- ¿Qué es un thread?

*Un thread es una serie de instrucciones en secuencia que tienen un propósito en común, este suele ser parte de un proceso.*

- ¿Qué es un Proceso?

*Un proceso es la ejecución de instrucciones que tienen un proósito en común, es decir de un programa.*

- ¿En C, Qué es lo que hace fork()?

- Explique con sus palabras lo que pasa

*fork () se utiliza para duplicar un proceso actual, así el proceso padre espera a que se termine de ejecutar el proceso hijo para seguir. En lo único que se diferencian es que tienen diferentes espacios de memoria asignados.*

Diagrama

Descripción generada automáticamente - Apoye su razonamiento con una Imagen / Dibujo.

- ¿En C, Qué es lo que hace pthread\_join()?

- Explique con sus palabras lo que pasa

*Un pthread\_join() se utiliza para bloquear threads hasta que termine un thread específico.*

- Apoye su razonamiento con una Imagen / Dibujo

*int pthread\_join(pthread\_t thread, void \*\*value\_ptr);*

T1

T1

T4

- ¿Cuál de los dos códigos en C adjuntos en la carpeta tiene un mejor rendimiento?

- Analice el código y su funcionamiento

*Según se puede observar, uno de ellos crea procesos para lo que utiliza la función fork() y el otro crea threads utilizando pthread\_create() y pthread\_join().*

- Razone su respuesta en base a lo que esta pasando.

*En los resultados se tienen que un thread se crea más rápido que un proceso, y se dice que la creación de un thread normalmente es más eficiente en cuestión de recursos.*

- Utilice una Tabla Comparativa.

|  |  |
| --- | --- |
| Codigo1 | Codigo2 |
| Crea procesos | Crea threads |
| Utiliza fork() para duplicar procesos | Utiliza pthread\_create() y pthread\_join() |
| Es más lento crear procesos | Es más rápido crear threads |

- En JAVA

- ¿Cuáles son las diferencias entre implementar un Thread como un extends de la clase Thread y con un Runnable?

*Una de las principales diferencias es que al utilizar Thread se está utilizando herencia, y java no permite herencia múltiple por lo que solo esa clase se podrá heredar y Runnable es una interface que es más flexible en la programación*

- Para agregar otro argumento a su razonamiento con las diferencias entre las implementaciones en JAVA, cree dos archivos java para cada implementación Thread y Runnable, los cuales imprimirán su nombre 100 veces.

*Implementando en java estas dos herramientas, se puede apreciar que al utilizar runnable esta implementación es un poco más tediosa de escribir, aunque si bien, lleva un poquito más de tiempo a nivel de código porque sería utilizar un método Thread y ponerle como atributo un objeto de runnable “new Thread(new ObjetoRunnable())”, este es mucho más utilizado ya que se pueden compartir objetos entre procesos no olvidando que tenemos la ventaja de poder heredar alguna otra clase que sea más importante para lo que estemos implementando.*