

PRÁCTICA 6 DE SISTEMAS OPERATIVOS

TEMA: Interbloqueos

Nombre: Fernando Eliceo Huilca Villagómez

Carrera: Ingeniería de Software

Grupo: GR1SW

Fecha: 08 / 05 / 2024

Índice de Contenidos

1. OBJETIVOS	2
2. INFORME	2
3. CONCLUSIONES Y RECOMENDACIONES	14
4. BIBLIOGRAFÍA	14

Índice de Imágenes

Ilustración 1 Resultado de Show Deadlocked Processes.	3
Ilustración 2 Asignación de Recursos en un Sistema de Gestión de Procesos.....	4
Ilustración 3 Disponibilidad de Recursos.	5
Ilustración 4 Solución de Interbloqueos habilitando Disallow Hold and Wait.	6
Ilustración 5 Solución de Interbloqueos habilitando Disallow Hold and Wait y Disallow Circular Wait	8
Ilustración 6 Solución de Interbloqueos con todos los Campos Activados.	10
Ilustración 7 Problemas de Interbloqueo con Cuatro Procesos.....	11
Ilustración 8 Solución Interbloqueo Activado Disallow Circular Wait.	12
Ilustración 9 Solución Interbloqueo Activado todas las Opciones con Cuatro Procesos.	13

1. OBJETIVOS

- 1.1 Utilizar el simulador CPU Simulator para crear condiciones de bloqueo (deadlock).
- 1.2 Implementar métodos para prevenir condiciones de bloqueo.

2. INFORME

Para comprender el tema de Interbloqueos, se utiliza un simulador que permite observar los problemas que surgen y explorar diversas soluciones. Una vez completada la instalación del simulador siguiendo los pasos adecuados, se puede comenzar con la práctica.

En la imagen se muestra una simulación de la asignación de recursos en un sistema de gestión de procesos, que ha resultado en una situación de interbloqueo (deadlock). Los nodos amarillos (P1, P2, P3, P4) representan procesos, mientras que los nodos azules (R0, R1, R2, R3) representan recursos.

Las flechas indican la relación entre procesos y recursos:

- Una flecha de un recurso a un proceso indica que el recurso está asignado a ese proceso.
 - Una flecha de un proceso a un recurso indica que el proceso está solicitando ese recurso.
- Aunque se indica que no hay procesos interbloqueados ("There are no deadlocked processes"), el registro de eventos muestra un mensaje de interbloqueo detectado.

La imagen presenta una simulación en la que los procesos y recursos se representan en un grafo y un panel de control. Se ha detectado un interbloqueo, donde los procesos están esperando recursos que están siendo utilizados por otros procesos, creando un ciclo de dependencia sin solución inmediata. Los controles permiten gestionar la asignación de recursos y tomar acciones para prevenir o resolver el interbloqueo. Además, se pueden ajustar parámetros de simulación para estudiar diferentes escenarios y estrategias de manejo de interbloqueos.

