

## b) El acceso con exclusión mutua a un recurso.

La cola de mensajes se inicializa fuera del código de los procesos con un mensaje que no requiere tener ningún contenido y que se denotará como nulo.

Un proceso que desea ejecutar su sección crítica primero debe ejecutar una operación receive para recibir un mensaje. Si el buzón está vacío entonces se bloquea hasta que llegue un mensaje a la cola. Cuando dicho evento ocurra entonces el proceso obtiene el mensaje (que se supone que se borra de cola) y puede entrar en su sección crítica. Cuando sale ejecuta una operación send para enviar el mensaje de nuevo a la cola, y permitir que otro proceso que estuviera bloqueado en la cola de mensajes pueda desbloquearse y acceder a su región crítica.

## Preguntas de autoevaluación tema 5

### 5.1. ¿Qué es un interbloqueo?

Aquella situación en la cual un conjunto de procesos está bloqueado en espera de la liberación de uno o varios recursos que se encuentran asignados a otro proceso del mismo conjunto. Como todos los procesos del conjunto están bloqueados, ninguno de ellos liberará los recursos que posee y que otro proceso necesita, y en consecuencia ninguno puede continuar su ejecución.

### 5.2. Enumerar y explicar las cuatro condiciones necesarias y suficientes para la existencia del interbloqueo.

- ❖ **Exclusión mutua.** Cada instancia de un recurso solo puede ser asignada a un proceso como máximo.
- ❖ **Retención y espera.** Cada proceso retiene los recursos que le han sido asignado mientras espera por la adquisición de los otros recursos que necesita.
- ❖ **No existencia de expropiación.** Si un proceso posee un recurso, éste no se le puede expropiar.
- ❖ **Espera circular.** Existe una cadena circular de dos o más procesos, de tal forma que cada proceso de la cadena se encuentra esperando por un recurso retenido por el siguiente proceso de la cadena.

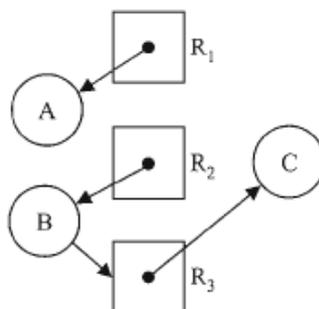
### 5.3. Enumerar las principales estrategias que puede implementar un sistema operativo para el tratamiento de los interbloqueos.

- ❖ **Prevención de interbloqueos.** Consiste en impedir que se produzca alguna de las cuatro condiciones necesarias para que se produzca el interbloqueo.
- ❖ **Evitación de interbloqueos.** Consiste en examinar las posibles consecuencias de asignar los recursos solicitados y evitar el interbloqueo mediante la no asignación de recursos en situaciones potencialmente peligrosas.
- ❖ **Detección y recuperación de interbloqueos.** Consiste en comprobar cada cierto tiempo el estado del sistema para detectar la existencia de interbloqueos y en el caso de que existan tomar las medidas oportunas para eliminarlos.

#### 5.4. ¿Cómo se construye un grafo de asignación de recursos?

Se representa a cada proceso con un círculo y a cada recurso con un cuadrado. Cada instancia de un mismo recurso, se representa con un punto dentro del cuadrado asignado al recurso. Si un proceso ha solicitado una instancia de recurso y se encuentra bloqueado en espera de que le sea asignada, entonces se representa una flecha que sale del proceso hacia el recurso. Por otra parte, si un proceso tiene asignado una instancia de un recurso, entonces se representa una flecha que va desde la instancia del recurso al proceso.

Considérese un computador con una unidad de DVD (recurso  $R_1$ ), una impresora (recurso  $R_2$ ) y un disco duro (recurso  $R_3$ ). Supóngase que se están ejecutando tres procesos A, B y C. En la Figura 5.1 se muestra el grafo de asignación de recursos en un determinado de tiempo  $T_1$ . El proceso A tiene asignado  $R_1$ , el proceso B tiene asignado  $R_2$  y se encuentra bloqueado en espera de que le asignen  $R_3$ , que está asignado al proceso C. Se observa que en el grafo no existe ningún ciclo, luego no existe interbloqueo.



#### 5.5. ¿Cómo se pueden detectar los interbloqueos en un grafo de asignación de recursos?

Si el grafo no contiene ningún camino que sea un ciclo entonces no existe interbloqueo. Por otra parte la existencia de un ciclo no es condición suficiente para que exista interbloqueo,

dependerá también del número de instancias de cada recurso implicado en el ciclo y del tipo de caminos de los que formen parte dichas instancias. Si los recursos que forman parte de un ciclo únicamente tienen una instancia entonces existe interbloqueo. Por otra parte, si en el ciclo existen recursos que tienen más de una instancia, entonces para que exista interbloqueo todas las instancias de dichos recursos deben formar parte de caminos que sean ciclos.

