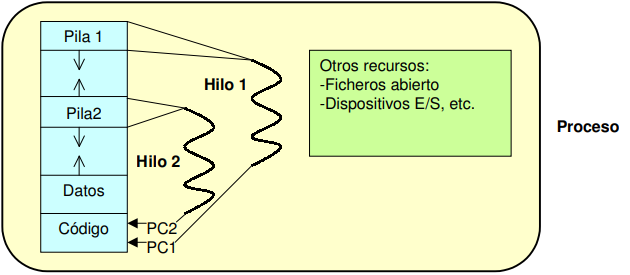
¿Qué es un hilo de ejecución?

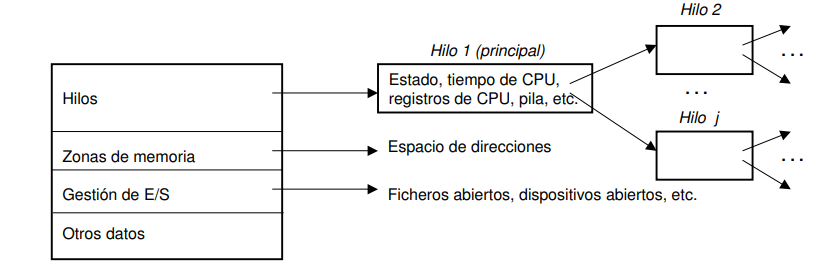
* También llamado hebra, proceso ligero, flujo, subproceso o “thread”.
* Programa en ejecución que comparte la imagen de memoria y otros recursos del proceso con otros hilos.
* Desde el punto de vista de programación: Función cuya ejecución se puede lanzar en paralelo con otras.
* Un proceso puede contener uno o más hilos.



*Hilo: unidad de planificación*

*Proceso: unidad de asignación de recursos*

|  |
| --- |
| **ESTRUCTURA DE UN PROCESO Y DE UN HILO** |

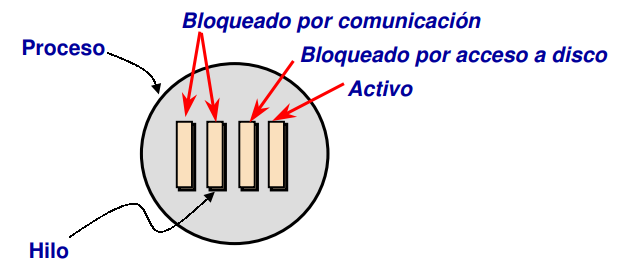


|  |
| --- |
| **CARACTERISTICAS DE UN HILO** |

* Todos los hilos de un proceso comparten el mismo entorno de ejecución (variables globales, espacio de direcciones, ficheros abiertos, etc.).
* Cada hilo tiene su propio juego de registros de CPU, pila, variables locales, etc.
* No existe protección entre hilos: un error en un hilo puede estropear la pila de otro.
* Para ordenar la forma en la que los hilos acceden a datos comunes hay que emplear mecanismos de sincronización.

|  |
| --- |
| **ESTADOS DE UN HILO** |

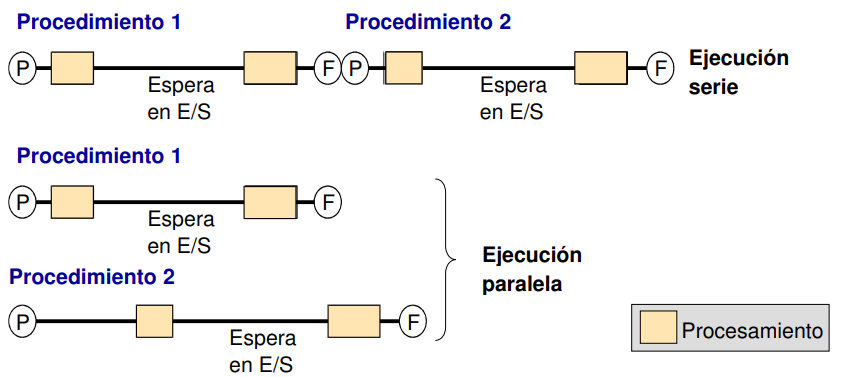
* Si hay un hilo en ejecución → Proceso en ejecución (EXEC)
* Si no hay hilos en ejecución pero sí preparados → Proceso preparado (READY)
* Si todos sus hilos bloqueados → Proceso bloqueado (WAIT)