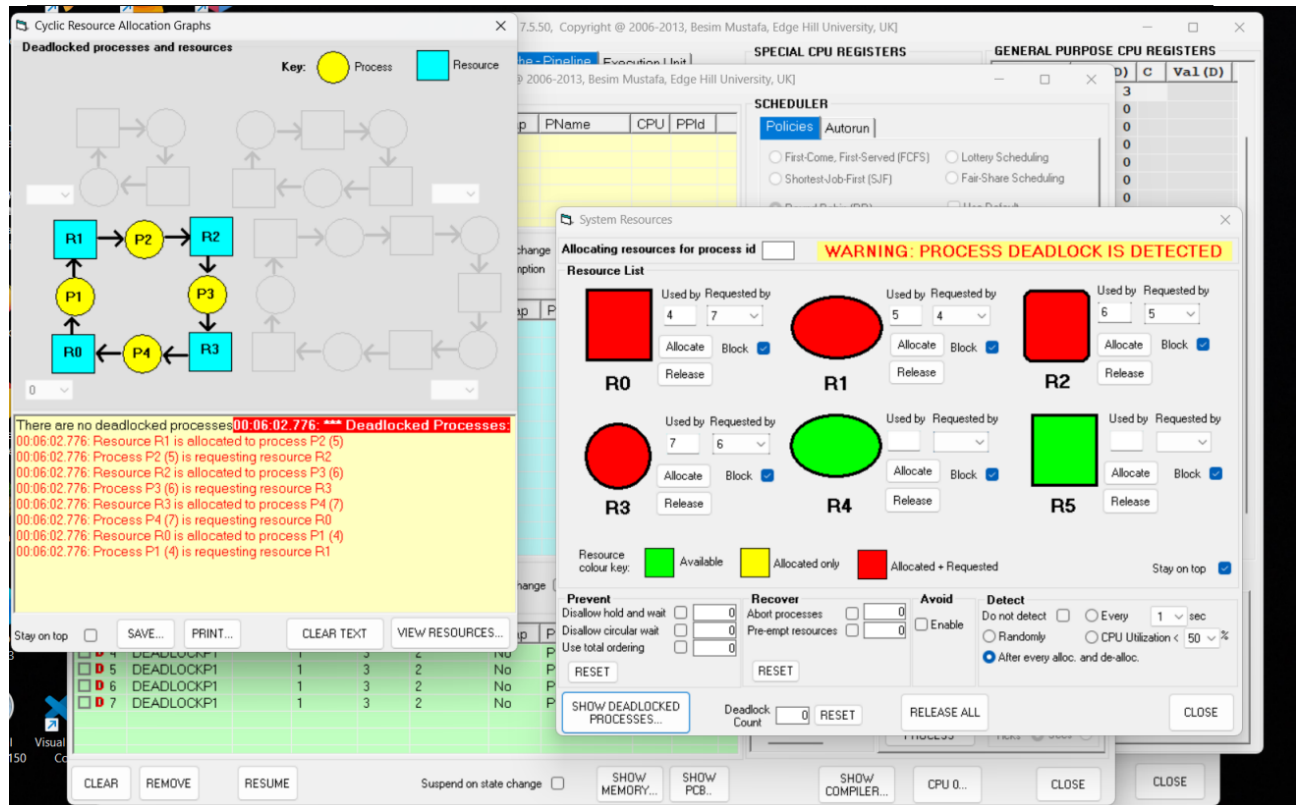


Ilustración 1 Resultado de Show Deadlocked Processes.

Se presenta otra instancia de una simulación de asignación de recursos en un sistema de gestión de procesos, similar a la anterior pero con algunas diferencias significativas. En esta sección del grafo, no se muestran relaciones activas entre procesos y recursos. Los nodos y las flechas están vacíos, lo que sugiere que no hay interacciones actuales en el grafo.

Los recursos están representados con círculos de colores:

- Rojos indican que el recurso está asignado y solicitado simultáneamente.
- Verdes indican que el recurso está disponible.
- Amarillos indican que el recurso está solo asignado.

La imagen muestra una simulación donde actualmente no hay procesos interbloqueados. Los recursos R0, R1, R2 y R3 están en uso y también solicitados por otros procesos, mientras que R4 y R5 están disponibles. No se muestra ninguna interacción actual en el grafo de asignación de recursos. Las opciones de prevención y recuperación de interbloqueos están disponibles para gestionar el sistema y evitar posibles interbloqueos en el futuro. Además, los controles permiten ajustar parámetros y simular diferentes escenarios para estudiar la eficacia de las estrategias de manejo de interbloqueos.



Ilustración 2 Asignación de Recursos en un Sistema de Gestión de Procesos.

Se muestra otra instancia de la simulación de asignación de recursos en un sistema operativo, pero en esta ocasión todos los recursos parecen estar disponibles y no hay procesos en ejecución o en espera. El mensaje en la parte inferior dice "There are no deadlocked processes", indicando que no hay interbloqueos en este momento. Los recursos R0, R1, R2, R3, R4 y R5 están todos disponibles, indicados en verde.

No hay uso ni solicitudes de recursos por parte de procesos. Las opciones como la deshabilitación de espera y retención, la espera circular y el uso de ordenación total para evitar interbloqueos están presentes.

La imagen muestra un estado donde no hay procesos en ejecución, listos o en espera. Todos los recursos están disponibles y no hay ninguna interacción entre procesos y recursos. Tampoco hay interbloqueos detectados. Las opciones de prevención y recuperación de interbloqueos están disponibles, pero actualmente no hay necesidad de utilizarlas, ya que el sistema está en un estado inactivo sin procesos que gestionen recursos. Además, este escenario permite estudiar la configuración y las estrategias de prevención sin la presión de un sistema activo.

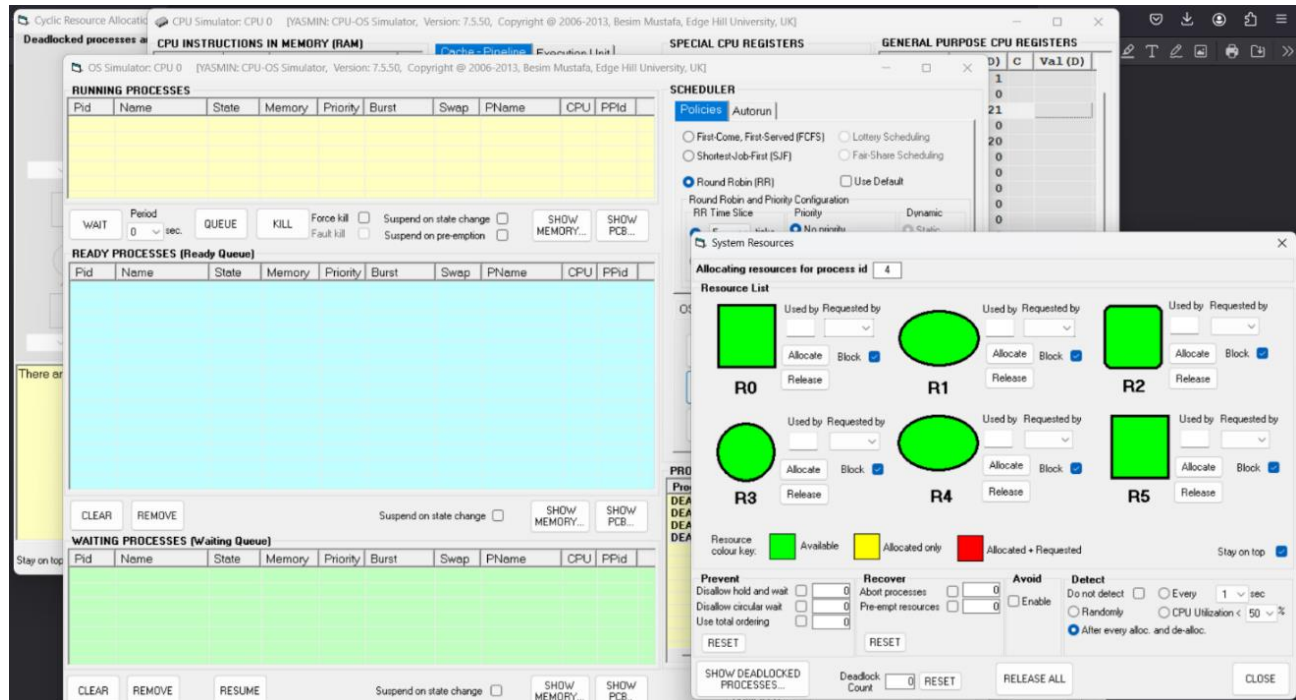


Ilustración 3 Disponibilidad de Recursos.