#### **5.6. ¿En qué consiste la estrategia de prevención de interbloqueos?**

Consiste en eliminar la aparición de alguna de las cuatro condiciones necesarias y suficientes para que se produzca un interbloqueo: exclusión mutua, retención y espera, no existencia de expropiación y espera circular.

#### **5.7. ¿Es aconsejable eliminar la condición de exclusión mutua para prevenir el interbloqueo?**

No si se quiere garantizar la integridad y la no corrupción de los mismos. Por ejemplo, si dos procesos quieren escribir en una impresora no deberían de hacerlo al mismo tiempo.

#### **5.8. Señalar los inconvenientes de la eliminación de la condición de retención y espera para prevenir el interbloqueo.**

Se puede eliminar forzando a que un proceso solicite a la vez todos los recursos que va a necesitar en su ejecución. Hasta que no pueda obtener todos los recursos permanecerá bloqueado.

Uno de los principales inconvenientes que tiene la eliminación de esta condición es que requiere conocer por anticipado todos los recursos que va a necesitar un proceso durante su ejecución, lo cual en muchas ocasiones (salvo quizás en trabajos por lotes) es difícil de saber.

Otro inconveniente es que produce problemas de rendimiento del sistema. Un proceso puede permanecer bloqueado un tiempo excesivo o incluso permanentemente (inanición) hasta poder obtener a la vez todos los recursos que necesita.

#### **5.9. Explicar cómo se consigue la eliminación de la condición de no existencia de expropiación en la estrategia de prevención de interbloqueo.**

Se consigue permitiendo que el SO pueda expropiar a un proceso los recursos que retiene. Solo es posible en recursos para los cuales el SO puede salvar el estado en el que se encontraban, para restaurarlo cuando el proceso expropiado consiga otra vez su uso. Este es el caso del procesador (cambio de contexto de un proceso) y la memoria principal (intercambio en la memoria secundaria). En consecuencia se debe valorar si la desaparición de los interbloqueos mediante la eliminación de la existencia de expropiación compensa la disminución del rendimiento del sistema debido a la sobrecarga por las tareas de guardar y restaurar el estado de los recursos.

#### **5.10. Explicar una posible forma de eliminar la condición de espera con objeto de prevenir el interbloqueo.**

Se puede eliminar asignando a cada recurso del sistema un número y forzando a que un proceso solo pueda solicitar los recursos en orden ascendente. La principal desventaja es que los recursos deben ser solicitados en el orden establecido, en lugar de pedirlos cuando se necesitan realmente. De esta forma, un proceso retiene un recurso que no va a utilizar hasta haber utilizado otros. Impidiendo así que pueda ser asignado a otro proceso.

#### **5.11. Explicar en qué consiste la estrategia de evitación de interbloqueos.**

También conocida como predicción de interbloqueos consiste en conceder a un proceso solamente aquellas peticiones de recursos que tengan garantizado que no conducirán a un estado de interbloqueo.

En esta estrategia un proceso tiene que especificar por adelantado todos los recursos que va a necesitar para su ejecución. Obviamente si los recursos solicitados por un proceso superan el número máximo de recursos (asignados o no) del sistema, entonces, dicho proceso no es admitido para ser ejecutado. Si están disponibles y su concesión no conduce a un estado de interbloqueo entonces se le concede su uso. En caso contrario, el proceso tiene que esperar y se bloquea.

El SO debe contabilizar el número de recursos disponibles, el número de recursos asignados a cada proceso y el número de

recursos que restan por asignar a cada proceso. Además, se encarga de comprobar si una asignación es segura o, por el contrario, puede conducir a un estado de interbloqueo.

#### **5.12. ¿Qué es un estado seguro?**

El estado del sistema con respecto a la asignación de sus recursos en un determinado instante de tiempo queda determinado por los vectores **Re**, **Rd** y las matrices **N** y **A**. Un estado es seguro si posibilita al menos una secuencia de asignación de recursos a los procesos que garantiza que éstos se pueden ejecutar hasta su finalización sin que se produzca interbloqueo, incluso aunque los procesos soliciten simultáneamente todos los recursos que van a necesitar.

#### **¿Y un estado inseguro?**

. Un estado inseguro es aquél que no garantiza que no se pueda producir un interbloqueo.

#### **5.13. Describir la técnica de denegación de asignación de recursos.**

La técnica más usada de evitación de interbloqueos es la denegación de asignación de recursos que consiste en no conceder a un proceso una petición adicional de un recurso si dicha petición puede conducir a un estado inseguro.