|  |
| --- |
| **LOS HILOS PERMITEN PARALELIZAR LA EJECUCIÓN DE UNA APLICACIÓN** |

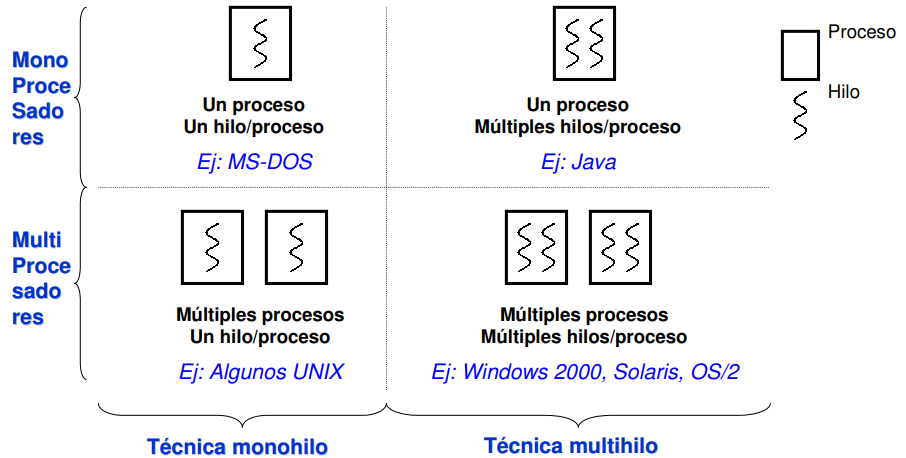
Mientras un hilo está bloqueado, otro podría ejecutarse

* Uso de llamadas al sistema bloqueantes por hilo



|  |
| --- |
| **CONCEPTO MULTIHILO** |

Capacidad de un SO para mantener varios hilos de ejecución dentro de un mismo proceso



|  |
| --- |
| **IMPLEMENTACION DE HILOS** |

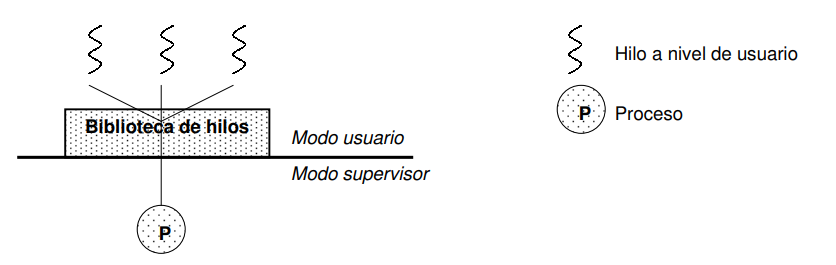
Existen dos categorías:

* ***Hilos a nivel de usuario*** , *(no son vistos por el sistemas operativo, son programables a traves de bibliotecas de hilos <ULT, User Level Thread>).*
* ***Hilos a nivel de núcleo o Kernel***, *(son vistos por el sistema operativo por ende son planificable <KLT Kernel Level Thread>).*

|  |
| --- |
| **HILO USUARIO** |

Gestión de hilos realizada por la aplicación a nivel de usuario mediante una biblioteca de hilos.

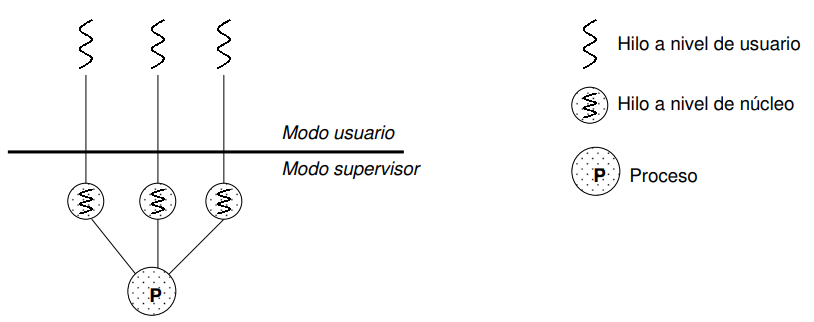
* El núcleo no es consciente de la existencia de hilos.
* Es posible programar cualquier aplicación como multihilo.



|  |  |
| --- | --- |
| **Ventajas frente a hilos a nivel de núcleo** | **Desventajas frente a hilos a nivel de núcleo** |
| No necesario acceder al núcleo para intercambio de hilos | Una llamada al sistema realizada por un hilo bloquea a todos los hilos del proceso |
| Algoritmos de planificación ad-hoc para cada aplicación | Una aplicación multihilo no puede aprovechar las ventajas del multiprocesamiento |
| Pueden ejecutarse en cualquier SO |  |