Se muestra una interfaz de simulación de sistema operativo, específicamente el "CPU-OS Simulator" desarrollado por Besim Mustafa en Edge Hill University. Vamos a analizar los elementos principales:

Procesos en ejecución:

- Hay un proceso llamado "DEADLOCKP2" en estado de ejecución con PID 7.

Procesos listos:

- Hay cuatro procesos en la cola de listos, todos con nombres que empiezan por "DEADLOCK" (P3, P4, P5, P6).

Recursos del sistema:

- Se muestra una ventana de "System Resources" que ilustra el estado de 6 recursos (R0 a R5). La mayoría de estos recursos están disponibles (indicados en verde), lo que sugiere que no están siendo utilizados en este momento.

Gestión de deadlocks:

- Hay opciones para prevenir, recuperar y evitar deadlocks. Está activada la opción "Disallow hold and wait" con un valor de 12.

Estado de deadlock:

- Hay un botón "SHOW DEADLOCKED PROCESSES", lo que indica que el sistema está simulando o detectando situaciones de interbloqueo (deadlock).

Controles de simulación:

- Existen botones para controlar la simulación como "WAIT", "QUEUE", "KILL", entre otros.

La situación que se está simulando parece ser un escenario de deadlock, donde múltiples procesos están compitiendo por recursos, lo que podría llevar a una situación de interbloqueo. El simulador permite experimentar con diferentes estrategias para manejar y prevenir deadlocks en un entorno de sistema operativo. Esto ofrece una oportunidad para analizar cómo las políticas de asignación de recursos y la gestión de procesos pueden influir en la ocurrencia de interbloqueos y la eficiencia general del sistema.

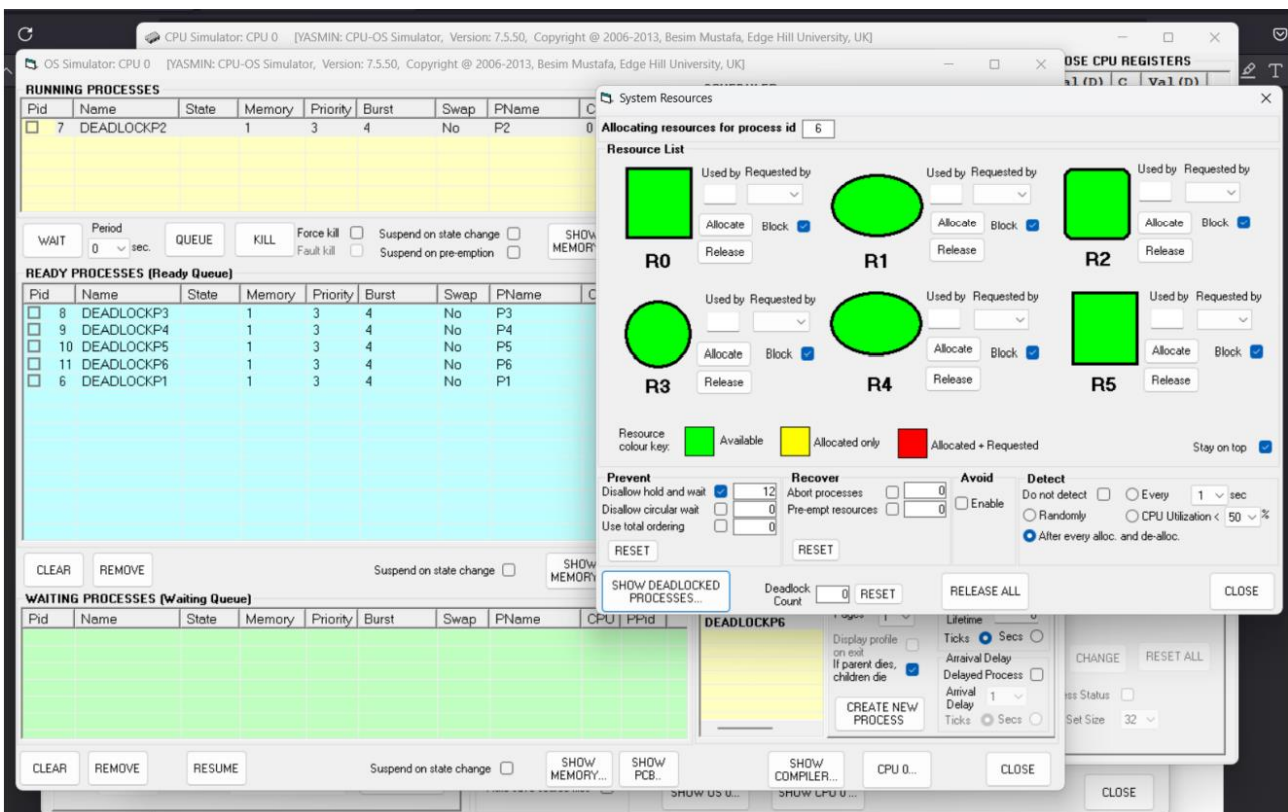


Ilustración 4 Solución de Interbloqueos habilitando Disallow Hold and Wait.

