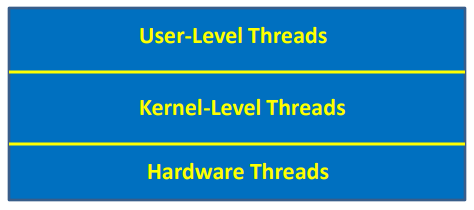
|  |  |  |
| --- | --- | --- |
| **Fecha de entrega:** 08 de diciembre de 2017 | **Título de la tarea: Thread a bajo nivel.** | **Número de la tarea:** 3.2 |
| **Grupo:** 3CM3 | **Alumno:** Estrada Granados Diego | **Unidad de aprendizaje:** Arquitectura de Computadoras |

**Atributos del concepto Thread a bajo nivel.**

**Desarrollo:**

El concepto de Thread se puede expresar de diferentes maneras en función al nivel en que nos encontremos desarrollando software, tomando como referencia que mientras a más bajo nivel describamos este concepto mayor serán los elementos necesarios para obtener una representación de proceso ligero a bajo nivel. Comencemos por definir a un hilo, proceso ligero o Thread a alto nivel como una secuencia de tareas desprendidas de un proceso que pueden ser ejecutadas por el sistema operativo. Este concepto solo busca ejemplificar que un hilo no se trata de un proceso sino de un subproceso que dé él se deriva, posteriormente se mencionaran los elementos que comparten y los que lo hacen diferente a un proceso.

Teniendo en claro el concepto de Thread a alto nivel agreguemos el principio que un Thread es una secuencia de instrucciones ejecutadas de manera independiente por otro tipo de secuencias de instrucciones, además un hilo del hecho que un hilo pueda crear otro hilo y que cada hilo lo mantiene en la máquina de estados actual. En la siguiente imagen se observan los diferentes niveles de hilos que se pueden manejar. (Valvano, 2016)



Cada uno de estos niveles se relaciona con su antecesor o sucesor de la siguiente manera.

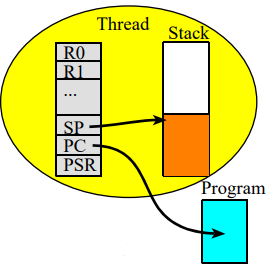
1 Hilo a nivel de usuario es mapeado a un solo hilo a nivel de kernel como ejemplo tenemos los hilos que los sistemas operativos crean para su uso privado (los usuarios no pueden acceder ni a los procesos ni a los hilos)

Muchos hilos a nivel de kernel se relacionan con un solo hilo a nivel de usuario, esto se representa cuando hacemos uso de librerías en algún lenguaje de alto nivel y creamos uno o n cantidad de hilos en el proceso que se ejecuta nuestro programa.

Cada hilo comparte la misma dirección de memoria que el proceso que lo desencadenó, pero cuentan con su propia pila de instrucciones. (Zahra, 2012)

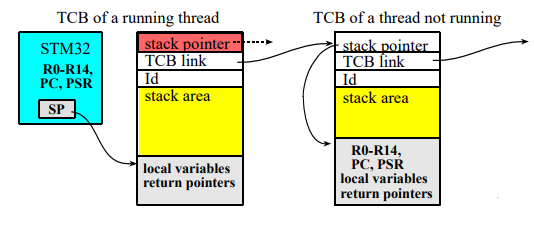
A bajo nivel los procesos ligeros tienen las siguientes características.

* Ejecución de una tarea de software.
* Registros propios.
* Pila propia.
* Las variables declaradas dentro del hilo tienen un ámbio privado.
* Los hilos cooperan entre sí por lograr una meta.
* Las variables globales son administradas por el sistema operativo y alojadas en el Thread Control Block (TCB). (Valvano, 2016)



El Thread Block Control consta de los siguientes elementos.

* Apunador a la pila del hilo.
* Siguiente / Anterior link.
* Identificador del hilo.
* Contador de sleep
* Prioridad (Valvano, 2016)



Finalmente, para complementar los atributos que posee un hilo es necesario mencionar que cada uno de ellos cuenta con un estado bien definido a partir del diagrama de estados, estos pueden ser (listo, en ejecución, bloqueado, dormido o finalizado). (Cornell Computer Science, 2017)

# Trabajos citados

Cornell Computer Science. (29 de agosto de 2017). Obtenido de http://www.cs.cornell.edu/courses/cs4410/2017fa/schedule/slides/03-processes-threads.pdf

Valvano, J. W. (abril de 2016). *The University of Texas*. Obtenido de http://users.ece.utexas.edu/~valvano/EE345M/view05\_threads.pdf

Zahra, M. (marzo de 2012). *New York University* . Obtenido de https://cs.nyu.edu/courses/fall12/CSCI-GA.3033-012/lecture6.pdf