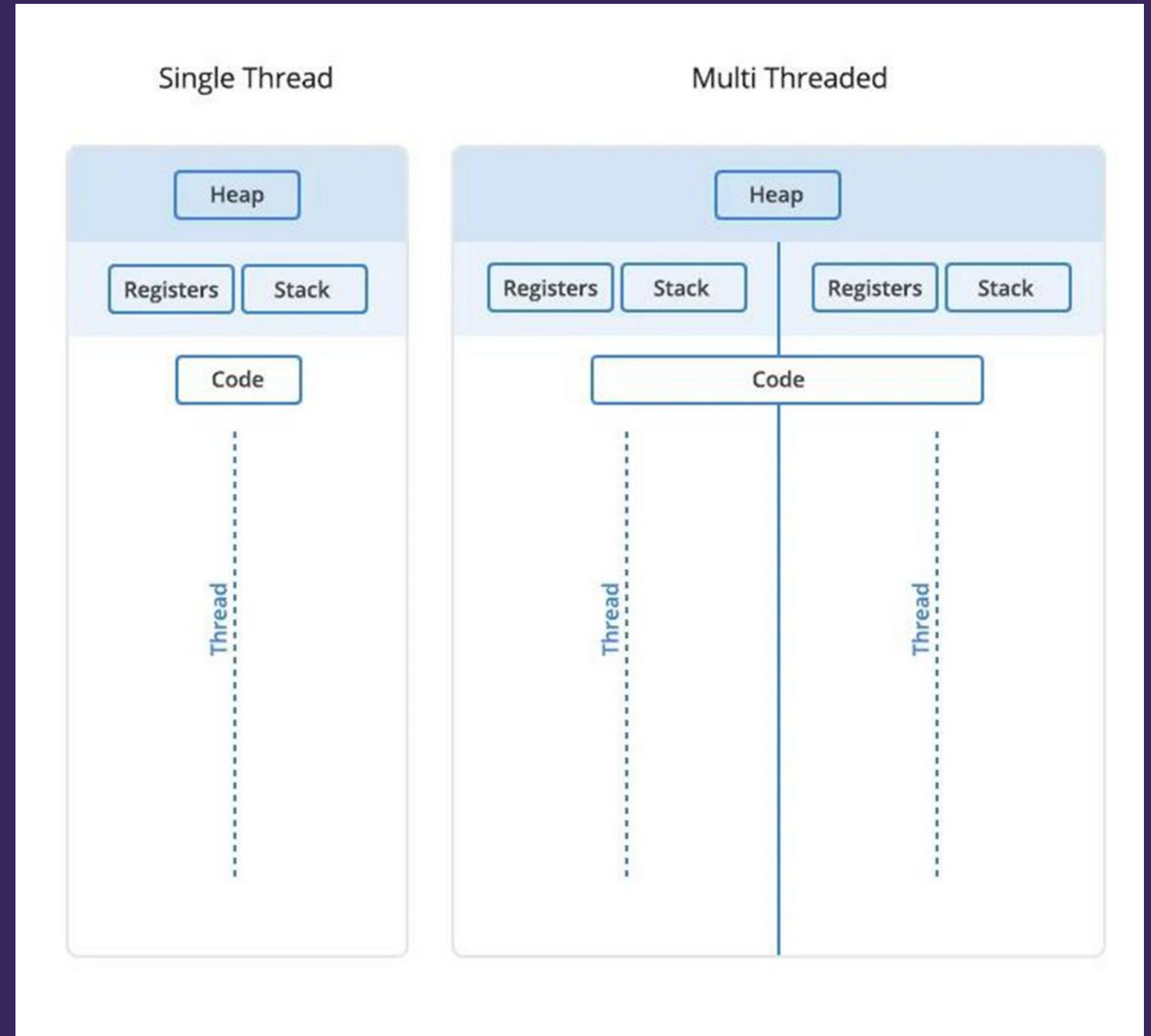
Un hilo de ejecución es la secuencia más pequeña de instrucciones que un programador puede gestionar de forma independiente.

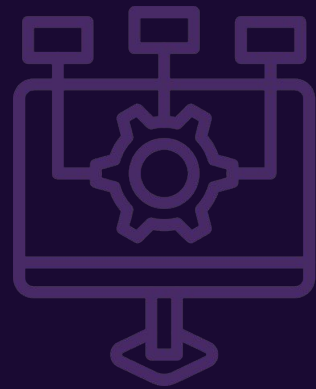
El sistema operativo crea y administra hilos, estos comparten la misma memoria y recursos que el programa que los creó, permitiendo que múltiples threads colaboren y trabajen de manera eficiente dentro de un solo programa.

MULTITHREADING

Multithreading es una técnica utilizada en los sistemas operativos para mejorar el rendimiento y la capacidad de respuesta de los sistemas informáticos al permitir que un solo proceso ejecute múltiples tareas al mismo tiempo.

El multithreading permite que varios hilos compartan los mismos recursos de un único proceso, como la CPU, la memoria y los dispositivos de E/S.

