

Sistemas Operativos II

Módulo II

ABRAZO MORTAL O INTERBLOQUEO

1

Temas del Módulo II

- **Fundamentos de interbloqueo (abrazo mortal).**
 - Definición de interbloqueo.
 - Diagrama de progreso conjunto.
 - Tipos de recursos.
 - Caracterización de la asignación de recursos.
 - Condiciones para el interbloqueo.
- **Detección del abrazo mortal.**
 - Grafos de asignación de recursos.
 - Grafos *wait-for*.
- **Recuperación del abrazo mortal.**
 - Recuperación por medio de apropiación.
 - Recuperación a través del retroceso.
 - Recuperación a través de la eliminación de procesos.
- **Prevención del abrazo mortal.**
 - Exclusión mutua. Retención y espera.
 - Sin expropiación. Espera circular.

2

Fundamentos de interbloqueo

Definición de interbloqueo o abrazo mortal.

- Se puede definir el interbloqueo o abrazo mortal, como el **bloqueo permanente** de un conjunto de procesos que compiten por recursos o se comunican entre sí.
- Cada proceso está **bloqueado esperando un evento** que sólo puede generar otro proceso bloqueado del conjunto.
- El interbloqueo es **permanente** porque **no puede producirse ninguno de los eventos**. Esto implica que los recursos involucrados dejan de estar disponibles en el sistema.
- A diferencia de otros problemas de la concurrencia, **no hay una solución eficiente para el caso general**.

3

Fundamentos de interbloqueo

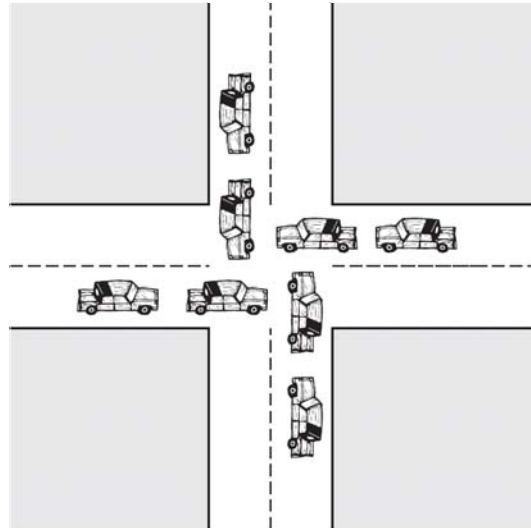
Definición de interbloqueo o abrazo mortal. (cont.)

- Todos los interbloqueos involucran necesidades conflictivas que afectan a los recursos de **dos o más procesos**.
- Un ejemplo típico es el del tráfico en el cruce de dos avenidas, cuando 4 autos llegan aproximadamente al mismo tiempo.
- La norma establece que la prioridad de paso la tiene el vehículo de la derecha, pero si llegaron los cuatro juntos, cada uno le cede el paso al otro y se produce el interbloqueo.
- Cuando son menos vehículos, la norma funciona sin problemas, lo que significa que hay al menos un recurso (una mano de una de las avenidas) disponible, por lo que uno de los vehículos pasará, mientras los otros esperan, y luego podrán pasar de a uno, sin que haya interbloqueo.

4

Fundamentos de interbloqueo

Definición de interbloqueo o abrazo mortal. (cont.)



© Stallings, 2005

5

Fundamentos de interbloqueo

Diagrama de progreso conjunto.

- Una manera práctica de visualizar el problema es el **diagrama de progreso conjunto**. Éste muestra el progreso de **dos procesos** compitiendo por **dos recursos**.
- Cada proceso necesita el **uso exclusivo de ambos recursos** por un cierto período de tiempo.
- Supongamos dos procesos, P y Q. Utilizando un sistema de ejes coordenados se representa, en el eje x, el progreso de la ejecución de P, mientras que en el eje y, el progreso de Q.
- En un monoprocesador sólo puede ejecutar un proceso a la vez, por lo que la trayectoria que representa los progresos consistirá en segmentos horizontales y verticales alternados.
- Entonces, en un **segmento horizontal P avanza y Q espera**, y en un **segmento vertical, Q avanza y P espera**.

6

Fundamentos de interbloqueo

Diagrama de progreso conjunto. (cont.)

