

**Programación paralela y con concurrente.**  
**Laboratorio 6.**  
**Dayana Marín Mayorga.**

**Pregunta a:**

**Revise la solución en Java**

**- Explique cómo se sincronizan los hilos.**

Para este caso se utilizan las funciones de acquire que hace una función de bloquear el hilo correspondiente y release que desbloquea el hilo correspondiente, controlando así el acceso sincronizado.

**- Explique cómo es manejada las condiciones de carrera**

Mediante las funciones acquire y reléase..

**Revise la solución en Python:**

En este caso se importa la clase Threading la cual nos permite la utilización de hilos y el método lock().

Este método nos permite que solo acceda un hilo el cual es pasado por parámetro a la función, y así se controla las condiciones de carrera.

También implementa los semáforos, con el Semaphore y con el signalH(self) o signalO(self) y sus homólogos con el wait, en ambos casos el "self" se utiliza para referirse a un hilo en específico.

## **Problemas de los filósofos comensales:**

### **Explicar que es un monitor y variables de condición.**

Las clases mutex implementan un mecanismo de exclusión mutua que permite que sólo un subproceso esté activo en una región crítica a la vez. Para que hilo entre en la región crítica otro hilo debe abandonarlo. Con mutex es imposible suspender un subproceso dentro de una región crítica y hacer que este subproceso se active nuevamente más adelante.

Entonces es aquí donde entra un monitor el cual es un mecanismo de sincronización que protege una región crítica. Un monitor permite suspender un hilo dentro de la región crítica, debido a esto se permite que otro hilo ingrese a la región crítica. El hilo original se despierta y continúa la ejecución dentro del monitor. Esto se extiende a cualquier número de subprocesos, así se puede suspender varios subprocesos dentro de un monitor.

### **Variables de condición:**

Para proporcionar mecanismos para sincronización de eventos un monitor puede contener variables de condición, las cuales pueden manipularse mediante las operaciones signal y wait que son análogas a las operaciones P y V en semáforos binarios, respectivamente.

### **En el directorio DiningPhilo/ ForkandSemaphores/**

**Revise la implementación de la clase Semaphore y explicar el funcionamiento de los métodos SP y SV.**

Estos métodos funcionan funcionan como un signal y un wait, el SV hace la función de signal y ese SP hace la función de wait.

## **Clase "philoFork-gv.cc".**

**Explique la manera de manipular la pantalla y de desplegar la salida en el cuadro.**

Se utiliza las funciones de las clases curses esto para manipular la pantalla, estas funciones básicamente permiten controlar la pantalla de la consola de Linux, para esto se define una serie de coordenadas para que pueda imprimir el resultado.

Para desplegar la salida se hace uso del método displayTitle mostrando los contenidos de la salida del programa, así como también el método displayPhilosopher que despliega la información del estado en que se encuentra los filósofos.

**Estudie la solución y explique cuales son las variables globales y cómo se utilizan.**

Se utiliza como variable global un arreglo de semáforos para acceder a los palillos o los llamados chopsticks para comer de manera simultánea y así indicarle a al philosopher si comer o no comer.

También se utiliza un arreglo de tipo mutex que regula el ingreso de código por parte de los diferentes procesos

**Explique la manera en que realiza la sincronización de procesos.**

Esta ocurre cuando se hace el llamado al proceso Philo, que permite por medio de semáforos cuando un philosopher utiliza los palillos para comer o cuando están pensando y se sienten hambrientos.

## **Clase philoFork-ngv.cc:**

Para esta se utiliza una estructura enviada por parámetro en el método philo, esta estructura posee un semáforo que se utiliza como un palillo disponible en la mesa, también otro semáforo se utiliza como una función de mutex , para controlar la impresión por consola, al hacer las cosas de esta manera no se necesitan variables globales.

**Explique la manera en que realiza la sincronización de procesos.**

Esta se da por medio de múltiples semáforos.

**Resuelve el problema de la misma manera que la solución anterior?**

Para este se utiliza la misma lógica pero no se hace con la misma solución.