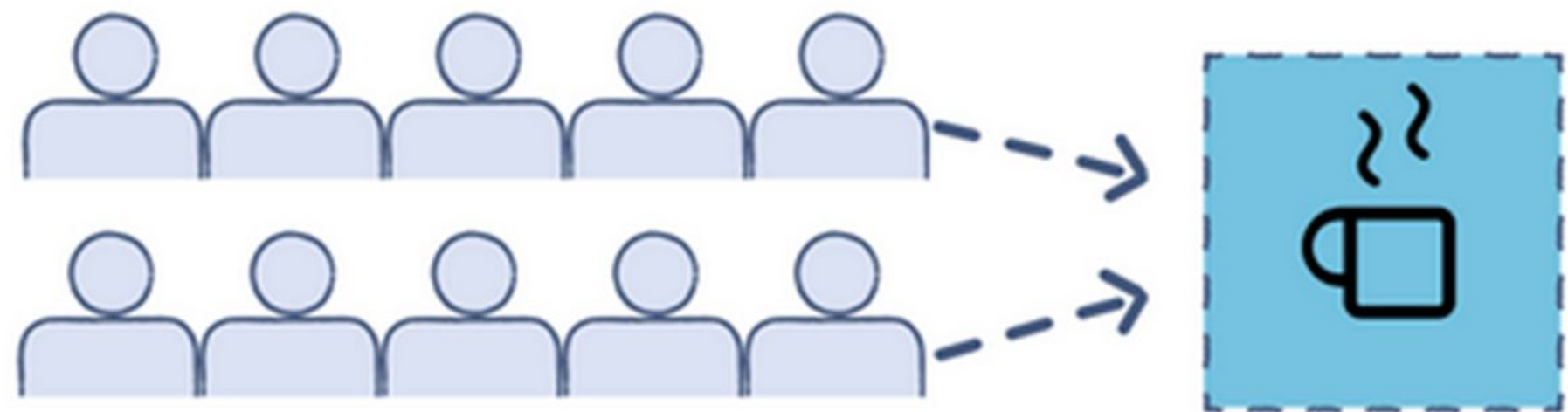




CONCURRENCIA

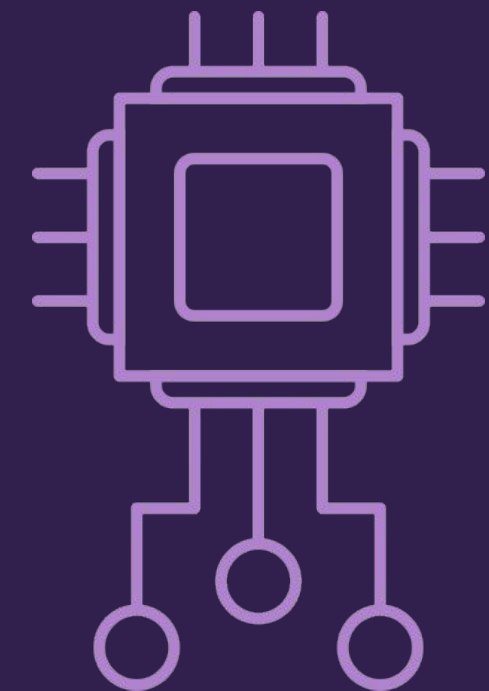
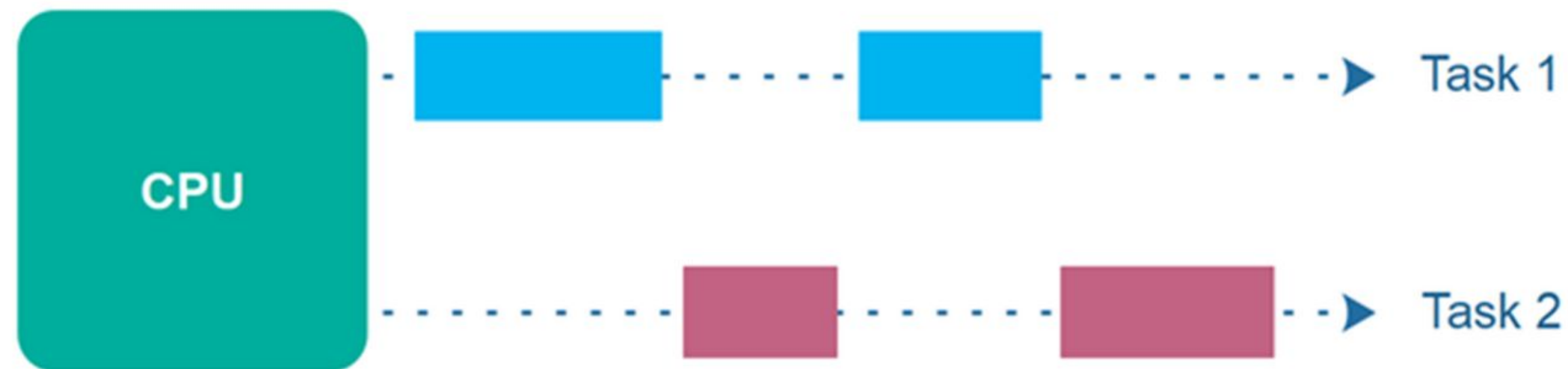
Una Unidad Central de Procesamiento (CPU, o simplemente un procesador) puede trabajar en una sola tarea a la vez. Si se le asignan varias tareas, por ejemplo, reproducir una canción y escribir código, simplemente cambia entre estas tareas. Este cambio es tan rápido y fluido que, para un usuario, se siente como una multitarea.

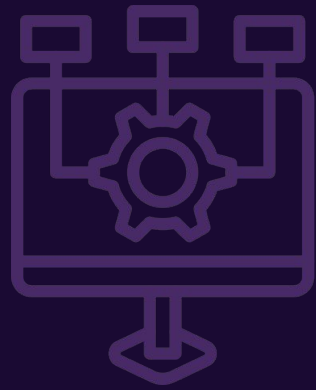
Esta capacidad de las CPU modernas para pausar y reanudar tareas tan rápido da la ilusión de que las tareas se ejecutan en paralelo. Sin embargo, esto no es paralelo. Esto es concurrente.



