

Programación paralela

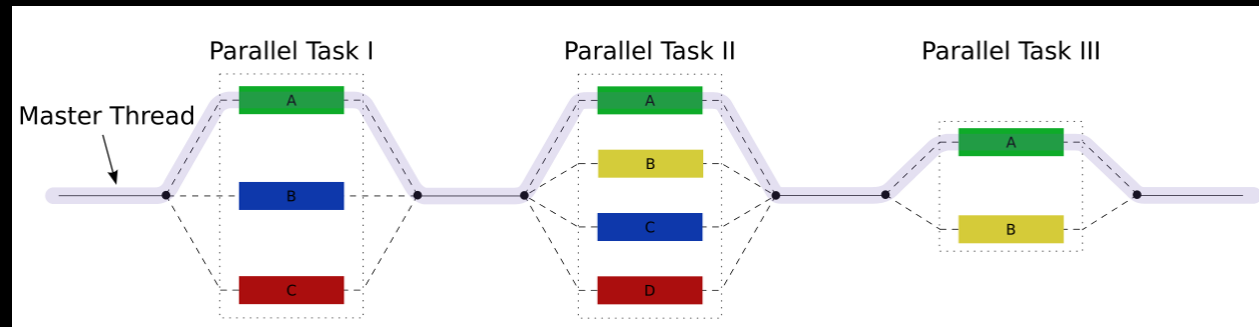
ARQUITECTURA DE COMPUTADORAS

¿Qué es el paralelismo en Programación?

Paralelismo es la ejecución simultánea de dos o más tareas.

Se considera una propiedad del hardware, ya que requiere recursos físicos para ejecutar cada tarea simultáneamente,

y su objetivo se basa en realizar una tarea en el menor tiempo posible.



Recordando nuevamente...

*El **paralelismo**: acelera la ejecución de una tarea dividiéndola en computaciones independientes y ejecutándola sobre hardware capaz de realizar computaciones simultáneas, como por ejemplo un procesador con varios núcleos.*

*Pero cuando ejecutamos una tarea paralelizada en múltiples ordenadores, en vez de en los múltiples cores de un solo ordenador, decimos que la computación paralela es **distribuida**.*

“Ejemplo, cada búsqueda en Google se ejecuta simultáneamente en cientos de ordenadores, cada uno de los cuales busca al mismo tiempo en un subconjunto del índice del web.”

Ley de Wirth

“El Software se expande para ocupar todo el espacio disponible”

“El software se ralentiza más deprisa de lo que se acelera el hardware.”

Ley de Wirth

El hardware se está volviendo, claramente, más rápido a medida que pasa el tiempo y parte de ese desarrollo está cuantificado por la **Ley de Moore**.

Los programas tienden a hacerse más grandes y complicados con el paso del tiempo y a veces los programadores se refieren a la Ley de Moore para justificar la escritura de código lento o no optimizado,

pensando que no será un problema porque el hardware sobre el que correrá el programa será cada vez más rápido.



Ley de Wirth

Un ejemplo de la Ley de Wirth que se puede observar es:

- Que el tiempo que le toma a un PC actual arrancar su sistema operativo no es menor al que le tomaría a un PC de hace cinco o diez años con un sistema operativo de la época.

“Porque los programadores de aplicaciones, en la mayoría de los casos, no sabemos emplear la potencia del hardware y no se optimiza”



Multitarea

UNIVERSIDAD DE SONSONATE



Multitarea

Hacer varias cosas de manera simultánea, que gran regalo y que lio

No, solo hay que entender en que se basa y como se programa.

¿Para qué queremos la multitarea?

Una respuesta rápida sería:

- Para que el usuario no se enfade con nosotros, por esperar demasiado, en momentos en los que no hace falta
- Notar que nos referimos la mayoría de las veces del bloqueo al usuario, no a la ejecución que se produce debajo de esa pantalla de carga.

¿Para qué queremos la multitarea?

Otro efecto molesto es aquel que se da cuando el usuario pulsa un botón y la aplicación se bloquea, pero no se bloquea por un error, sino porque debajo se está ejecutando un montón de código que requiere tiempo de proceso para concluirse.

Todo esto se soluciona con la multitarea, pero puede que antes haya que pensar un poco para en la solución con ésta.

¿Para qué queremos la multitarea?

