

## Examen de Sistemas Operativos Parte Práctica

## Planificación de procesos y sincronización

Un determinado sistema operativo monoprocesador tiene que ejecutar un programa de cálculo que se compone de tres hilos de ejecución, tal y como aparece a continuación:

```
//Variables globales a todos los hilos
bool finh2=false;
bool finh3=false;
int resultadoA;
int resultadoB;
int resultadoC;

void hilo3() {
    resultadoA=calcula();
    finh3=true;
}

void hilo2() {
    resultadoB=calcula();
    finh2=true;
}

void hilo1() {
    while (!finh2 && !finh3) {};
    resultadoC=resultadoA+resultadoB;
}
```

Para responder a las siguientes cuestiones, considere que los tres hilos son los únicos hilos en estado *ready* del sistema y que en dicha cola `hilo1` es el primero, luego `hilo2` y por último `hilo3`. Además, considere que todas las instrucciones tardan 1 u.t en ejecutarse excepto la que contiene la llamada a la función `calcula()`, que tarda 4 u.t. Por instrucción entenderemos cada línea del programa, teniendo en cuenta que si la instrucción es un bucle, cada iteración contará como una instrucción.

1. El responsable del proyecto se pregunta si sería posible utilizar planificación FIFO para los hilos, o por lo contrario habría que recurrir a un sistema más caro con planificación *round-robin* con un  $q=2$ . Complete el cronograma 1 para ambos esquemas de planificación, FIFO y *round-robin* y a la vista de los resultados haga una propuesta justificada (máximo 5 líneas). *3 puntos*.

**RESPUESTA:**

No se puede usar FIFO, ya que el bucle de espera activa monopolizaría la CPU y no dejaría que otro proceso utilizara la CPU. El cronograma resultante sería el siguiente:

[illegible]

Sería necesario utilizar RR para que esta solución funcionara.

[illegible]

2. Uno de los ingenieros propone la siguiente mejora basada en el uso de semáforos.

```

1
2 //Variables globales a todos los hilos
3 semaphore S2 <- 0;
4 semaphore S3 <- 0;
5 int resultadoA;
6 int resultadoB;
7 int resultadoC;
8
9 void hilo3() {
10     resultadoA=calcula();
11     V(S3);
12 }
13
14 void hilo2() {
15     resultadoB=calcula();
16     V(S2);
17 }
18
19 void hilo1() {
20     P(S3);
21     P(S2);
22     resultadoC=resultadoA+resultadoB;
23 }

```

Las instrucciones de las líneas 3 y 4 indican la declaración de dos variables de tipo semáforo inicializadas ambas a 0. Las operaciones P y V, como cualquier otra operación, tardan 1 u.t, independientemente de que provoquen bloqueo o no.

Hay quien dice que esta solución no es válida porque lleva a bloqueo, otros dicen que no mejora a la anterior porque obliga a utilizar planificación *round-robin* que es más cara. Rellene el cronograma 2 para ambos esquemas de planificación, FIFO y *round-robin* y a la vista de los resultados formule su propia opinión justificada. 7 puntos.

7 puntos.

(10 puntos)

**RESPUESTA:**

La solución por semáforos no provoca abrazo mortal ni obliga a utilizar *round-robin*. Simplemente lleva a perfiles de ejecución diferentes.

[illegible]

Con RR sería así:

H1	■									■	■						
H2		■	■			■	■		■								
H3				■	■			■	■		■						