La concurrencia significa ejecutar múltiples tareas al mismo tiempo, pero no necesariamente simultáneamente.

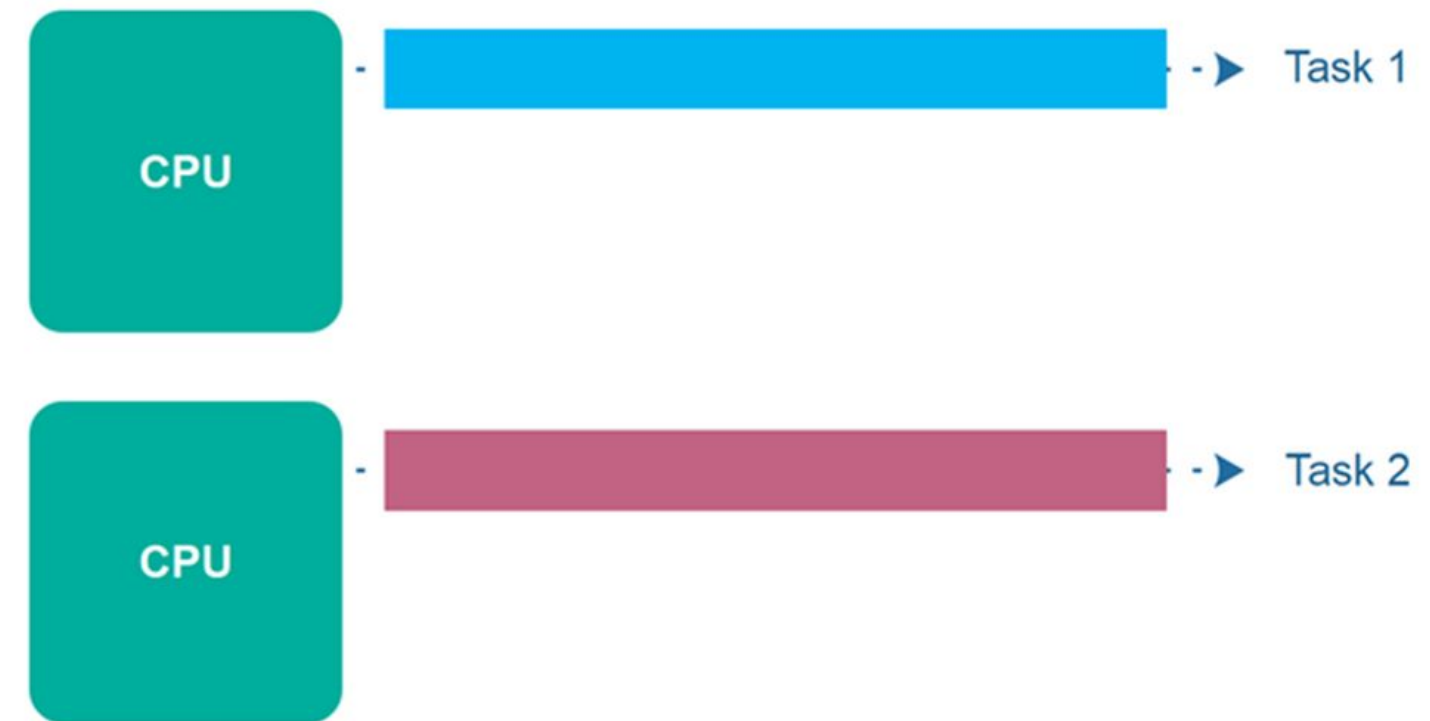
Si hay dos tareas que se ejecutarán simultáneamente en una CPU de 1 núcleo, la CPU decidirá ejecutar una tarea primero y luego la otra tarea o ejecutar la mitad de una tarea y la otra mitad, etc. De esta manera pueden comenzar dos tareas, ejecutar y completar en períodos de tiempo superpuestos

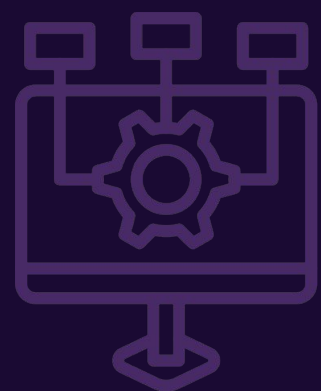




EJECUCIÓN PARALELA

La ejecución paralela es cuando una computadora tiene más de una CPU o núcleo de CPU y avanza en más de una tarea simultáneamente. Sin embargo, la ejecución paralela no se refiere al mismo fenómeno que el paralelismo.