- Supongamos que la estructura general de los procesos es la siguiente:

Proceso P	Proceso Q
...	...
Solicita A	Solicita B
...	...
Solicita B	Solicita A
...	...
Libera A	Libera B
...	...
Libera B	Libera A
...	...

7

Fundamentos de interbloqueo

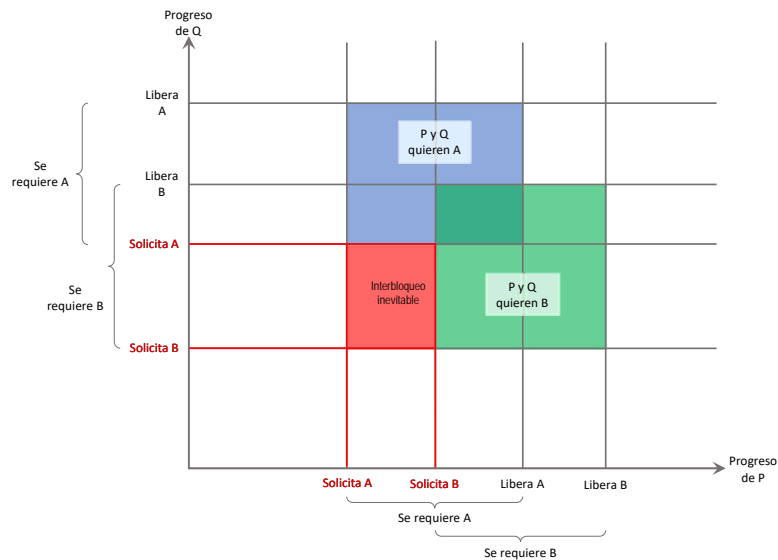
Diagrama de progreso conjunto. (cont.)

- En la figura siguiente, se muestran áreas en las que tanto P como Q requieren el recurso A; áreas en las que ambos procesos requieren el recurso B; y áreas en las que ambos requieren ambos recursos.
- Como asumimos que cada proceso requiere el uso exclusivo de un recurso, todas éstas son **regiones prohibidas**.
- Esto significa que es imposible que una trayectoria que represente el progreso de la ejecución conjunta de P y Q entre en una de estas regiones.
- El área sombreada en rojo se puede denominar **región fatal**. Si una trayectoria de ejecución entra en ella, el interbloqueo es inevitable.
- Si bien la existencia de esta región depende de la lógica de los procesos, el interbloqueo sólo es inevitable si se entra en ella.

8

Fundamentos de interbloqueo

Diagrama de progreso conjunto. (cont.)



9

Fundamentos de interbloqueo

Diagrama de progreso conjunto. (cont.)

- La figura anterior muestra seis diferentes trayectorias de ejecución:
 1. Q adquiere B y luego A; más tarde libera ambos. Cuando P continúe su ejecución podrá adquirir ambos recursos.
 2. Q adquiere B y luego A; P ejecuta y se bloquea al pedir A. Q libera B y A, y cuando P continúe, podrá adquirir ambos recursos.
 3. Q adquiere B y luego P adquiere A. El interbloqueo es inevitable, ya que cuando la ejecución siga, Q se bloqueará al pedir A, y P al pedir B.
 4. P adquiere A y luego Q adquiere B. El interbloqueo es inevitable, ya que cuando la ejecución siga, Q se bloqueará al pedir A, y P al pedir B.
 5. P adquiere A y luego B; Q ejecuta y se bloquea al pedir B. P libera A y B, y cuando Q continúe, podrá adquirir ambos recursos.
 6. P adquiere A y luego B; más tarde libera ambos. Cuando Q continúe su ejecución podrá adquirir ambos recursos.

10

Fundamentos de interbloqueo

Diagrama de progreso conjunto. (cont.)