Procesos en ejecución:

- Hay un proceso llamado "DEADLOCKP3" con PID 8 en estado de ejecución.

Procesos listos:

- En la cola de procesos listos (Ready Queue) hay 4 procesos:
 - DEADLOCKP4 (PID 10)
 - DEADLOCKP5 (PID 11)
 - DEADLOCKP1 (PID 6)
 - DEADLOCKP2 (PID 7)

Recursos del sistema:

- Se muestra una ventana de "System Resources" con 6 recursos (R0 a R5). Todos los recursos están en verde, lo que indica que están disponibles y no están siendo utilizados por ningún proceso.

Gestión de deadlocks:

- En la sección "Prevent", la opción "Disallow hold and wait" está activada con un valor de 36. Esto sugiere que se está implementando una estrategia para prevenir deadlocks.

Estado de deadlock:

- Hay un botón "SHOW DEADLOCKED PROCESSES", pero el contador de deadlocks muestra 0, lo que indica que actualmente no hay deadlocks detectados.

Controles de simulación:

- Se observan varios botones para controlar la simulación como "WAIT", "QUEUE", "KILL", entre otros.

Creación de procesos:

- Hay un botón "CREATE NEW PROCESS" visible, lo que permite añadir nuevos procesos a la simulación.

La situación que se está simulando parece ser un escenario donde múltiples procesos (todos con nombres relacionados con "DEADLOCK") están compitiendo por recursos. Sin embargo, en este momento, todos los recursos están disponibles y no se ha detectado ningún deadlock. El simulador permite experimentar con diferentes estrategias de gestión de recursos y procesos para prevenir y resolver posibles interbloqueos, proporcionando una plataforma para el análisis y la optimización del manejo de deadlocks en sistemas operativos.

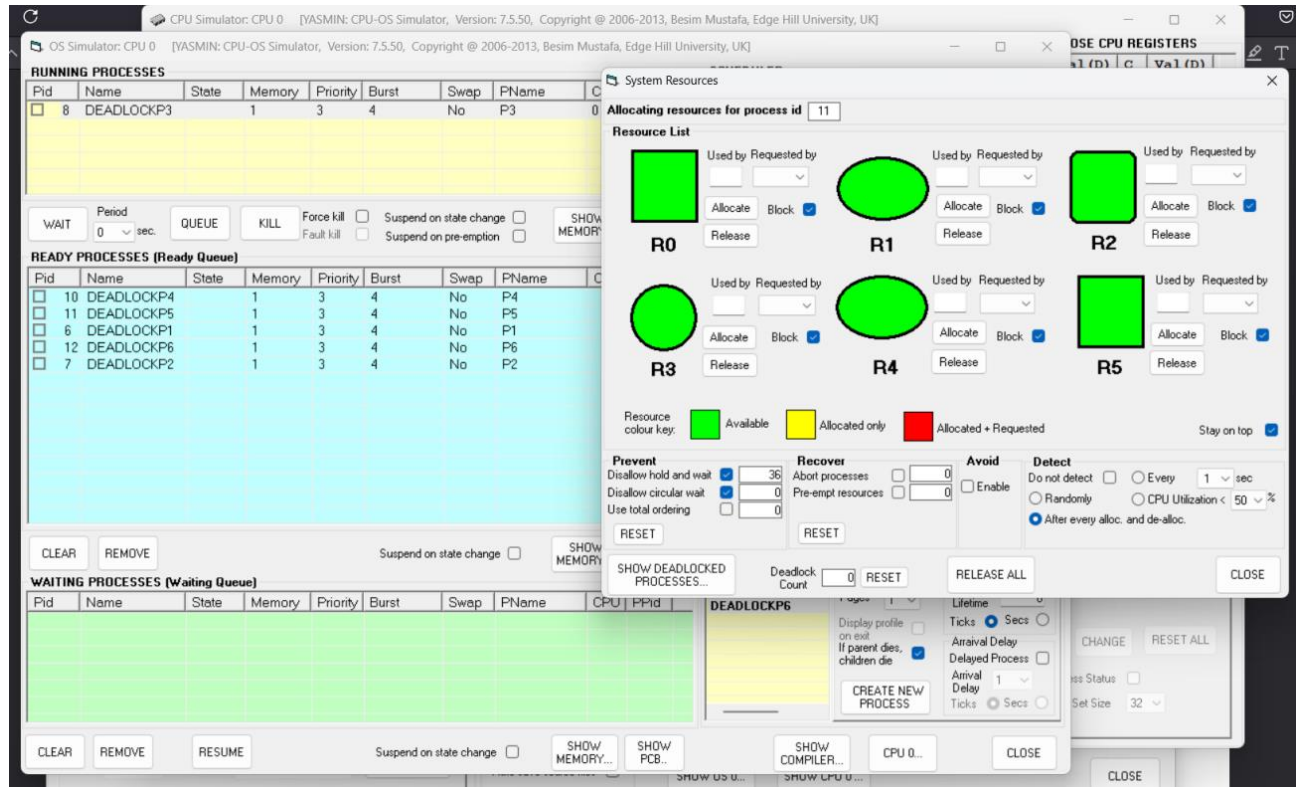


Ilustración 5 Solución de Interbloqueos habilitando Disallow Hold and Wait y Disallow Circular Wait

En esta imagen se muestra una simulación más avanzada del sistema operativo usando el "CPU-OS Simulator". Vamos a analizar los elementos clave:

Procesos en ejecución:

- No hay procesos en ejecución en este momento.