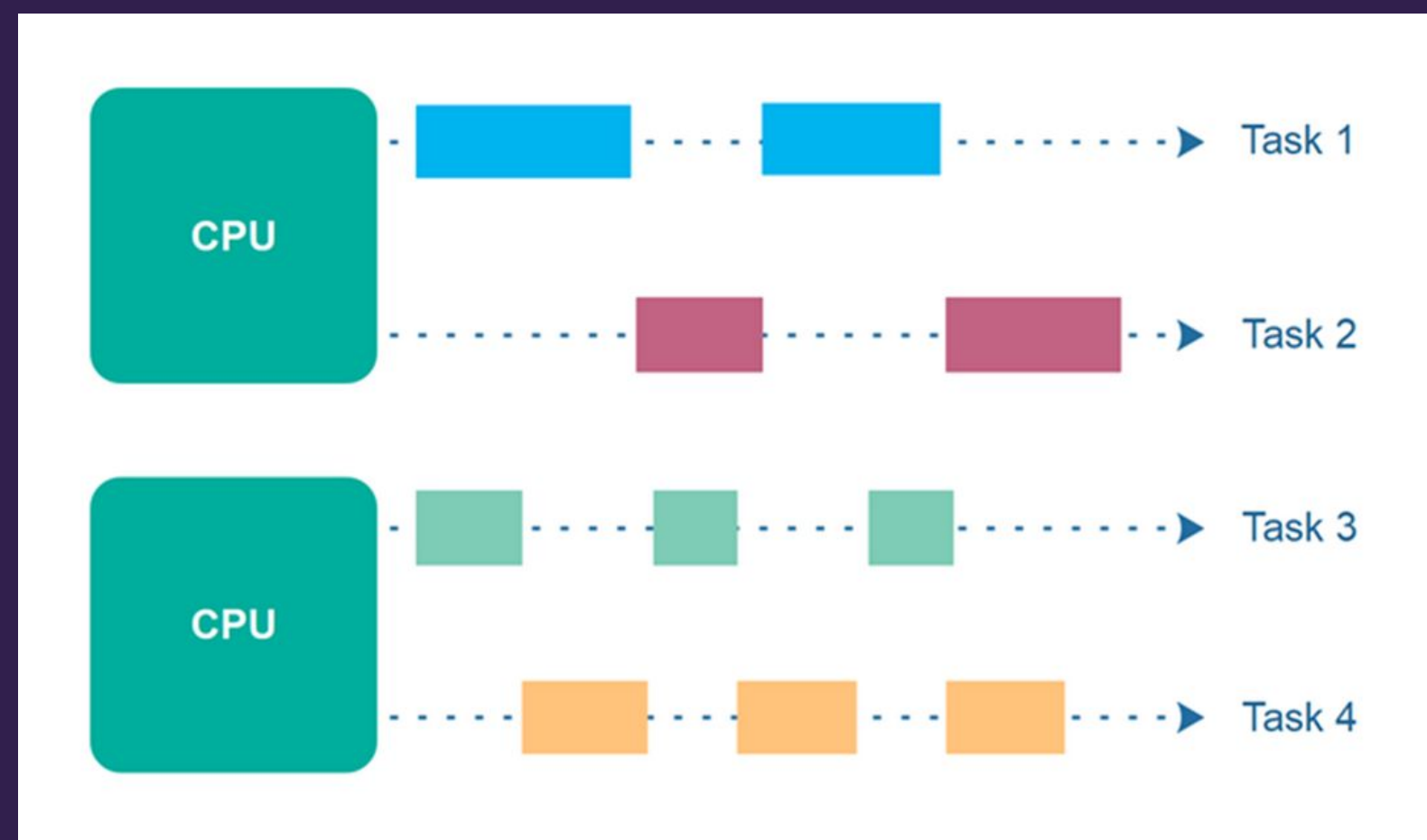


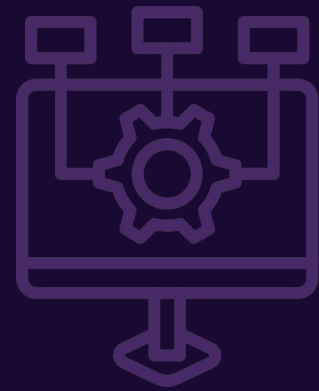


EJECUCIÓN CONCURRENTE PARALELA

Es posible tener una ejecución concurrente paralela, donde los subprocesos se distribuyen entre varias CPU.

Por lo tanto, los subprocesos ejecutados en una misma CPU se ejecutan concurrentemente, mientras que los subprocesos ejecutados en diferentes CPU se ejecutan en paralelo.