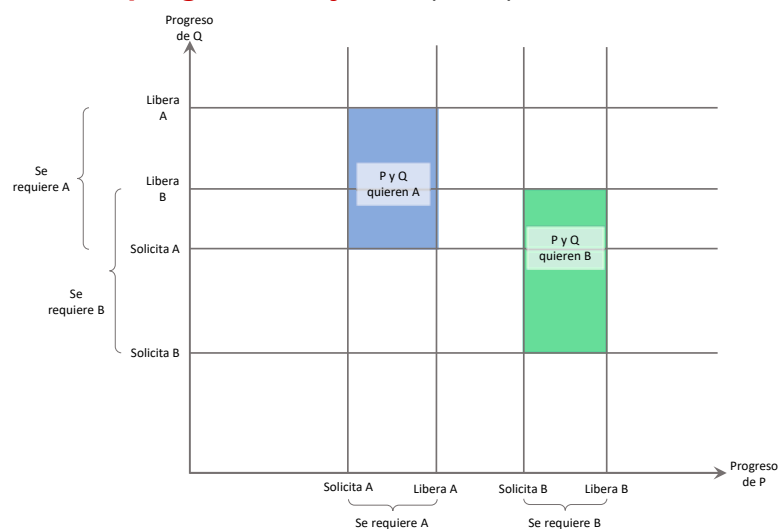
- Las aparición de un interbloqueo depende tanto de la dinámica de ejecución como de los detalles de la aplicación. Por ejemplo, si P no necesitase ambos recursos a la vez, de modo que los procesos tienen la siguiente estructura:

Proceso P	Proceso Q
...	...
Solicita A	Solicita B
...	...
Libera A	Solicita A
...	...
Solicita B	Libera B
...	...
Libera B	Libera A
...	...

11

Fundamentos de interbloqueo

Diagrama de progreso conjunto. (cont.)



12

Fundamentos de interbloqueo

Diagrama de progreso conjunto. (cont.)

- En la figura anterior hay dos zonas en las que uno de los procesos quedará bloqueado hasta que el otro libere los recursos.
- Sin embargo, independientemente de la temporización de los dos procesos, no puede ocurrir interbloqueo, ya que no existe una región fatal.
- Esto pone en evidencia lo dicho anteriormente: la aparición de abrazo mortal o interbloqueo depende de la dinámica de ejecución y de los detalles de la aplicación.
- El diagrama de progreso conjunto sirve para dos procesos. Para más procesos, se pueden agregar dimensiones y los principios de las regiones fatales e interbloqueos permanecen iguales.

13

Fundamentos de interbloqueo

Tipos de recursos.

- Pueden distinguirse dos categorías de recursos:
 - **Reutilizables.**
 - **Consumibles.**
- **Recursos reutilizables.**
- Son aquellos que sólo pueden ser utilizados de forma segura por un proceso a la vez, y que **no se destruyen después de su uso.**
- Los procesos obtienen unidades del recurso que más tarde liberarán para que puedan volver a usarlas otros procesos.
- Ejemplos de recursos reutilizables son:
 - Procesadores.
 - Memoria principal y secundaria.
 - Canales de E/S.
 - Archivos, bases de datos.

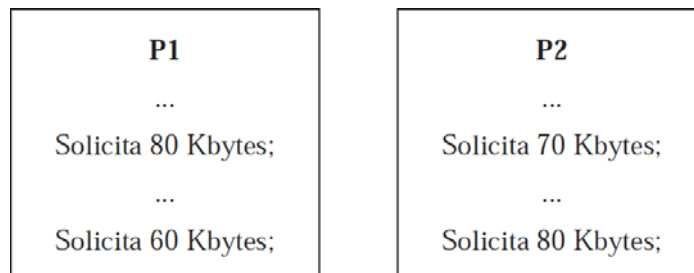
14

Fundamentos de interbloqueo

Tipos de recursos. (cont.)

▪ **Recursos reutilizables.** (cont.)

- Un ejemplo de abrazo mortal con recursos reutilizables es cuando se hacen peticiones de reserva de memoria principal.
- Supongamos que el espacio disponible es de 200 KB, y se produce la siguiente secuencia de peticiones:



15

Fundamentos de interbloqueo

Tipos de recursos. (cont.)

▪ **Recursos reutilizables.** (cont.)

- El abrazo mortal sucede si ambos procesos avanzan hasta su segunda petición. Si no se conoce anticipadamente la cantidad de memoria que se va a solicitar, es difícil tratar este tipo de interbloqueos mediante restricciones en el diseño del sistema.
- Un error en el diseño de los programas suele ser la causa más común por la que se producen los interbloqueos.
- Este problema puntualmente, se puede evitar utilizando un esquema de memoria virtual como vimos en Sistemas Operativos I.