Procesos listos:

- En la cola de procesos listos (Ready Queue) hay 3 procesos:
 - DEADLOCKP4 (PID 9)
 - DEADLOCKP5 (PID 10)
 - DEADLOCKP6 (PID 11)

Procesos en espera:

- En la cola de procesos en espera (Waiting Queue) hay 3 procesos:
 - DEADLOCKP1 (PID 6)
 - DEADLOCKP2 (PID 7)

- DEADLOCKP3 (PID 8)

Recursos del sistema:

- En la ventana "System Resources":
 - R0, R1, R2, y R3 están disponibles (indicados en verde).
 - R4 y R5 están asignados pero no solicitados (indicados en amarillo).
 - R4 está siendo usado por el proceso 10.
 - R5 está siendo usado por el proceso 11.

Gestión de deadlocks:

- "Disallow hold and wait" está activado con un valor de 42.
- "Disallow circular wait" está activado.
- "Use total ordering" está activado con un valor de 0.

Estado de deadlock:

- El contador de deadlocks muestra 0, indicando que no se han detectado deadlocks.

Configuración de detección:

- La detección de deadlocks está configurada para realizarse "After every alloc. and dealloc." (después de cada asignación y desasignación).

Esta simulación muestra un escenario donde varios procesos están en la cola de espera y listos, pero actualmente no hay procesos en ejecución. Los recursos del sistema están en su mayoría disponibles, aunque dos están asignados pero no solicitados. Las estrategias de gestión de deadlocks están activadas para prevenir situaciones de interbloqueo, y el sistema está configurado para detectar deadlocks después de cada operación de asignación o desasignación. Esto proporciona un entorno controlado para experimentar y analizar cómo diferentes políticas y configuraciones afectan la ocurrencia y resolución de deadlocks en un sistema operativo.

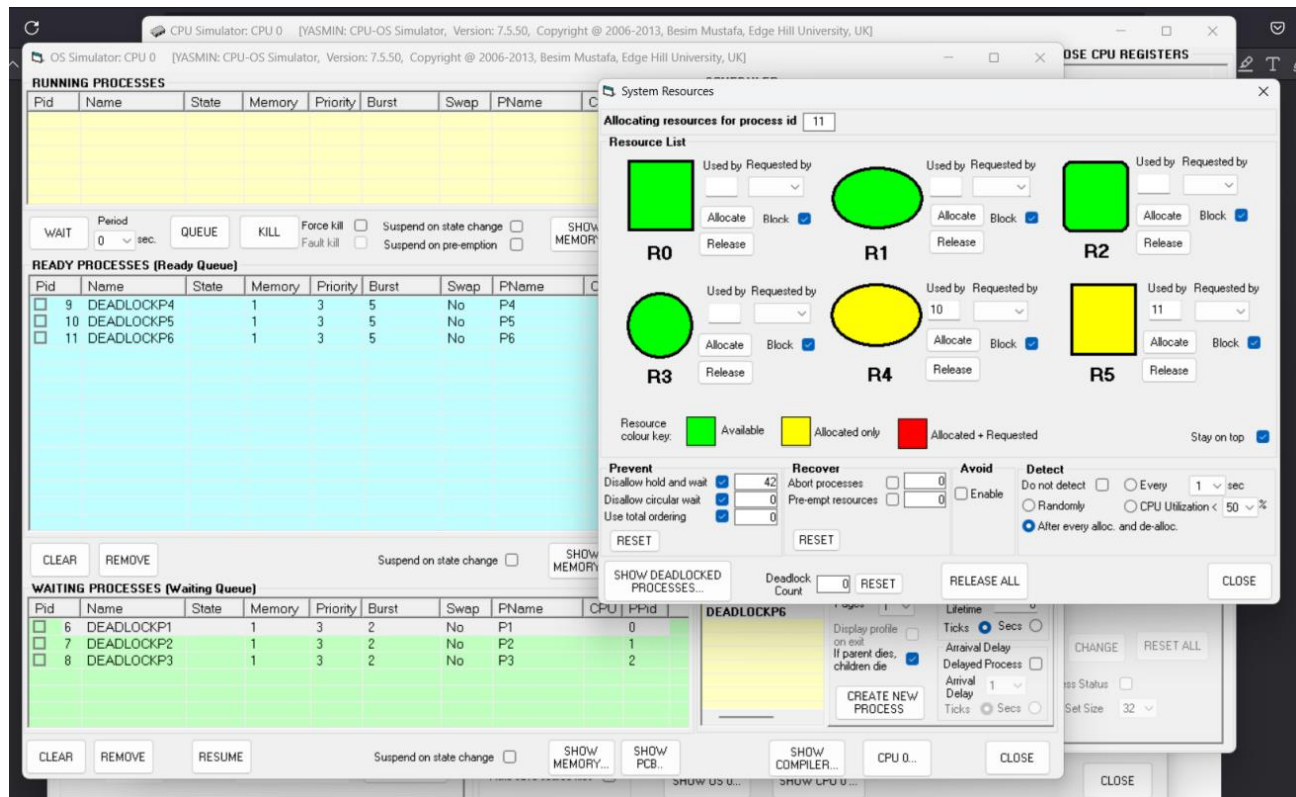


Ilustración 6 Solución de Interbloqueos con todos los Campos Activados.