#### **5.14. ¿Para qué se utiliza el algoritmo del banquero?**

La implementación más conocida de la técnica de degeneración de asignación de recursos. Simula el comportamiento de un banquero que realiza préstamos y recibe pagos sin caer nunca en la posibilidad de no poder satisfacer todas las necesidades de sus clientes. La analogía es la siguiente: el banquero representa al SO, los clientes son los procesos y el dinero prestado los recursos.

El algoritmo asegura que el número de recursos asignados a todos los procesos nunca puede exceder del número de recursos del sistema. Además, nunca se puede hacer una asignación peligrosa, es decir, asignar recursos de modo que no queden suficientes para satisfacer las necesidades de todos los procesos.

#### **¿Cuáles son sus pasos?**

Los pasos de que consta este algoritmo son los siguientes:

1. Se parte del un estado inicial de asignación seguro  $S_k$  con  $k = 0$ .
  2. Cuando un proceso realiza una petición de asignación de recursos, se simula que se concede la petición y se actualiza el estado del sistema. A este estado ficticio se le denota como  $S'$ :
  3. Se comprueba si  $S'$  es seguro.
- . En caso afirmativo se concede la petición al proceso y se actualiza el estado del sistema:
- $$S_{k+1} = S'$$
- . En caso negativo la petición se deniega y el proceso se bloquea hasta que se le puedan conceder los recursos que solicita.
1. Volver al paso 2.

### ¿Qué inconvenientes presenta su uso?

En la práctica no se utiliza, ya que es difícil conocer por adelantado los recursos que van a necesitar los procesos durante su ejecución. Además dicho algoritmo parte de la suposición de que el número de procesos y de recursos existentes es fijo, lo cual no tiene por qué ser cierto. En general, la población de procesos varía dinámicamente en el tiempo. Por otra parte, los recursos que inicialmente están disponibles pueden que con el tiempo no lo estén debido a posibles fallos, por ejemplo, que se quede atascado papel en una impresora o que se dañe una unidad de cinta.

### 5.15. Describir la técnica de denegación de la iniciación de un proceso.

Consiste en no iniciar un proceso si el número máximo de recursos que va a necesitar durante su ejecución pueden conducir a un interbloqueo.

El proceso  $p+1$  puede iniciar su ejecución si el número máximo de instancias de cada recurso  $j$  que necesita sumadas al número máximo de instancias de dichos recursos necesitadas por los procesos ya existentes no supera el número de instancias existentes. En consecuencia, el proceso puede iniciar su ejecución si el sistema está en un estado seguro.

No es óptima ya que supone que todos los procesos van a solicitar simultáneamente todas las instancias de recursos que van a necesitar, es decir, trabaja con el peor de los casos posibles

#### 5.16. ¿En qué consiste la estrategia de detección y recuperación de interbloqueos?

En no limitar las asignaciones de recursos a los procesos y comprobar periódicamente, mediante la utilización de algún algoritmo, si se ha producido algún interbloqueo. En caso afirmativo se emplea alguna técnica para intentar recuperar al sistema del interbloqueo.

#### 5.17. Enumerar los pasos del algoritmo de Coffman para la detección de interbloqueos.

Este algoritmo utiliza los siguientes vectores y matrices:

- **Re**: vector de recursos existentes
- **Rd**: vector de recursos disponibles
- **A**: Matriz de recursos asignados a cada proceso
- **M**: Matriz de recursos necesitados por cada proceso

adicionalmente de los que ya posee.

1. Se marca cada proceso que tenga una fila  $i$  ( $i = 1, 2, \dots, p$ ) de la matriz de asignación **A** igual a cero ya que dicho proceso no retiene ningún recurso y en consecuencia no puede producir interbloqueo.
2. Se realiza la asignación  $\mathbf{X} = \mathbf{Rd}$  donde **X** es un vector auxiliar.
3. Para cada "i" no marcado se comprueba la condición  $M_i$  es mayor o igual a  $X$  donde  $M_i$  es la fila  $i$  de la matriz **M**.
4. Si no existe ningún  $i$  que cumpla la condición, entonces el algoritmo finaliza. Si existe algún  $i$  que cumpla la condición, entonces se marca el proceso, se realiza la suma vectorial  $\mathbf{X} = \mathbf{X} + \mathbf{A}_i$  y se vuelve al paso 3.