16

Fundamentos de interbloqueo

Tipos de recursos. (cont.)

▪ **Recursos consumibles.**

- Un recurso consumible es aquél que puede **crearse (producirse)** y **destruirse (consumirse)**.
- Un proceso productor puede crear un número ilimitado de estos recursos. Cuando un procesos consumidor adquiere un recurso, éste **deja de existir**.
- Ejemplos de recursos consumibles son:
 - Las interrupciones.
 - Los mensajes.
 - La información en buffers de E/S.

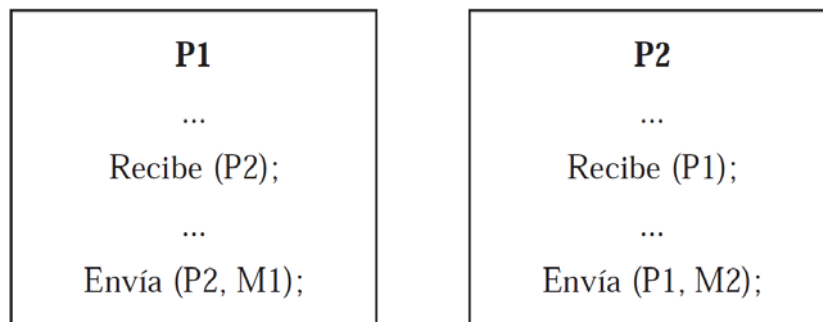
17

Fundamentos de interbloqueo

Tipos de recursos. (cont.)

▪ **Recursos consumibles.** (cont.)

- Un abrazo mortal que involucra recursos consumibles puede darse cuando tenemos dos procesos, y cada uno intenta recibir un mensaje del otro, y a continuación le envía un mensaje.



18

Fundamentos de interbloqueo

Tipos de recursos. (cont.)

▪ Recursos consumibles. (cont.)

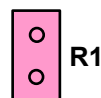
- El interbloqueo se produce si la función de recepción es bloqueante, es decir, el proceso se bloquea hasta que recibe el mensaje.
- Nuevamente, un error de diseño es la causa del abrazo mortal, y suelen ser muy difíciles de detectar. Incluso puede pasar que un proceso ejecute durante años hasta que se produzca el abrazo mortal.

19

Fundamentos de interbloqueo

Caracterización de la asignación de recursos.

- Una herramienta útil para representar la asignación de recursos a los procesos es el **grafo de asignación de recursos**.
- Consiste en un grafo dirigido que representa el estado del sistema en un determinado instante del tiempo.
- Los procesos y los recursos serán los nodos del grafo.
 - Los **procesos** se representan con **círculos**.
 - Los **recursos** se representan con **cuadrados o rectángulos**.
 - En el interior de cada recurso, se muestra **un punto por cada instancia** de ese recurso.

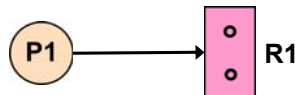


20

Fundamentos de interbloqueo

Caracterización de la asignación de recursos. (cont.)

- Los arcos unen los procesos con los recursos, y el sentido que tienen representará una situación distinta:
 - De un proceso a un recurso indica que el proceso solicitó el recurso, pero aún no le fue asignado, y por lo tanto se encuentra bloqueado a la espera de que le sea asignado.



- De un recurso a un proceso indica que el recurso ha sido asignado al proceso, y por lo tanto, está en su posesión. Si el recurso tiene más de una instancia, la arista irá desde una de ellas hacia el proceso.



21

Fundamentos de interbloqueo

Caracterización de la asignación de recursos. (cont.)

- Como todo grafo, se pueden expresar también en función de sus conjuntos de nodos y arcos:
 - $P = \{P_1, P_2, \dots, P_n\}$ será el conjunto de procesos.
 - $R = \{R_1, R_2, \dots, R_n\}$ será el conjunto de recursos.
 - $E = \{P_i \rightarrow R_j \mid R_j \rightarrow P_i, \dots, P_k \rightarrow R_m \mid R_m \rightarrow P_k\}$ será el conjunto de arcos.
- Nótese que al ser un grafo dirigido, los arcos se deben representar el sentido que tienen.
- El grafo de asignación de recursos se utiliza en la detección de abrazo mortal, que veremos más adelante.