Esta imagen muestra una interfaz más detallada del CPU-OS Simulator. Vamos a analizar los elementos principales:

Recursos del sistema:

- Se muestran 6 recursos (R0 a R5), todos en verde, lo que indica que están disponibles y no asignados.

Gestión de deadlocks:

- En la sección "Prevent":
 - "Disallow hold and wait" está activado con un valor de 16.
 - Las opciones "Disallow circular wait" y "Use total ordering" están desactivadas.
- La detección de deadlocks está configurada para realizarse "After every alloc. and dealloc." (después de cada asignación y desasignación).
- El contador de deadlocks muestra 0, indicando que no hay deadlocks detectados.

Planificador (Scheduler):

- La configuración del planificador se ve parcialmente:

- El tiempo de quantum (Time Slice) está configurado a 5 ticks.
- La prioridad está configurada como "No priority".

Procesos:

- Se pueden ver parcialmente algunos procesos en la lista (LOCKP1, LOCKP2, LOCKP3, LOCKP4).
- Hay una ventana para crear un nuevo proceso (P5) con una prioridad de 3 y 1 página de memoria.

Controles adicionales:

- Hay botones para ver diferentes aspectos del sistema como la utilización de CPU, lista de procesos, estados de procesos, recursos, memoria, entre otros.

Prevención de deadlocks:

- La única estrategia activa es "Disallow hold and wait", lo que sugiere que se está utilizando para prevenir deadlocks.

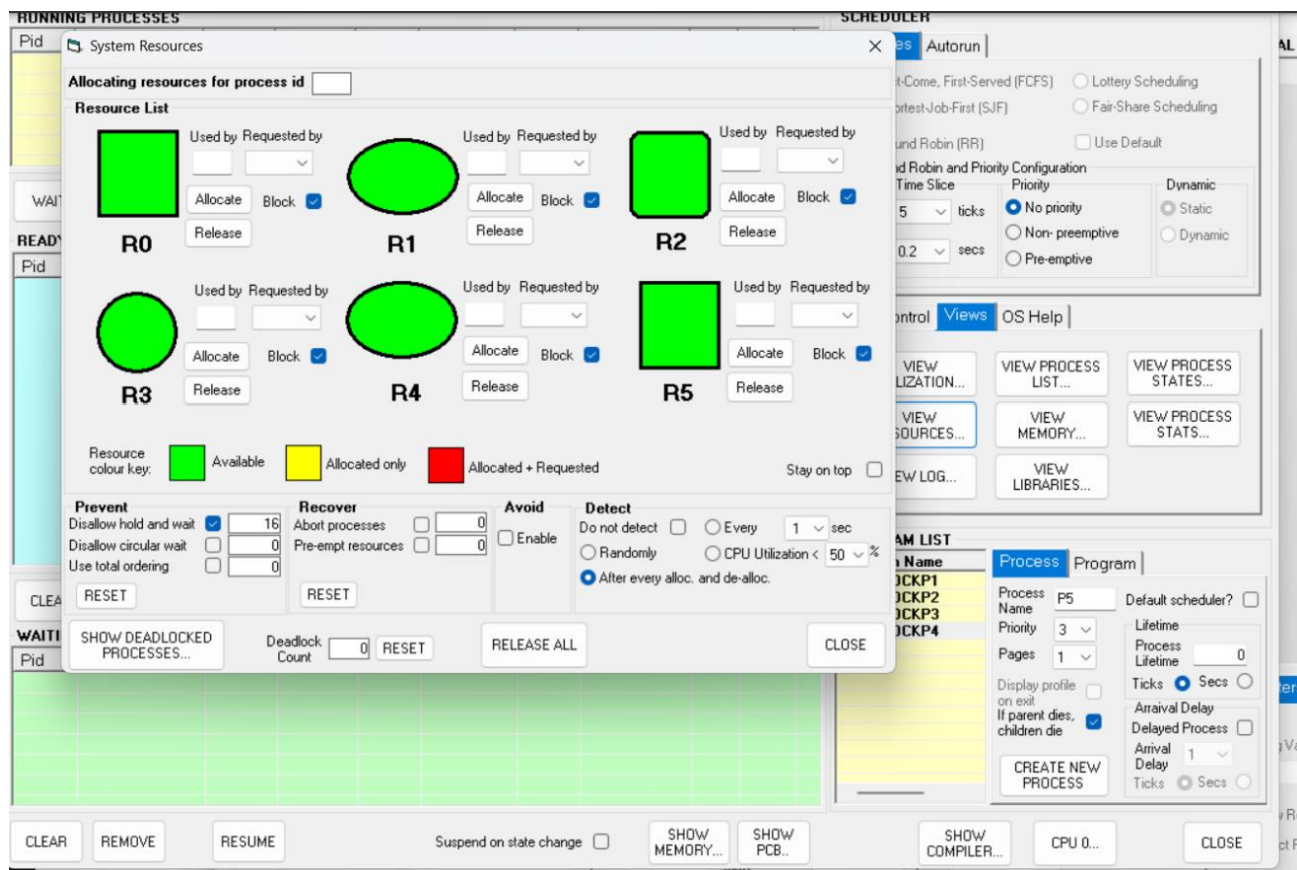


Ilustración 7 Problemas de Interbloqueo con Cuatro Procesos.

