



## Programación Concurrente y Paralela: Cuestionario 2

Alumno: Bañuelos López Marco Antonio

Matrícula: 201726707



### CUESTIONARIO 2

1. ¿Cuáles son las ventajas que ofrecen los monitores en comparación con los semáforos?

Aunque ambos se usan para resolver problemas de la sección crítica y lograr la sincronización de procesos, las ventajas de los monitores frente a los semáforos, es la exclusión mutua automática, lo que permite superar errores de sincronización, además de utilizar variables compartidas globales, lo que nos proporciona un código más organizado y estructurado, con una implementación fácil

2. Describa en qué consisten las variables condición aplicadas sobre monitores

Condición es un tipo de variable predefinido que solo puede instanciarse en los monitores, las variables de este tipo se utilizan para suspender y activar procesos que han accedido al monitor, y que por el estado en que se encuentran, no pueden continuar su ejecución. Las variables de tipo condición se declaran dentro de un monitor proporcionando la capacidad de sincronización a los monitores.

3. Describa en qué consisten las operaciones DELAY, RESUME y EMPTY aplicadas a las variables condición de los monitores.

- Delay: suspende el proceso que lo ejecuta y lo incluye en la lista de una variable Condición.
- Resume: reactiva un proceso de la lista asociada a una variable Condición.
- Empty: función que retorna true si la lista de procesos de la variable está vacía.

4. La operación resume implica un problema potencial, el activar dos procesos ¿Cómo se resuelve?

Puede resultar que al ejecutar *Resume*, resulten dos procesos activos dentro del monitor, un proceso que le ejecuta, y otro proceso de la cola asociada a la variable *condition* se proponen 3 maneras de resolver el problema.

- Reactiva y Continúa: El proceso activado permanece suspendido hasta que el proceso que ha invocado *resume* libera el monitor.



## Programación Concurrente y Paralela: Cuestionario 2

Alumno: Bañuelos López Marco Antonio

Matrícula: 201726707



- **Reactivación Inmediata**: La ejecución de la operación *resume*, supone la **liberación** del monitor por parte del proceso que la realiza. El único proceso activo dentro del monitor es el proceso que se ha reactivado.
- **Reactiva y espera**: El proceso que ejecuta *resume* se bloquea en una cola especial denominada “**urgente**”, y el proceso activado es el que tiene el acceso sobre el recurso. Cuando el proceso suspendido sobre la cola “**urgente**” se activa ejecuta la sentencia que sigue a la sentencia resume que la bloqueó.

5. En un programa con N procesos concurrentes ¿cuál es el número máximo posible de procesos bloqueados en la cola de un monitor?

6. En un programa concurrente hay 7 procesos y 1 monitor con una variable de tipo condición. ¿Es posible que en momento determinado haya un proceso en estado de ejecución y 6 procesos en estado bloqueado, 4 de ellos en la cola de entrada al monitor, otros 2 en la variable condición?

No, ya que simplemente no puede haber 2 procesos bloqueados en la variable condición, eso significaría que hay un problema de interbloqueo

7. ¿Puede haber procesos bloqueados en la cola de entrada y en cola urgente de un monitor?

Si se realiza una correcta implementación del método de *reactiva y espera*, entonces no es posible eso.

8. Considerar el monitor y los procesos concurrentes:

¿Qué se escribe en pantalla si el proceso 1 ejecuta primero el procedimiento P1 del monitor?

¿Qué se escribe en pantalla si el proceso 2 ejecuta primero el procedimiento P2 del monitor?



## Programación Concurrente y Paralela: Cuestionario 2

Alumno: Bañuelos López Marco Antonio

Matrícula: 201726707



```
MONITOR M;
EXPORT P1,P2;
VAR q: CONDITION;

PROCEDURE P1;
BEGIN
  DELAY(q);
  RESUME(q);
  WRITE('1')
END;

PROCEDURE P2;
BEGIN
  RESUME(q);
  DELAY(q);
  WRITE('2')
END;

END;

PROCESS Procesos1;
BEGIN
  M.P1
END;

PROCESS Procesos2;
BEGIN
  M.P2
END;
```

9. Sean M1 y M2 los dos siguientes procedimientos exportados del monitor M:

Si un proceso P1 ejecuta una sola vez M1, y otro proceso P2 ejecuta una sola vez M2, escriba las salidas por pantalla en los dos siguientes casos:

a) El proceso P1 ejecuta primero el procedimiento M1 del monitor:

**Resume C2, Escribe A**

b) El proceso P2 ejecuta primero el procedimiento M2 del monitor:

**Delay C2, escribe C**

```
MONITOR M;
EXPORT M1, M2;
VAR C1, C2 : CONDITION;

PROCEDURE M1;
BEGIN
  IF NOT EMPTY(C2) THEN
    RESUME(C2);
  DELAY(C1);
  WRITE('A')
END;

PROCEDURE M2;
BEGIN
  IF EMPTY(C1) THEN
    DELAY(C2);
  WRITE('B');
  IF NOT EMPTY(C1) THEN
    RESUME(C1);
  WRITE('C')
END;
```

10. Dado el siguiente monitor y los siguientes procesos concurrentes: MONITOR M;



## Programación Concurrente y Paralela: Cuestionario 2

Alumno: Bañuelos López Marco Antonio

Matrícula: 201726707



```
MONITOR M;
EXPORT P1, P2;
VAR q: CONDITION;

PROCEDURE P1;
BEGIN
    DELAY(q);
    WRITE('1');
    IF NOT EMPTY(q) THEN
        RESUME(q)
    END;
END;

PROCEDURE P2;
BEGIN
    RESUME(q);
    WRITE('2');
    IF EMPTY(q) THEN
        DELAY(q)
    END;
END;

PROCESS Proceso1;
BEGIN
    M.P1
END;

PROCESS Proceso2;
BEGIN
    M.P2
END;
```

¿Qué se escribe en la pantalla en los dos siguientes casos?

a) El proceso 1 ejecuta primero el procedimiento P1 del monitor.

El proceso escribe 1 y 2, “q” se queda con P2 encolado

b) El proceso 2 ejecuta primero el procedimiento P2 del monitor.

Escribe 2 y “q” se queda con P1 encolado

11. Dado el siguiente monitor y los siguientes procesos concurrentes:

```
MONITOR M;
EXPORT P1, P2;
VAR q : CONDITION;

PROCEDURE P1;
BEGIN
    DELAY(q);
    WRITE('A');
    RESUME(q);
    DELAY(q);
    WRITE('B');
END;

PROCEDURE P2;
BEGIN
    RESUME(q);
    WRITE('C');
    DELAY(q);
    RESUME(q);
    WRITE('D');
END;

PROCESS Proceso1;
BEGIN
    M.P1;
END;

PROCESS Proceso2;
BEGIN
    M.P2;
END;
```

Escriba las salidas por pantalla en los dos siguientes casos:

a) El proceso 1 ejecuta primero el procedimiento 1 del monitor:

Imprime: “AC”



## **Programación Concurrente y Paralela: Cuestionario 2**

*Alumno:* Bañuelos López Marco Antonio

*Matrícula:* 201726707



b) El proceso 2 ejecuta primero el procedimiento 2 del monitor:

Imprime: "C"