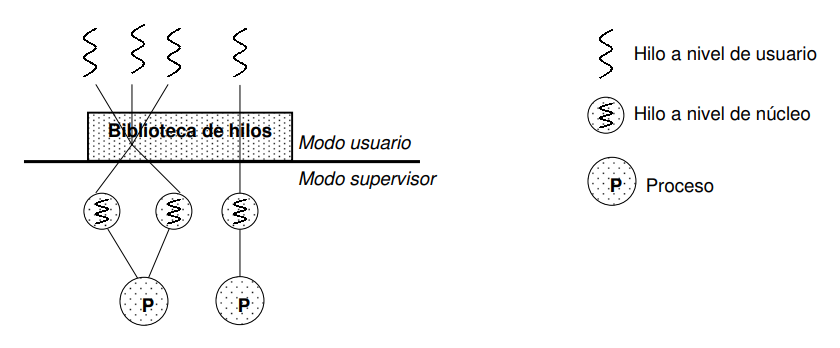
|  |
| --- |
| **HILO NUCLEO o KERNEL** |

Es posible programar cualquier aplicación como multihilo

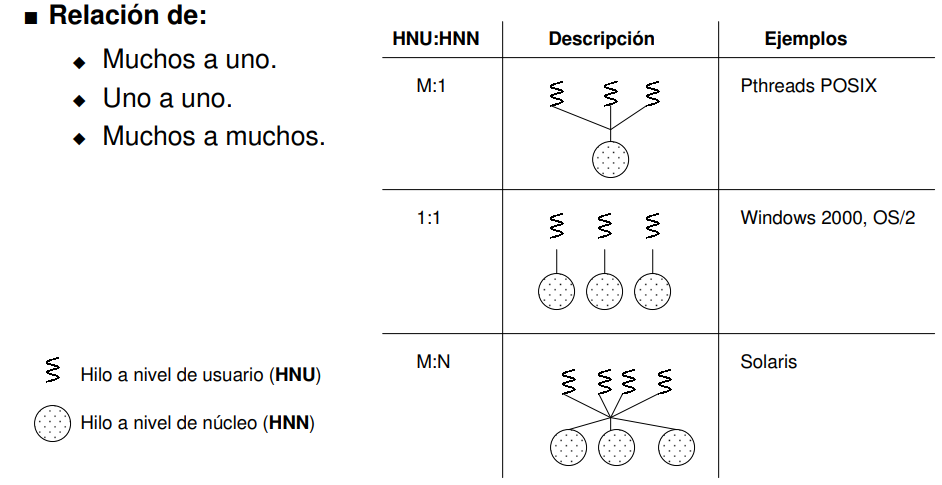


|  |  |
| --- | --- |
| **Ventajas frente a hilos a nivel de usuario** | **Desventajas frente a hilos a nivel de usuario** |
| El bloqueo de un hilo no supone (necesariamente) bloquear todo el proceso | El intercambio de hilos se realiza accediendo al núcleo |
| Planificación de múltiples hilos de un proceso en múltiples procesadores |  |
| Las funciones del núcleo pueden ser multihilo |  |

|  |
| --- |
| **APROXIMACION COMBINADA** |



|  |
| --- |
| **MODELO MULTIHILO** |



|  |  |  |
| --- | --- | --- |
| Ventajas de utilizar múltiples hilos | La ejecución concurrente de hilos de un mismo proceso puede mejorar la eficiencia del sistema | Paralelismo dentro del proceso (en multiprocesadores. |
| Las operaciones bloqueantes no paralizan al proceso (completo). |
| Mayor eficiencia que con múltiples procesos en | Creación/eliminación de unidades de planificación |
| Cambio de contexto |
| Una buena solución para sistemas cliente/servidor | |
| Facilidad de implementación | |

|  |
| --- |
| **EJERCICIO DE HILOS RESUELTO** |

Ejemplo:

Planificación S.O.: R.R. = 5

Planificación biblioteca hilos: R.R. = 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Proc | A | CPU | I/O1 | I/O2 | CPU |
| KLT | 4 | 7 | 10 |  | 5 |
| P1 | 2 | 10 |  |  | 5 |
| ULT1.1 |  | 6 |  | 5 | 5 |
| ULT1.2 |  | 10 | 10 |  | 4 |
| P2 | 0 | 7 |  |  | 9 |
| ULT2.1 |  | 1 | 5 |  | 10 |
| ULT2.2 |  | 6 |  | 5 | 10 |