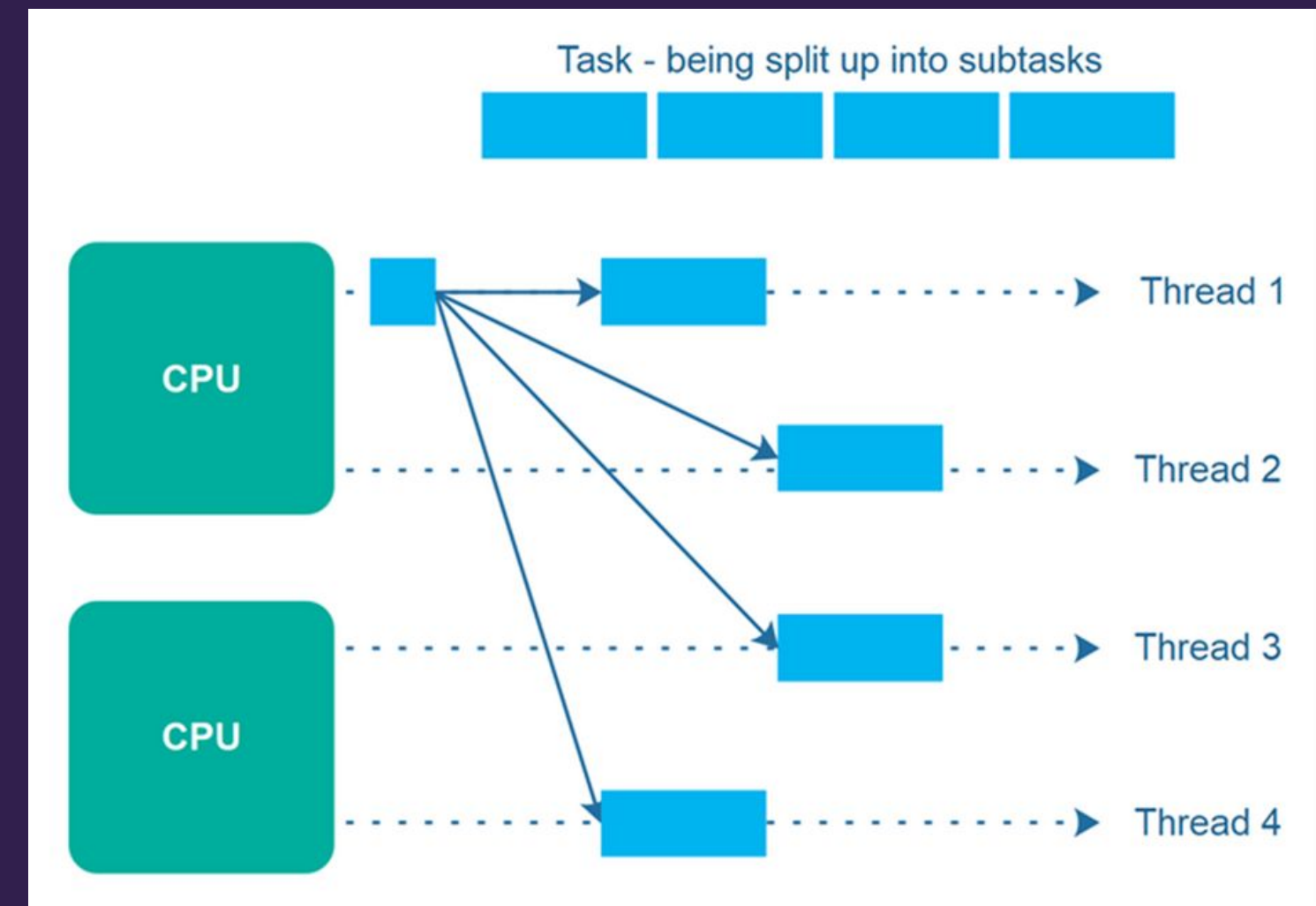




PARALELISMO

El paralelismo significa que una aplicación divide sus tareas en subtareas más pequeñas que pueden procesarse en paralelo, por ejemplo, en varias CPU al mismo tiempo.

El paralelismo no requiere que existan dos tareas, es capaz de ejecutar físicamente partes de tareas o múltiples tareas al mismo tiempo utilizando la infraestructura multinúcleo del CPU, asignando un núcleo a cada tarea o subtarea.