En el gráfico de la interfaz del CPU-OS Simulator:

- **Flechas entre procesos y recursos:**
 - Una flecha de un proceso a un recurso indica que el proceso está solicitando ese recurso.
 - Una flecha de un recurso a un proceso indica que el recurso ha sido asignado a ese proceso.
- **Detección de Bloqueos:**
 - La sección "DEADLOCK" permite detectar y recuperar de bloqueos, asegurando que los procesos no queden atascados esperando recursos indefinidamente.
- **Gestión de Recursos:**
 - Los botones y menús permiten al usuario ejecutar, pausar, reiniciar y priorizar procesos, así como asignar y liberar recursos.

Este gráfico es una herramienta valiosa para quienes estudian o trabajan en campos relacionados con la informática, particularmente en el diseño y análisis de sistemas operativos, al proporcionar una visualización clara de cómo se gestionan los recursos y se detectan los bloqueos en el sistema.

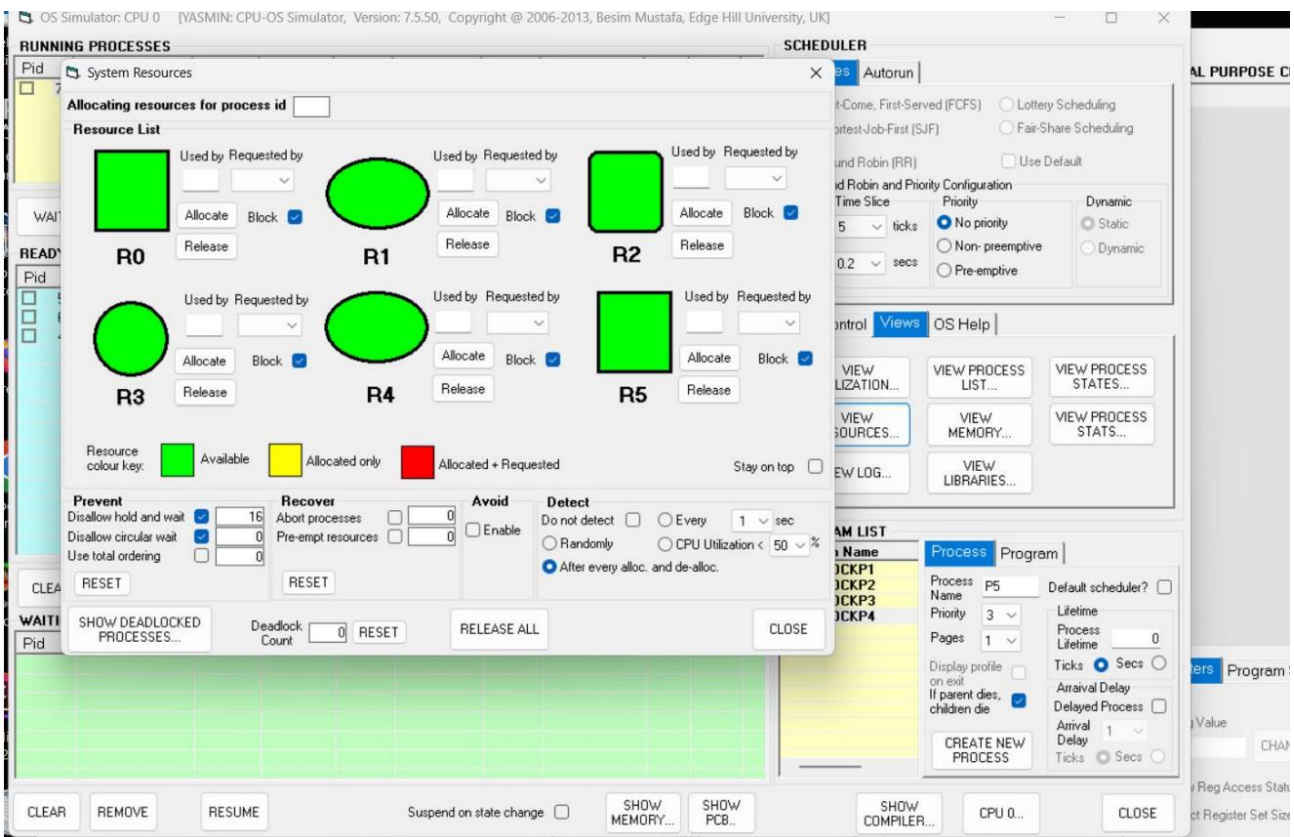


Ilustración 8 Solución Interbloqueo Activado Disallow Circular Wait.

La imagen muestra una interfaz de una 'Línea de Ensamblaje Automatizada' con varias herramientas y opciones:

- **Opciones de control:**
 - **‘Start’ (Iniciar):** Permite iniciar el proceso de la línea de ensamblaje.
 - **‘Stop’ (Detener):** Detiene el proceso en curso.
 - **‘E-STOP’ (Parada de emergencia):** Ofrece una opción de parada inmediata en caso de emergencia.
- **Editor de código o depurador:**
 - Incluye líneas de código y opciones como **‘Program Select’ (Seleccionar Programa)** y **‘Watch’ (Ver)** para examinar y depurar el código en ejecución.
- **Aplicación tipo bloc de notas:**
 - Enumera varios propósitos para los registros de la CPU en la programación en lenguaje ensamblador, proporcionando información relevante sobre cómo se utilizan estos registros.