22

Fundamentos de interbloqueo

Condiciones para el interbloqueo.

- Deben presentarse tres condiciones de gestión para que sea posible un interbloqueo:
 - **Exclusión mutua.** Sólo un proceso puede utilizar un recurso en cada momento. Ningún proceso puede acceder a una unidad de un recurso que se ha asignado a otro.
 - **Retención y espera.** Un proceso puede mantener los recursos asignados mientras espera la asignación de otros recursos.
 - **Sin expropiación.** No se puede forzar la expropiación de un recurso a un proceso que lo posee.
- Por diversos motivos estas condiciones son deseables. Por ejemplo, se necesita la exclusión mutua para la integridad de la información en una base de datos.

23

Fundamentos de interbloqueo

Condiciones para el interbloqueo. (cont.)

- Si se cumplen estas tres condiciones se puede producir un abrazo mortal, pero aún así puede que no lo haya. Para que realmente se produzca, se requiere una cuarta condición:
 - **Espera circular.** Existe una lista cerrada de procesos, de tal manera que cada uno posee al menos un recurso necesitado por el siguiente proceso de la lista.
- Las tres primeras condiciones son necesarias pero no suficientes para que haya un interbloqueo. La espera circular es, de hecho, una consecuencia de estas, y es irresoluble porque se cumplen las otras tres condiciones.
- Por lo tanto, las cuatro condiciones de manera conjunta forman las condiciones necesarias y suficientes para el abrazo mortal.

24

Fundamentos de interbloqueo

Condiciones para el interbloqueo. (cont.)

- Para clarificar esto, volvamos al diagrama de progreso conjunto, en donde la región fatal existe si se cumplen las primeras tres condiciones.
- Para que se produzca el abrazo mortal, además de la región fatal, debe existir una secuencia de peticiones de recursos que conduzcan a ella.
- Si se cumple la condición de espera circular, se ha entrado en la región fatal. Por lo tanto, las cuatro condiciones son suficientes para que se produzca el interbloqueo.

25

Fundamentos de interbloqueo

- El tratamiento del abrazo mortal o interbloqueo se verá en tres partes:
 - **Detección**, utilizando grafos como herramientas.
 - **Recuperación**, únicamente con técnicas “drásticas”.
 - **Prevención**, teniendo en cuenta las cuatro condiciones vistas anteriormente.

26

Detección del abrazo mortal

Grafos de asignación de recursos.

- Un grafo de asignación de recursos permite describir formalmente una situación de abrazo mortal.
- Existen tres operaciones representadas en el grafo:
 - **Solicitud** de un recurso: se indica con un arco desde un proceso a un recurso. El proceso permanece bloqueado siempre que tenga una solicitud.
 - **Asignación** de un recurso: indicada con un arco que va desde un recurso a un proceso. Si no tiene solicitudes, el proceso está listo para ejecutar.
 - **Liberación** de un recurso: simplemente, desaparece el arco del grafo.
- Cabe recordar que **el grafo es una instantánea del sistema** en un determinado instante de tiempo.

27

Detección del abrazo mortal

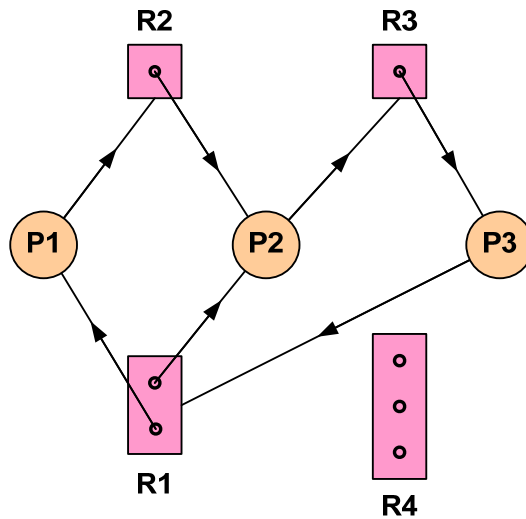
Grafos de asignación de recursos. (cont.)

