

Meza Vargas Brandon David

-----Lenguajes de programación y los mecanismos de comunicación de los procesos-----

La comunicación es algo esencial en la vida, la usamos todo el tiempo, desde que nos levantamos hasta la hora de dormir. Nos comunicamos para expresar alguna necesidad, establecer acuerdos, es decir, interactuamos con los demás con el objetivo de compartir información. Esto pasa incluso en los sistemas operativos, pues al existir más de un proceso, se crea la necesidad de comunicación y sincronización entre ellos. Existen varios mecanismos de comunicación, estos son los semáforos, la memoria compartida y tuberías, cada uno con sus propias características e implementación.

Podemos encontrar que estos mecanismos pueden ser programados en un lenguaje de programación, uno de los lenguajes más populares con el que podemos implementar estos mecanismos es C. Actualmente hay muchos lenguajes de programación, entonces, ¿hay otros lenguajes de programación que hagan uso de los semáforos, memoria compartida y tuberías?

Primeramente, hablemos de la programación concurrente, cuando un programa es concurrente puede soportar dos o más acciones en progreso, aquí los procesos se ejecutan de manera independiente. La concurrencia es una forma de estructurar una solución que puede ser paralelizable.

Antes de entrar en los lenguajes de programación que hacen uso de la programación concurrente y, por consiguiente, usan los mecanismos de comunicación entre procesos, veamos cada uno de estos mecanismos.

Tenemos a la memoria compartida, para tener memoria compartida, se crea una región fuera del espacio de direccionamiento de un proceso y cada proceso que necesite acceder a dicha región, la incluirá como parte de su espacio de direccionamiento virtual, de esta forma podrán leer o escribir datos en la memoria compartida.

Como parte de la memoria compartida tenemos a los semáforos, un semáforo es una variable especial que constituye el método clásico para restringir o permitir el acceso a recursos compartidos, es decir, la memoria compartida.

Los semáforos fueron introducidos por Dijkstra, y como bien sabemos, son elementos de los lenguajes de programación concurrentes, son elementos que pueden implementarse de forma muy eficiente, además de ser fáciles de entender.

Las tuberías permiten la comunicación y sincronización entre procesos, es común que usen el buffer de datos entre elementos consecutivos. Las tuberías solo pueden ser usadas en procesos relacionados, es decir, en procesos que tengan un ancestro en común.

Lo anterior se resume en la programación concurrente y contestando a la pregunta del inicio, si, si existen más lenguajes de programación que hagan uso de los elementos de la programación concurrente, aunque de los mecanismos para comunicar procesos, de los menos usados son las tuberías, lo contrario con la memoria compartida. Algunos lenguajes de programación que usan semáforos, memoria compartida y tuberías son:

Rust: este fue desarrollado por Firefox con el propósito de ser seguro, concurrente y práctico. Una de sus principales características es que es posible arreglar bugs en el tiempo de ejecución.

Go: go es un lenguaje compilado, concurrente y no orientado a objetos, este lenguaje es muy parecido a los lenguajes más usados, es implementado en modelos de negocios y manejo de servidores.

Elixir: este es utilizado en muchos frameworks para la implementación de chatbots en distintas redes sociales.

Ada: este nació en los 70s, es usado ampliamente en la industria de las infraestructuras de riesgo grande, como los sistemas de aviones, trenes, tanques y misiles.

Erlang: es un lenguaje de alto nivel, diseñado para escribir aplicaciones concurrentes y distribuidas de funcionamiento interrumpido.

Java: permite la concurrencia con la implementación de una librería llamada threads. Además, cuenta con otras librerías para el control de procesos y métodos de planificación, entre ellos semáforos.

Python: es el lenguaje más usado según la revista IEEE, es un lenguaje de sintaxis simple. Cuenta con varias librerías para la implementación de la programación concurrente y su respectivo control, entre estas librerías están threads y multiprocessing, que proveen herramientas como lo son los semáforos.

El hecho de que la mayoría de las personas conozca C para la implementación de semáforos, memoria compartida y tuberías, no quita el hecho de que hay más lenguajes de programación que también incluyen estos mecanismos como se vio en el presente ensayo.

A pesar de la existencia de muchos lenguajes de programación, C es de los más rápidos y sigue siendo de los más usados al momento de desarrollar un sistema operativo, ya que facilita muchas operaciones con el sistema, entre ellos, la comunicación de procesos.

En lo personal, C me resulta cómodo para trabajar de manera concurrente, además que no he profundizado en muchos lenguajes de programación. El uso de cada lenguaje de programación para el ámbito de la concurrencia y trabajar con la comunicación de procesos ya dependerá de cada persona, en la situación que se encuentre y sus necesidades o las de la empresa.