Hilos permiten que una tarea se ejecute en simultáneo con otra. Proceso, la ejecución del código e interacción con los inputs, outputs y la memoria. Cuando un hilo modifica un dato de memoria, los otros hilos también pueden acceder a ese dato. Cuando termina el hilo se liberan todos sus recursos. Algunos lenguajes como Java o Delphi pueden manejar hilos, otros como C o C++ requieren de librerías especiales que dependen del sistema operativo. Los hilos comparten los recursos, las direcciones de memoria y los datos. Hay dos tipos de hilos, los ULT (*user level thread*) y KLT (*kernel level thread*).

Los hilos surgieron como una forma de simplificar, de ayudar a programar códigos complejos, no tanto para acelerar el proceso. Los procesos están compuestos de múltiples hilos. Un procesador de ocho núcleos puede realizar ocho tareas “al mismo tiempo”, ignorando las llamadas al sistema operativo, lectura de periféricos, etc.

Permiten separar las tareas complejas en algunas más simples, aprovechando los recursos del equipo, además de que si un proceso falla no falla toda la aplicación. Lo más importante, **hay veces que tenemos tareas que no sabemos cuánto van a tardar, ni si van a culminar con éxito.** Si la tarea está pendiente no podríamos avanzar. Podemos utilizar hilos para realizar varias tareas “simultáneamente” sin tener que esperar a que una de ellas termine para poder continuar, además de poder separar los posibles errores y controlarlos en lugar de detener toda la aplicación.

Cuando a un hilo se le invoca el método start(), el sistema operativo le asigna un porcentaje de los recursos y ejecuta el método run().

Información sobre el ciclo de un proceso <https://facturapp.weebly.com/hilos.html>

En un equipo con varias CPUs, si solo hay un hilo este proceso solo puede ser ejecutando por una CPU, desperdiciando el potencial de tener varias, sin embargo, cuando se utilizan múltiples hilos cada uno puede ser ejecutado por una CPU distinta, aprovechando así los recursos del equipo.

En los UTL no se relaciona con el SO, por lo cual solo se reconoce como un único hilo.

Investigar sobre el espacio de direcciones de usuario y además el cambio de contexto.

Un microprocesador de un núcleo sólo puede ejecutar una instrucción por ciclo.

Un proceso es la forma en que el SO ejecuta una parte de algún programa, por lo cual la ejecución de un programa se lleva a cabo mediante varios procesos, los cuales, están compuestos a su vez de lo que se conoce como hilos; en estos últimos puede haber concurrencia para ejecutar varios conjuntos de instrucciones “simultáneamente”. Cada hilo de procesamiento ejecuta un fragmento de toda la tarea a realizar. Así los recursos del sistema pueden ser aprovechados más eficientemente. Si un programa no es multi-hilo, este solo tendrá un único hilo asignado por el SO.

En un SO multi-hilos hay bloques de control y pilas separadas para cada hilo. Un hilo es una unidad básica de utilización de la CPU; consiste de un contador de programa, varios registros y un espacio de pila. Los hilos de un mismo proceso comparten sección de código, sección de datos, recursos del SO archivos, señales, etc. Comparte también código, variables globales y archivos abiertos. Etapas de los hilos: creación, bloqueo, desbloqueo terminación. Un hilo puede leer o escribir sobre la pila de otro hilo del mismo proceso.

Si un hilo hace un bloqueo al sistema operativo (e.g. I/O), todos su hilos aledaños también se bloquean en ULT.

Paralelismo vs Concurrencia <https://www.researchgate.net/post/Concurrency_Vs_Parallelism>

Paralelismo se refiere a hacer dos tareas al mismo tiempo, debido al uso de dos núcleos del procesador, paralelismo en un procesador de un solo núcleo es simplemente concurrencia, la cual consiste en realizar instrucciones de una tarea y de la otra alternativamente.