- Se puede demostrar que si el grafo **no contiene ningún ciclo**, el sistema **no se encuentra en estado de abrazo mortal**.
- Si por el contrario, el grafo posee **al menos un ciclo**, **es posible que exista** un abrazo mortal.
- Un ciclo representa el cumplimiento de las tres primeras condiciones para haya un interbloqueo, y la cuarta condición es que **ese ciclo sea una espera circular irresoluble**, en cuyo caso **habrá una situación de abrazo mortal**.
- En la figura siguiente, se muestra un ejemplo en el que el grafo de asignación de recursos representa una situación de abrazo mortal.

28

Detección del abrazo mortal

Grafos de asignación de recursos. (cont.)



29

Detección del abrazo mortal

Grafos de asignación de recursos. (cont.)

- Se pueden observar dos ciclos:
 - $P1 \rightarrow R2 \rightarrow P2 \rightarrow R3 \rightarrow P3 \rightarrow R1 \rightarrow P1$
 - $P2 \rightarrow R3 \rightarrow P3 \rightarrow R1 \rightarrow P2$
- Se tiene que tanto P1, P2 y P3 sí están en estado de abrazo mortal porque:
 - P2 está esperando por R3, que está en posesión de P3.
 - P3 está esperando por un recurso de la clase R1 que están en posesión de P1 y P2.
 - Finalmente, P1 está esperando por R2 que, a su vez, está en posesión de P2.
- Estas esperas circulares son irresolubles, a pesar de que R1 cuenta con dos instancias; al estar ambas asignadas, éste recurso no está disponible para ningún otro proceso.

30

Detección del abrazo mortal

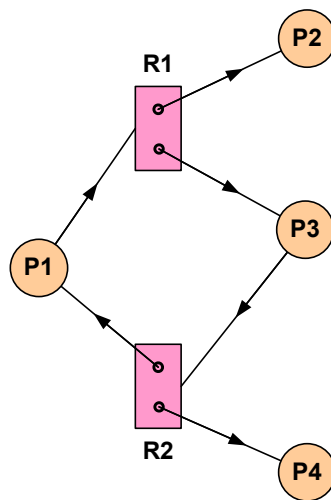
Grafos de asignación de recursos. (cont.)

- En la figura siguiente se muestra un ejemplo en el que el grafo no representa una situación de abrazo mortal.
- Se tiene el ciclo:
 - $P1 \rightarrow R1 \rightarrow P3 \rightarrow R2 \rightarrow P1$
- Sin embargo, no se tiene una situación de abrazo mortal, ya que, cuando P4 libere a R2 y P2 libere a R1, se rompe el ciclo en el grafo.
- Obsérvese además, que tanto P4 como P2 no están bloqueados, y pueden liberar sin condicionamientos a sus respectivos recursos.

31

Detección del abrazo mortal

Grafos de asignación de recursos. (cont.)

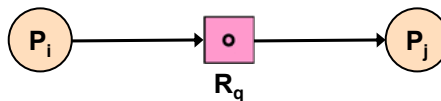


32

Detección del abrazo mortal

Grafos *wait-for*.

- Un caso particular es cuando todos los tipos de recursos analizados tienen una única instancia.
- Se puede construir un nuevo tipo de grafo en el que se elimina la representación de los recursos, llamado **grafo *wait-for***.
- Un arco en un grafo *wait-for* existe sí y solo sí existen 2 arcos en un grafo de asignación de recursos del tipo:
 - $P_i \rightarrow R_q$
 - $R_q \rightarrow P_j$



33

Detección del abrazo mortal

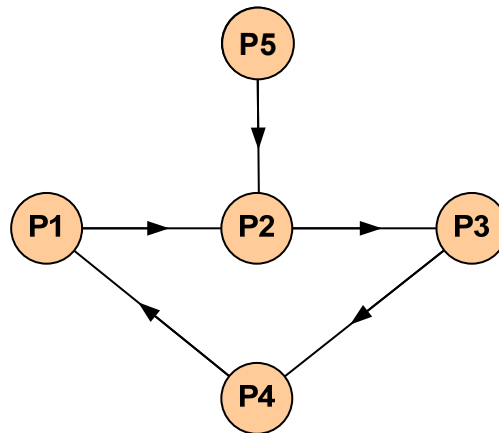
Grafos *wait-for*. (cont.)