#### 5.18. Describir algunas técnicas de recuperación del interbloqueo.

- ❖ **Recuperación mediante expropiación de recursos.**  
Consiste en ir expropiando recursos a algunos procesos y concedérselos a otros hasta conseguir salir del interbloqueo. Esta técnica requiere establecer algún criterio para seleccionar los procesos a los que se les va a expropiar los recursos. En ocasiones como por ejemplo en los sistemas por lotes la expropiación se debe hacer de manera manual por el administrador del sistema.
- ❖ **Recuperación mediante el retroceso de los procesos interbloqueados a algún punto de control previo.** Consiste en ir realizando puntos de control (checkpoints) periódicamente durante el tiempo de vida de un proceso. En cada punto de control se guarda el contexto del proceso en un fichero distinto. Cuando se detecta que el proceso está interbloqueado se retrocede a su ejecución a un punto de control anterior con la esperanza de que se evite el interbloqueo.
- ❖ **Recuperación mediante el aborto selectivo de procesos.**  
Consiste en ir abortando selectivamente algún proceso para que libere los recursos que posee y puedan ser asignados a algún proceso interbloqueado de forma que pueda salir de dicho estado y continuar la ejecución. Esta técnica requiere establecer algún criterio de selección de los procesos que van a ser abortados, ya sea alguno de los interbloqueados o cualquier otro proceso.
- ❖ **Recuperación mediante el aborto de todos los procesos interbloqueados.** Se trata de la solución más drástica y curiosamente la más utilizada.

#### **5.19. Señalar las ventajas y los inconvenientes de las técnicas de detección y recuperación de interbloqueos.**

Es una estrategia de tratamiento de interbloqueos mucho menos conservativa que la prevención y la evitación de interbloqueos, ya que al no limitar el número de asignaciones permite un mayor grado de concurrencia de procesos.

La detección de interbloques introduce un grado de sobrecarga al sistema que puede ser tolerable mediante el uso de algoritmos de detección eficientes y mediante el ajuste de la frecuencia de invocación de dicho algoritmo.

Por otra parte la recuperación de interbloques puede producir una sobrecarga elevada al sistema. Además puede producir un desaprovechamiento de los recursos para aquellos procesos que son reiniciados o retornados a un punto de control anterior.

En general puede afirmarse que la recuperación de interbloques es bastante útil en sistemas con baja probabilidad de interbloques.

#### **5.20. Explicar en qué consiste una estrategia mixta de tratamiento de interbloques.**

Las estrategias de tratamiento de interbloques descritas pueden utilizarse conjuntamente con objeto de aprovechar sus ventajas y reducir sus inconvenientes. La forma de proceder es la siguiente:

1. Agrupar los recursos existentes del sistema en diferentes clases.
2. Ordenar las clases para evitar la espera circular
3. Utilizar para cada clase la estrategia de tratamiento de los interbloques que se considere más oportuna.

#### **5.21. ¿En qué hipótesis se basa la estrategia de ignorar los interbloques?**

Tratar los interbloques supone algún coste al sistema. Por este motivo muchos SO como UNIX o Windows, no utilizan ninguna estrategia de tratamiento de interbloques. Consideran que los usuarios prefieren sufrir algún interbloqueo de vez en cuando que disponer de un sistema que les limite el uso de recursos o con alta sobrecarga. Se basan en la suposición de que la probabilidad de que se produzca un interbloqueo es pequeña y confían en el azar para que no se produzca. En el caso de que se produzca será el

administrador del sistema el que tenga que encargarse de detectar el interbloqueo y recuperarlo.

## **Preguntas de autoevaluación tema 6**

### **6.1. Definir los siguientes conceptos: a) Espacio del núcleo. b) Espacio de usuario.**

a). Un SO para ser ejecutado debe estar cargado en la memoria principal. El SO se carga cuando arranca el computador mediante un programa cargador. Al espacio ocupado por el código, las estructuras de datos y la pila (o pilas) del núcleo del SO se le denomina espacio del núcleo.

b). Aparte del SO, en la memoria principal pueden estar cargados (total o parcialmente) uno o varios procesos. Al espacio de memoria principal ocupado por la imagen de un proceso, es decir, por su espacio de direcciones de memoria lógica, se le denomina espacio de usuario.

### **6.2. ¿Qué es el área de intercambio en memoria secundaria?**

Cuando se desea ejecutar un programa en el SO tiene que crear un proceso asociado a un programa en la memoria secundaria y cargarlo en la principal. Para ello debe buscar el archivo ejecutable, crear una copia de la imagen del proceso en la memoria secundaria y cargar dicha imagen en memoria principal. Algunos SO directamente crean y cargan la imagen del proceso en la memoria principal sin crear una copia en la memoria secundaria.

El SO reserva espacio en la memoria secundaria, típicamente un HD, para almacenar las copias de las imágenes de los procesos. A dicho espacio se le denomina área de intercambio. El tamaño de esa zona suele estar predefinido, aunque algunos sistemas permiten que su tamaño máximo sea configurado por el administrador.

### **6.3. ¿En qué consiste la operación de intercambio de procesos?**