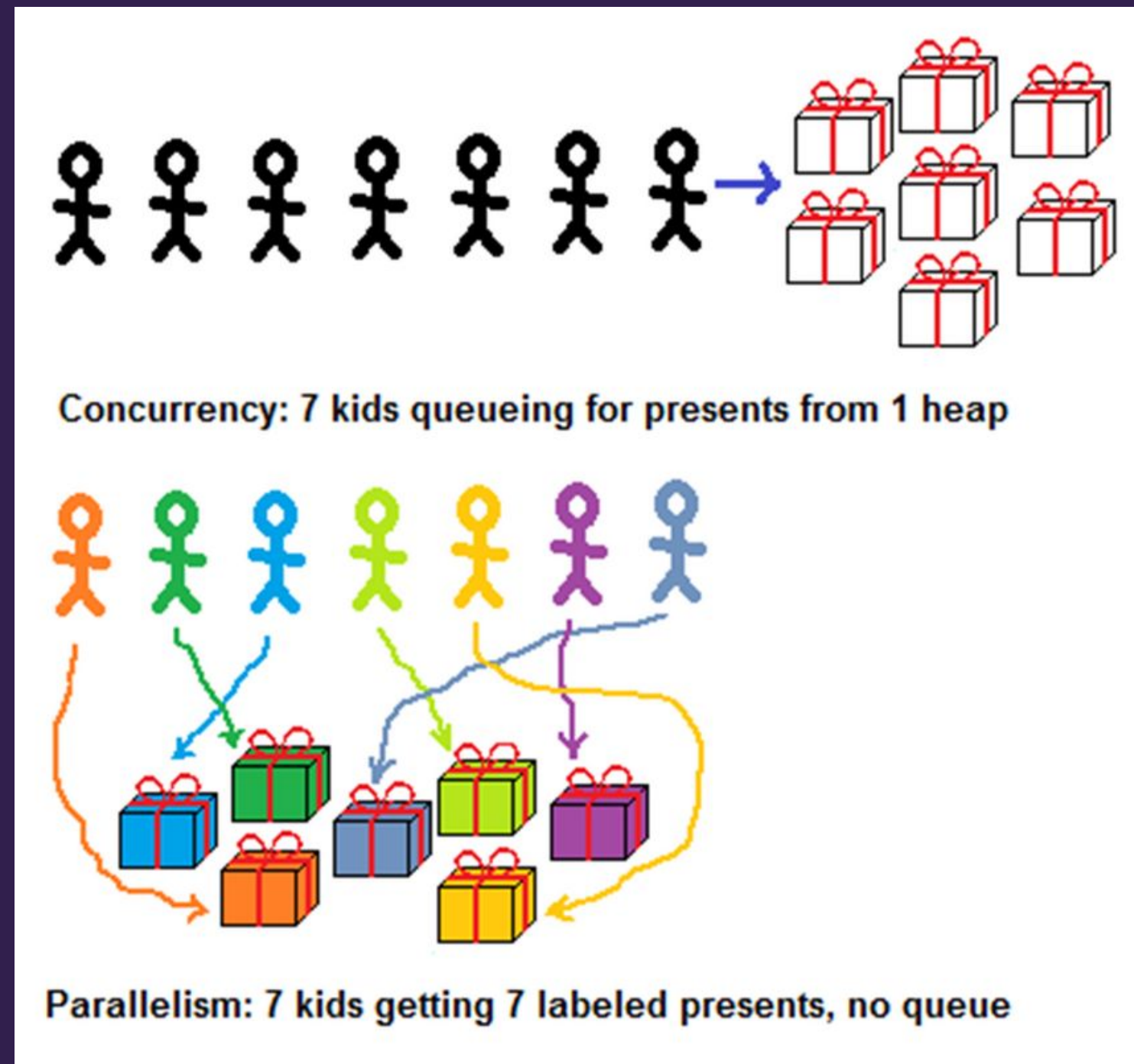


En su forma más básica, hay dos marcas que se pueden codificar en una función. La primera es paralela, lo que sugiere al compilador que la función se complete al mismo tiempo que otras funciones paralelas. El otro es secuencial, lo que significa que la función debe completarse individualmente (de forma concurrente).

Las funciones paralelas pueden acelerar significativamente las operaciones porque utilizan automáticamente más recursos de la CPU de la computadora.

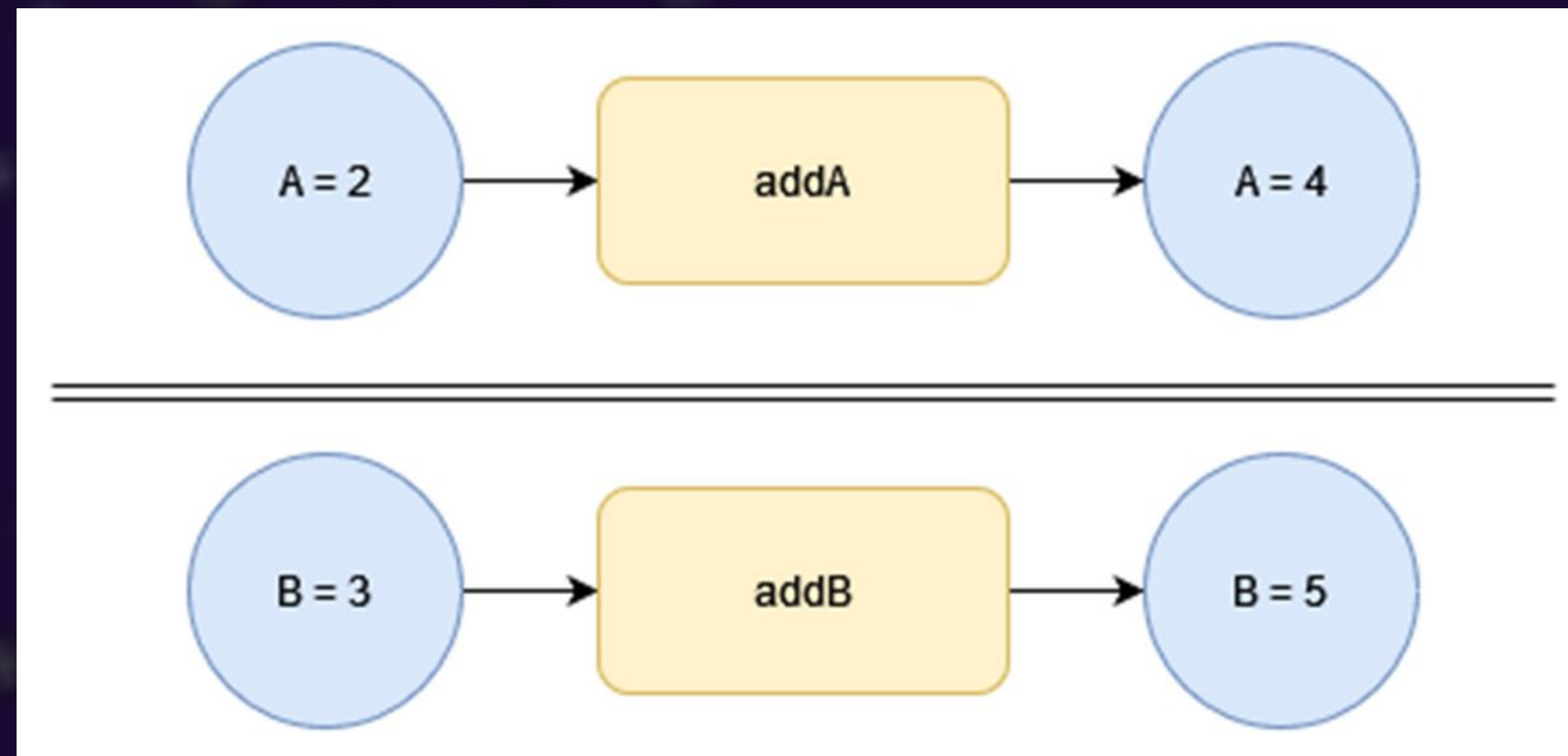
Video:

<https://youtu.be/IGY4hXyU5tc>

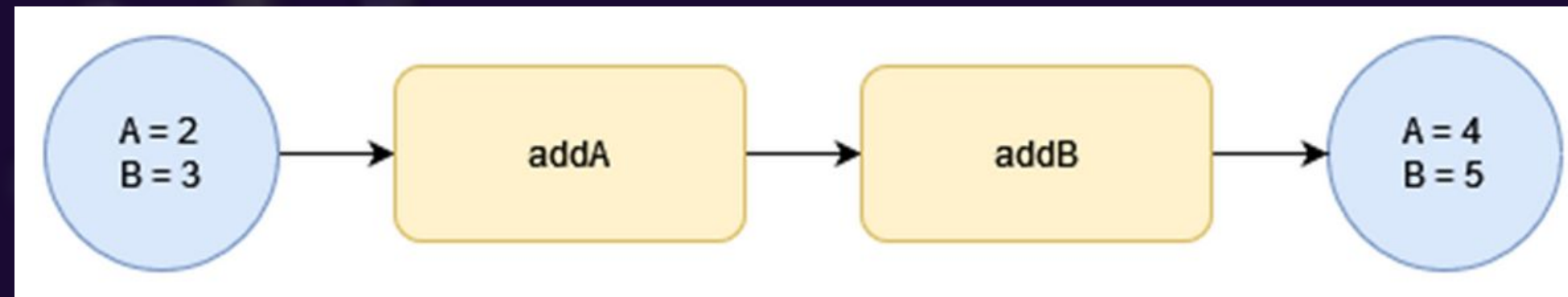


Sin embargo, es mejor guardar el paralelismo para funciones que tienen poca interacción con otras funciones mediante dependencias o edición de datos. Esto se debe a que, si bien se trabaja en ellos simultáneamente, no hay forma de saber cuál se completará primero, lo que significa que el resultado es impredecible a menos que se utilice sincronización.

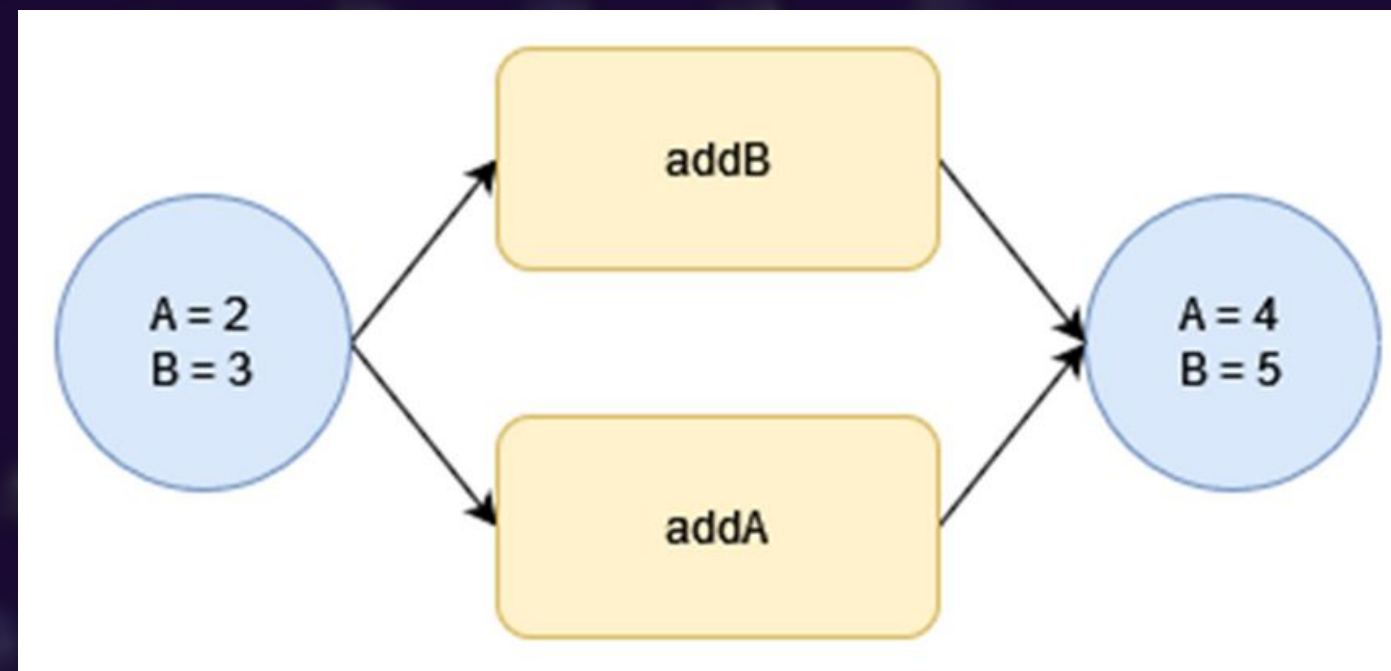
Supongamos que tenemos un programa que tiene dos variables, A y B, así como dos funciones que se encargan de sumarle 2 a estos valores, addA y addB



Al comportamiento de addA es independiente del comportamiento de addB y, por lo tanto, no tiene problemas para ejecutarse de manera concurrente



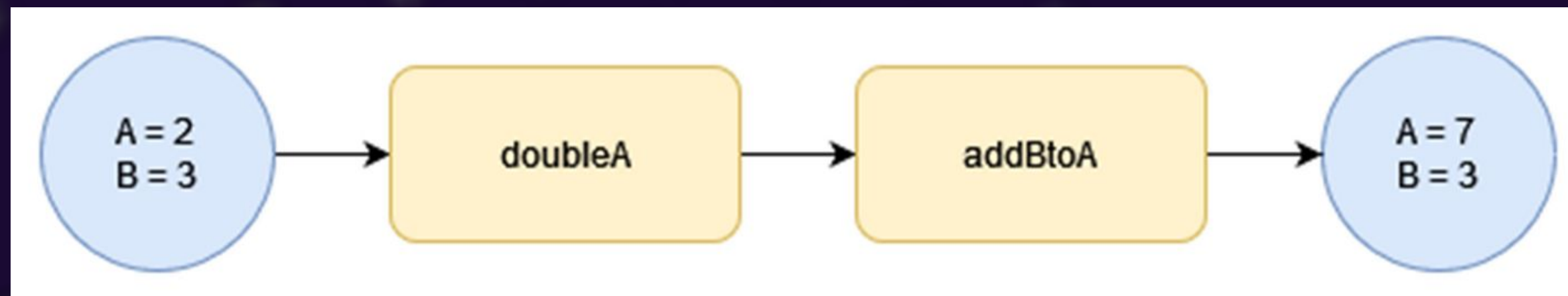
Ni ocurren problemas al ejecutarse de manera paralela.