- En el grafo *wait-for* esta situación se indica como:
 - $P_i \rightarrow P_j$
- Esto implica que el proceso P_i está esperando que el proceso P_j libere el recurso que tiene actualmente en posesión.
- Entonces, habrá una situación de abrazo mortal cuando **existe un ciclo en el grafo *wait-for***. Para este tipo de grafo, ésta es una condición necesaria y suficiente para que haya interbloqueo.
- En la figura siguiente, se muestra un ejemplo de grafo *wait-for* que representa una situación de abrazo mortal.

34

Detección del abrazo mortal

Grafos *wait-for*. (cont.)



35

Detección del abrazo mortal

Grafos *wait-for*. (cont.)

- El sistema operativo debe construir y mantener un grafo *wait-for* de los procesos, y periódicamente invocar un algoritmo que busque si existe un ciclo en dicho grafo.
- Se debe determinar el “momento adecuado” en que el algoritmo debe ser invocado, típicamente:
 - Cuando a un proceso no se le ha otorgado el recurso solicitado en un dado tiempo.
 - En intervalos periódicos de tiempo, por ejemplo, cada 30 minutos.
 - Cuando la utilización promedio del procesador baja de un cierto umbral.

36

Recuperación del abrazo mortal

- Se mencionó anteriormente que no hay una solución eficiente para el caso general de abrazo mortal. Una vez detectado, **salir de ese estado implica medidas necesariamente drásticas**.
- Existen básicamente tres opciones para la recuperación de una situación de abrazo mortal:
 - **Por medio de apropiación.**
 - **A través del retroceso.**
 - **A través de la eliminación de procesos.**
- Analizaremos en detalle cada una de estas opciones.

37

Recuperación del abrazo mortal

Recuperación por medio de apropiación.

- En algunos casos puede ser posible quitar temporalmente un recurso a su propietario actual y otorgarlo a otro proceso. En muchos casos se puede requerir intervención manual.
- La habilidad de **quitar un recurso a un proceso**, hacer que otro proceso lo utilice y después regresarlo sin que el proceso lo note, **depende** en gran parte **de la naturaleza del recurso**.
- Esta opción sólo es factible, hasta cierto punto, para los recursos reutilizables.
- Con frecuencia es difícil o imposible recuperarse de esta manera. Elegir el proceso a suspender depende en gran parte de cuáles procesos tienen recursos que se pueden quitar con facilidad.

38

Recuperación del abrazo mortal

Recuperación a través del retroceso.

- Si los diseñadores de sistemas saben que es probable que haya interbloqueos, pueden hacer que los procesos realicen **puntos de comprobación** en forma periódica.
- Realizar puntos de comprobación en un proceso significa que su estado se escribe en un archivo para poder reiniciarlo más tarde.
- El punto de comprobación no sólo contiene la imagen de la memoria, sino también el estado del recurso; en otras palabras, qué recursos están asignados al proceso en un momento dado.
- Para que sean más efectivos, los nuevos puntos de comprobación no deben sobrescribir a los anteriores, sino que deben escribirse en nuevos archivos, para que se acumule una secuencia completa a medida que el proceso se ejecute.

39

Recuperación del abrazo mortal

Recuperación a través del retroceso. (cont.)

- Para realizar la recuperación, un proceso que posee un recurso necesario se revierte a un punto en el tiempo **antes de que haya adquirido ese recurso**, para lo cual se inicia uno de sus puntos de comprobación anteriores.
- Se pierde todo el trabajo realizado desde el punto de comprobación. En efecto, el proceso se restablece a un momento anterior en el que no tenía el recurso, que ahora se asigna a uno de los procesos en interbloqueo.
- Si el proceso reiniciado trata de adquirir el recurso de nuevo, tendrá que esperar hasta que vuelva a estar disponible.

40

Recuperación del abrazo mortal

Recuperación a través de la eliminación de procesos.

- La forma más cruda y simple de romper un interbloqueo es **eliminar uno o más procesos**.
- Una posibilidad es eliminar a uno de los procesos en el ciclo. Con un poco de suerte, los demás procesos podrán continuar. Si esto no ayuda, se puede repetir hasta que se rompa el ciclo.
- De manera alternativa, se puede elegir como “víctima” a un proceso que no esté en el ciclo, para poder liberar sus recursos.
- En este método, el proceso a eliminar se elige con cuidado, debido a que está conteniendo recursos que necesita cierto proceso en el ciclo.