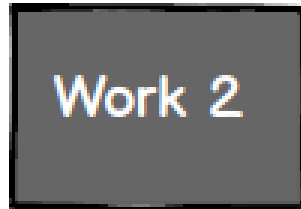
Con la multitarea el usuario nunca quedará bloqueado –Logrando de esta manera seguir usando la aplicación mientras algo muy gordo se ejecuta debajo;

se eliminan la mayoría de pantallas de carga o no interrumpirán la experiencia de uso de la aplicación; y se ejecutará de manera más óptima, haciendo que el procesador no esté esperando continuamente y con cuellos de botella por llegarle un montón de cosas a la vez.

Work1 : InProcess



Work2 : In Process



Threads

ARQUITECTURA DE
COMPUTADORAS

Asynchronous way

Hilos

Un Hilo es un trozo de código de nuestro programa que puede ser ejecutado al mismo tiempo que otro.

¿Qué puede ejecutarse de manera simultánea a otro?

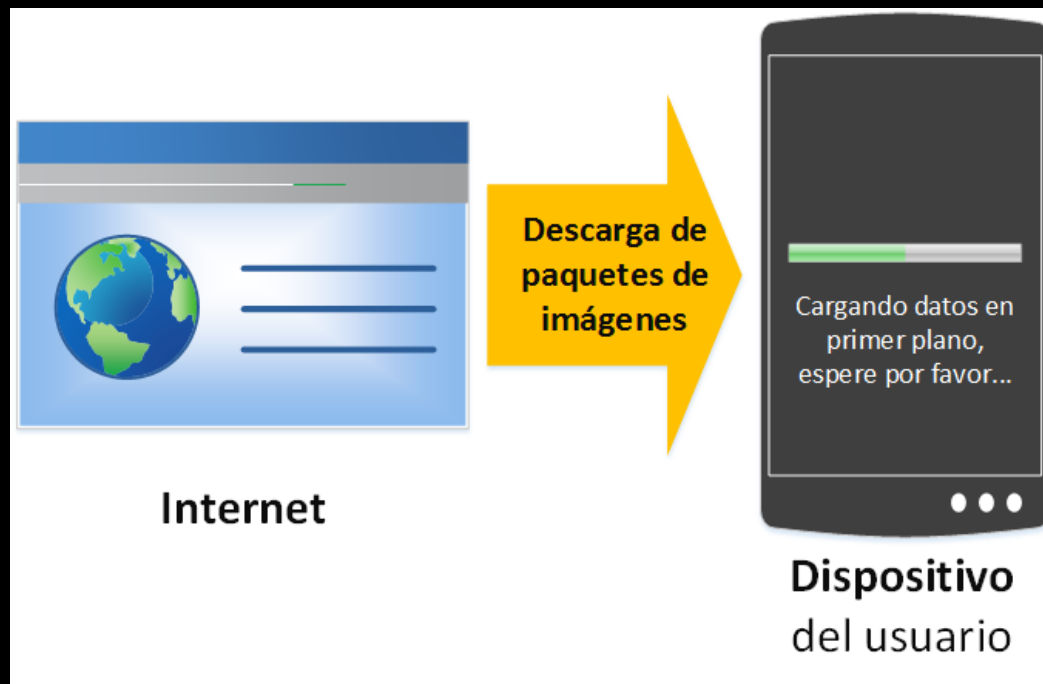
por ejemplo, queremos ver un listado de 100 imágenes que se descargan desde Internet,

como usuario

¿Cuál de las dos opciones siguientes elegirías?

Hilos [Escenario 1]

Descargar las imágenes 100 imágenes, haciendo esperar al usuario con una pantalla de “cargando” hasta que se descargan todas. Luego podrá ver el listado con las imágenes.



Hilos [Escenario 2]

Que mientras se descargan las 100 imágenes, el usuario pueda ir viendo y usando las que ya se han descargado.



Hilos

Como desarrollador la opción A

“Es más sencilla que un usuario preferiría. “

La B es lo que todo usuario quiere de una aplicación:

“Tener que esperar no es una opción. “

Volviendo al punto de vista del desarrollador, tampoco es una opción.

Un buen desarrollador de aplicaciones hace bien las cosas y se inclina por la opción B

la opción A no existe, la opción A nos lleva de cabeza a la “ley de Wirth” antes descrita- queremos ser profesionales y la opción B es la opción a aplicar.

Hilos

Desde el punto de vista del usuario existen dos áreas bien diferenciadas, que el desarrollador ha de tener en cuenta:

- **Primer plano:** Aquí se ejecuta únicamente un hilo llamado “hilo principal”.
- **Segundo plano (o en inglés background):** Se ejecuta todo el resto de hilos.

Ejemplo

ARQUITECTURA DE COMPUTADORAS