

Documentación - Parte 3

Comenzamos declarando el tamaño de la memoria, la cantidad de CPUs y los semáforos.

Luego hicimos la tarea de **Semáforo**, junto a sus operaciones básicas, como `wait`, `signal` e `init`.

Una vez hecho esto, realizamos un array de Semáforos.

Debajo, definimos la tarea de la **Memoria**, que tiene dos operaciones, `Escribir`, que recibe una posición y un valor, guardando en esa posición el valor recibido, y `Leer`, que también una posición y un valor, donde se guardará lo que está en la posición dada.

Luego, definimos la task de la **CPU**, que posee un boolean Running, para saber cuándo dejar de correr el loop, un Integer A, que indica el valor acumulado, otro Integer IP que es el Instruction Pointer, que indicará la instrucción a seguir, un Integer Valor y un Pos para la posición, y por último, un Integer ID que es el identificador de la CPU. Dentro del `begin` tenemos las operaciones básicas de la CPU (indicadas en la letra del obligatorio).

```
0 = LOAD [Parámetro]
1 = STORE [Parámetro]
2 = ADD [Parámetro]
3 = SUB [Parámetro]
4 = BRCPU [Parametro]
5 = SEMINIT [Parametro], [Parametro]
6 = SEMWAIT [Parametro]
7 = SEMSIGNAL [Parametro]
```

→ Usamos la memoria dividida en dos, del 0 al 64 son los parámetros correspondientes a las funciones del 64 al 127. Por ejemplo, al inicial el programa, en el index 64 de la memoria (primera instrucción del IP para funciones) tenemos la función LOAD, y su parámetro se encuentra en la posición 0.

Para una referencia de cómo estructurar nuestro programa utilizamos Chatgpt, apoyándonos en el material del curso y clases grabadas.

Encontramos que nuestro código no funciona como esperábamos, por cuestiones de tiempo logramos entregar un gran avance de lo que sería el programa completo, sin errores ni advertencias.