41

Recuperación del abrazo mortal

Recuperación a través de la eliminación de procesos. (cont.)

- Por ejemplo, un proceso podría contener una impresora y querer un trazador, en donde otro proceso podría contener un trazador y querer una impresora. Estos dos procesos están en interbloqueo.
- Un tercer proceso podría contener otra impresora idéntica y otro trazador idéntico, y estar felizmente en ejecución. Al eliminar el tercer proceso se liberarán estos recursos y se romperá el interbloqueo que involucra a los primeros dos.
- Donde sea posible, es mejor eliminar un proceso que se pueda volver a ejecutar desde el principio sin efectos dañinos.
- Por ejemplo, una compilación siempre podrá volver a ejecutarse, ya que todo lo que hace es leer un archivo de código fuente y producir un archivo de código objeto.

42

Prevención del abrazo mortal

- La estrategia de prevención del abrazo mortal consiste, de manera simplificada, en diseñar un sistema de manera que se excluya la posibilidad de interbloqueo.
- Se pueden clasificar los métodos de prevención en dos categorías:
 - Un **método indirecto**, que consiste en impedir la aparición de una de las tres condiciones necesarias para que haya interbloqueo (las tres primeras).
 - Un **método directo**, que impide que se produzca una espera circular (cuarta condición).
- A continuación veremos las técnicas relacionadas con cada una de las cuatro condiciones.

43

Prevención del abrazo mortal

Exclusión mutua.

- En general, la primera de las condiciones no puede eliminarse. Si el acceso a un recurso requiere exclusión mutua, el sistema operativo debe proporcionarlo.
- Algunos recursos, tales como los archivos, pueden permitir múltiples accesos de lectura, pero acceso exclusivo sólo para escritura.
- Incluso en este caso, puede haber abrazo mortal si más de un proceso requiere permiso de escritura.

44

Prevención del abrazo mortal

Retención y espera.

- Esta condición puede eliminarse estableciendo que un proceso debe pedir de una sola vez todos los recursos que va a necesitar, y bloqueándolo hasta que les puedan ser asignados.
- Esto tiene varios inconvenientes:
 - Por un lado, un proceso puede quedarse esperando mucho tiempo hasta que todas las solicitudes de recursos se satisfagan, pudiendo haber ejecutado con sólo algunos de los que necesita.
 - Por otro lado, los recursos asignados a un proceso pueden permanecer no disponibles, impidiendo que otros procesos los ocupen.
 - Además, un proceso puede no conocer por anticipado todos los recursos que va a necesitar.
- En caso de procesos multihilos, hay que tener en cuenta todos los recursos que también puede llegar a necesitar cada hilo.

45

Prevención del abrazo mortal

Sin expropiación.

- Existen varias maneras de impedir esta condición:
 - Si a un proceso que mantiene varios recursos se le deniega una petición posterior, ese proceso deberá liberar sus recursos originales, y si es necesario, los solicitará de nuevo junto con el recurso adicional.
 - Alternativamente, si un proceso solicita un recurso que otro proceso mantiene actualmente, el sistema operativo puede expropiar al segundo proceso y obligarle a liberar sus recursos.
- Este último esquema impediría el abrazo mortal sólo si no hay dos procesos con la misma prioridad.
- Estas estrategias son aplicables únicamente para aquellos recursos que tienen la posibilidad de salvar sus estados y restaurarlos nuevamente más tarde, como por ejemplo, los procesadores.

46

Prevención del abrazo mortal

Espera circular.

- Se puede impedir definiendo un orden lineal entre los distintos tipos de recursos. Si a un proceso le han asignado recursos de tipo R , posteriormente sólo puede pedir aquellos cuyo tipo tenga un orden posterior a R .
- Se puede asociar un índice a cada tipo de recurso, de modo que R_i precede a R_j en la ordenación si $i < j$. Si tenemos dos procesos, A y B, y el proceso A adquirió R_i y solicitó R_j , si el proceso B adquirió R_j , no podrá solicitar R_i por la condición de ordenamiento, con lo que se evita la situación de abrazo mortal.
- Tal como en el caso de la retención y espera, la prevención de la espera circular puede ser ineficiente, ya que puede ralentizar los procesos y denegar innecesariamente el acceso a un recurso.

47

Fin del Módulo II

48