Sin embargo, si ambas funciones afectaran a la misma variable, deberíamos preferir utilizar una ejecución concurrente.

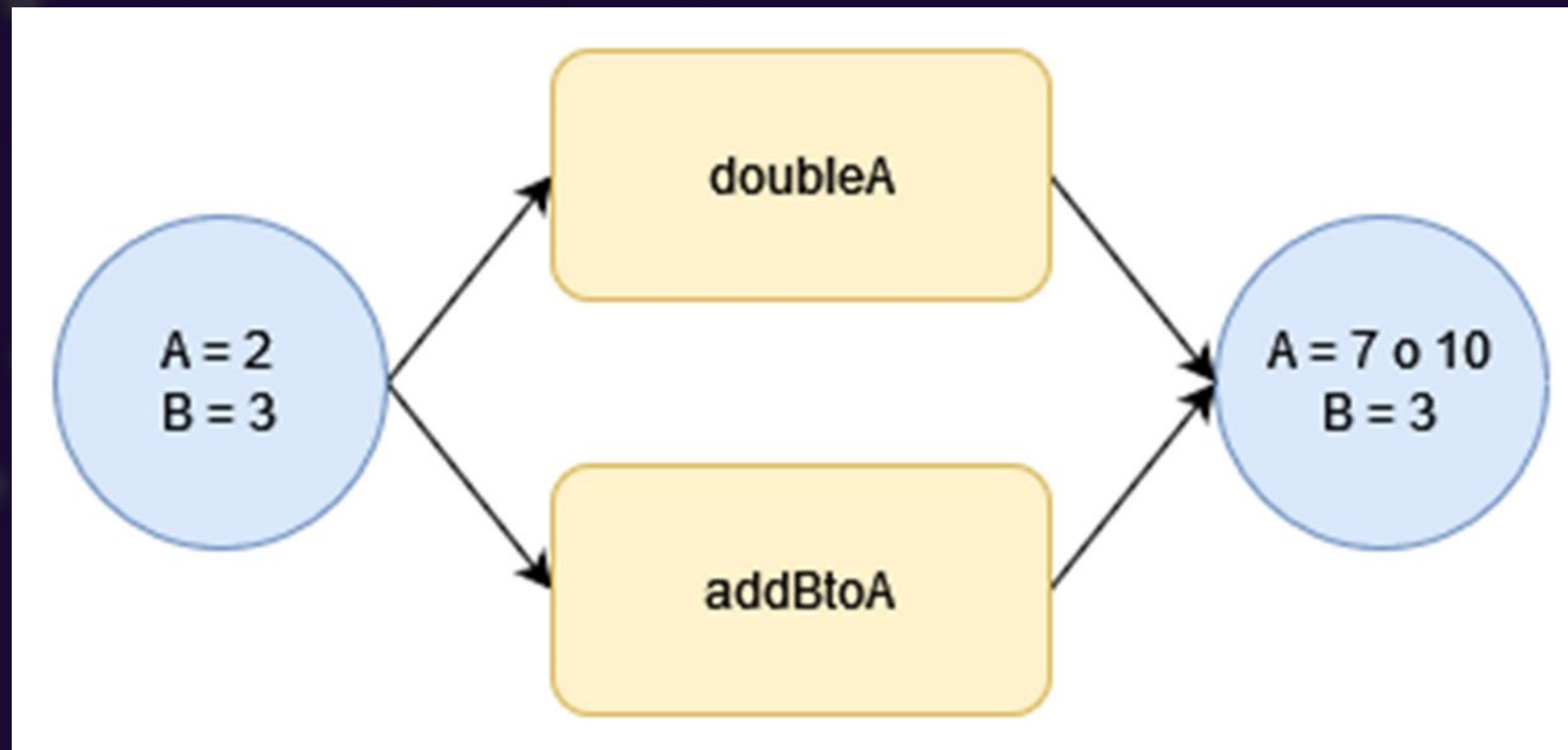
Cambiemos las funciones, donde la función doubleA duplicara el valor de la variable A y la funcion addBtoA le sumará el valor de B a A.

Ejecutando este programa de manera concurrente podemos decidir cual de estas se realizará primero por lo tanto podemos estar seguros del resultado.



En este caso, no querríamos utilizar la ejecución paralela ya que el resultado de este conjunto de funciones depende de cuál se completa primero, por lo que el programa en diferentes ejecuciones podría mostrar un resultado diferente.

Esta incertidumbre es algo que tenemos que evitar siempre cuando estemos diseñando sistemas o aplicaciones.



The background is a dark purple gradient. On the left and right sides, there are stylized, 3D-rendered brains in a lighter purple color. In the center, there is a faint, glowing wireframe mesh that resembles a topographical map or a network structure.

¡GRACIAS POR LA ATENCIÓN!

¿Dudas?