Esta imagen ilustra la complejidad de la gestión de procesos y recursos en sistemas informáticos, mostrando cómo el software especializado puede facilitar la supervisión y el control de estos elementos para garantizar un funcionamiento eficiente. Proporciona una visión detallada de cómo se manejan los procesos automatizados y la depuración del código en entornos de programación avanzada

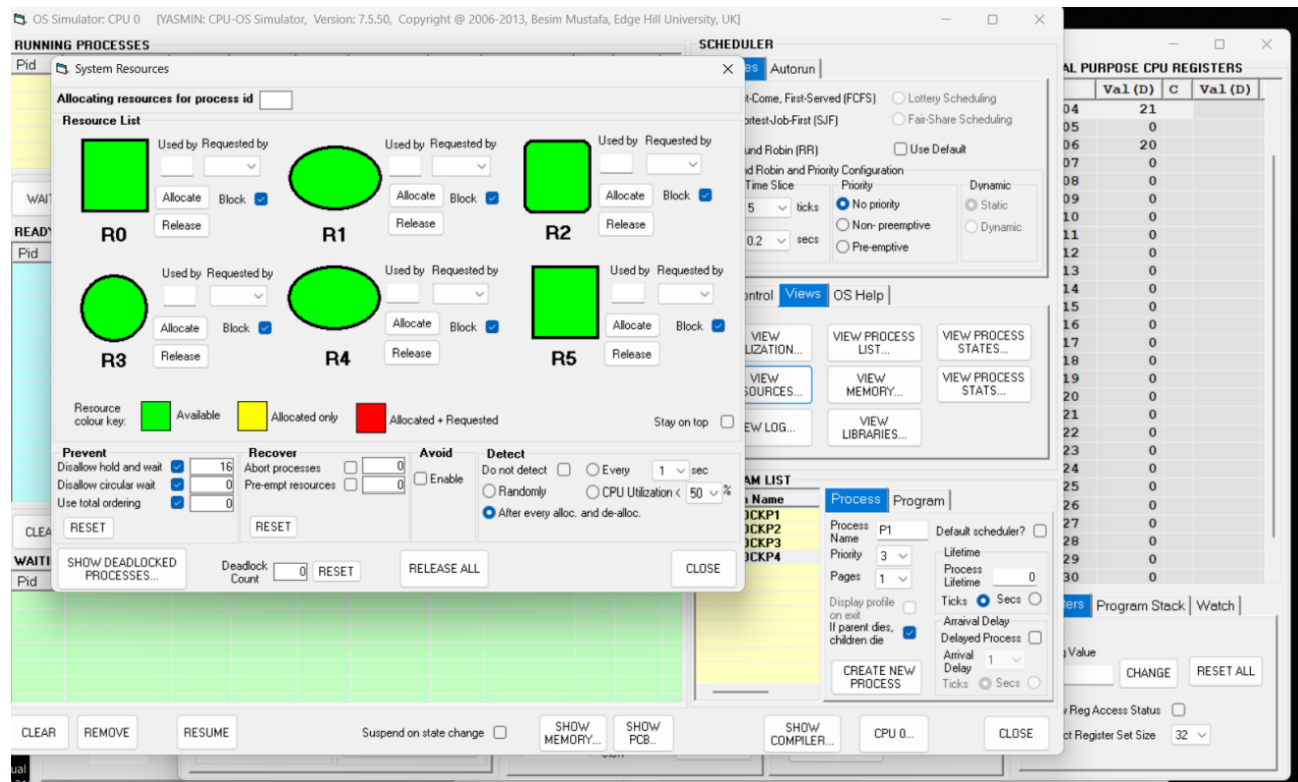


Ilustración 9 Solución Interbloqueo Activado todas las Opciones con Cuatro Procesos.

3. CONCLUSIONES Y RECOMENDACIONES

Conclusiones:

- La interfaz del CPU-OS Simulator permite una visualización clara de la asignación y el estado de los recursos, facilitando la identificación de procesos que solicitan o tienen recursos asignados. Esto es crucial para la gestión efectiva de recursos y la prevención de deadlocks.
- La capacidad para activar y configurar estrategias como "Disallow hold and wait" y la detección de deadlocks en tiempo real ayuda a evitar situaciones en las que los procesos se quedan esperando recursos indefinidamente. La configuración para detectar deadlocks después de cada asignación o desasignación refuerza esta capacidad.
- Las herramientas como los botones de control ("Start", "Stop", "E-STOP") y el editor de código/depurador proporcionan un entorno interactivo para la gestión y depuración de procesos. La aplicación tipo bloc de notas que describe los propósitos de los registros de la CPU es útil para entender y optimizar la programación en ensamblador.

Recomendaciones:

- Considerar la activación de opciones adicionales como "Disallow circular wait" y "Use total ordering" para una prevención más robusta de deadlocks. Estas estrategias pueden proporcionar una protección adicional contra situaciones de interbloqueo.
- Revisar y ajustar la configuración del planificador según las necesidades específicas del sistema. Implementar prioridades más dinámicas y revisar el tiempo de quantum puede mejorar el rendimiento y la eficiencia del sistema.
- Aprovechar el editor de código y las herramientas de depuración para optimizar el código en ensamblador y resolver posibles problemas antes de la implementación en un entorno real. Esto puede reducir errores y mejorar la estabilidad del sistema.

4. BIBLIOGRAFÍA

- [1] A. Silberschatz, H. Korth, and S. Sudarshan, *Database System Concepts*, 7th ed. McGraw-Hill, 2019.
- [2] C. A. Sturman and R. S. Greiner, "Deadlock Avoidance in Distributed Systems," *IEEE Transactions on Software Engineering*, vol. SE-9, no. 1, pp. 10-18, Jan. 1983.