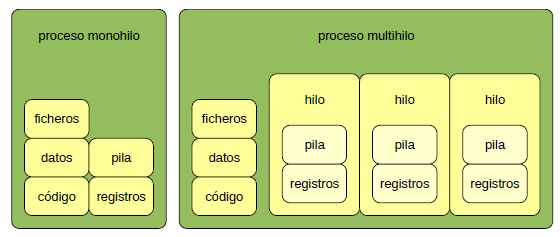
1. **Mencionar de forma breve y en sus palabras que es un hilo y que es un proceso**

**Programa** -> entidad pasiva con ordenes e instrucciones próximas a ejecutar.

Un **proceso** es una instancia de programa en ejecución, una entidad activa, si solo está realizando una tarea es considerado un solo hilo, tienen recursos independientes y diferentes estados (nuevo, listo, en ejecución, bloqueado y terminado) en los que se identifica la situación del proceso respecto a las tareas que está ejecutando.

Un **hilo** es la unidad básica del uso del CPU en sistemas multihilo, podríamos decir que es una pequeña fracción del código que se ejecuta de forma independiente o dicho de otra manera es una línea de ejecución de un proceso. Tienen dos tipos de flujo, único (monohilo) y múltiple (multihilo) en este caso se puede decir que hay un hilo padre que gobierna a hilos hijo.



1. **Cuál es la diferencia entre un hilo y un proceso**

Entre la información recabada pude llegar a las siguientes **diferencias** entre hilos y procesos:

* Los hilos son independientes en ejecución y trabajo, pero no en recursos en este caso coexisten (excepto que cada hilo deberá tener su propio stack), los procesos tienen sus propios recursos y no los comparten con otro proceso, al igual que su código y datos.
* Los procesos son completamente independientes, los hilos no; por lo que si un hilo se bloquea perjudica a los demás, es por ello que se deben programar cuidadosamente.
* El proceso es más costoso y un hilo es más ligero (en cuanto a trabajos/tareas)

Beneficios de usar hilos:

* Podemos decir que si segmentamos un proceso tenemos hilos con tareas diferentes según se haya programado, con ello la carga del trabajo se reparte entre los hilos.
* Económicamente es más fácil crearlos ya que comparten recursos de esta manera al sistema le resulta más ágil gestionarlos, además su contexto también es bajo ya que se guarda y recupera menos información.
* Se aprovecha la arquitectura del CPU ya que los hilos pueden ejecutarse en paralelo en distintos procesadores. Además, optimizan el tiempo de CPU al minimizar los tiempos de inactividad.

1. **Con sus palabras mencionar cual sería un esquema o como harían para que nuestro sistema, además de ser multitarea pudiéramos atender varios procesos a la vez, de qué forma controlar que cada núcleo tome una tarea diferente respetando el orden en que llegaron.**

No sé sentí un poco extraño el planteamiento, pero, propongo un sistema que sea multiproceso y a la vez paralelo. Multiproceso simétrico para que la carga de las tareas sea equilibrada y paralelo para que los procesos se segmenten en hilos y estos no lleguen a realizarse de forma concurrente en cada núcleo. Tal vez el orden de llegada debería ser mediante prioridad, pero de igual forma serían atendidos a la par dado que sería un sistema paralelo. Me supongo.

1. **Mencionar que es un sistema operativo multi tarea, un sistema operativo multi proceso y un sistema operativo paralelo.**

Sistema operativo **multitarea**:

Que permite al usuario realizar varias tareas al mismo tiempo. Existen varios tipos de multitareas:

* Conmutación de contextos: Dos o más apps se cargan al mismo tiempo, pero solo se procesa la que se encuentra en primer plano.
* Cooperativa: La tarea que se encuentra en segundo plano recibe atención mediante los tiempos muertos de la tarea en primer plano.
* Compartido: Cada tarea recibe atención del multiprocesador durante una fracción de segundo

Sistema operativo **multiproceso**:

Es capaz de usar todos los procesadores para distribuir la carga de trabajo. Estos sistemas trabajan de dos formas:

* Simétrica: Envía los procesos indistintamente a cualquier procesador disponible
* Asimétrica: Usa un procesador maestro el cual distribuye los procesos a los procesadores esclavos.

Sistema operativo **paralelo**:

Tiene la capacidad de realizar varias operaciones de manera simultánea. Pretende que cuando existan dos o más procesos que compitan por algún recurso se puedan realizar o ejecutar al mismo tiempo.

**Conclusiones:**

Considero que comprender la forma en que un sistema trabaja es importante dado que lo usamos día a día; y como siempre, el funcionamiento interno es indiferente para el usuario, pero es todo un reto para el programador, ya que aún se buscan maneras de darle una mejor experiencia al usuario. Una de las cosas que me resultaron interesantes es que anteriormente los esquemas se centraban en atender una tarea, pero con el paso del tiempo y debido a que aumentaban los requerimientos del cliente también aumentaban la cantidad de procesos a ejecutar; de esta manera llego a la hipótesis de que se planificaron esquemas para que se aprovecharan los recursos y se satisfagan las necesidades del cliente, programación de procesos ligeros y sincronización. Hoy en día debido a que nos encontramos con tecnología casi a cúspide, extendiendo horizontes a la “modernidad tecnológica” mis expectativas van a que con sistemas distribuidos aplica lo mismo, de una forma u otra se busca aprovechar y realizar las tareas de forma rápida y eficiente evolución de esquemas de comunicación, evolución de sistemas que atiendan procesos. Así como la tendencia a lo virtual, procesadores virtuales que apoyen en la ejecución de tareas.