SISTEMAS OPERATIVOS

PROGRAMA 5. ALGORITMO DE PLANIFICACIÓN RR (ROUND-ROBIN)

Profesor:
VIOLETA DEL ROCIO BECERRA VELAZQUEZ

VERDUZCO ROSALES LUIS ENRIQUE
223992388
Ingenieria en Computación

26/10/25

CUCEI DIVTIC D04

ÍNDICE

Objetivo de la actividad

En esta práctica, se mejoró el simulador de planificación de procesos para implementar el algoritmo Round-Robin (RR). El objetivo principal era adaptar la lógica del programa anterior (FCFS) para que ahora siga las reglas de planificación con ráfagas de tiempo, determinadas por un valor Quantum definido al iniciar la simulación por el usuario. El programa sigue administrando los estados de un proceso (nuevo, listo, ejecución, bloqueado y terminado) y sus transiciones, pero ahora un proceso puede ser expulsado del procesador si agota su Quantum, siendo colocado al final de la cola de listos para garantizar un reparto más equitativo del CPU.

Lenguaje Utilizado

Para la realización de esta práctica se utilizó JavaScript, junto con HTML para la estructura y CSS para los estilos. Elegí JavaScript porque permite crear un entorno más visual e interactivo que se ejecuta directamente en un navegador web, facilitando la visualización en tiempo real de cómo cambian de estado los procesos y las colas. Esto es especialmente útil para el algoritmo Round-Robin, ya que permite observar en tiempo real cómo los procesos rotan entre el estado de ejecución y la cola de listos, haciendo muy evidente el efecto del Quantum.

Soluciones Implementadas

La base del simulador sigue siendo un ciclo principal llamado `setInterval` que se ejecuta cada segundo para simular el avance del tiempo en el sistema.

Características clave realizadas para implementar Round-Robin:

1. **Entrada del Quantum:**
   Al inicio de la simulación, además de preguntar por el número de procesos iniciales, se añadió un campo para que el usuario ingrese el valor del Quantum. Este valor se almacena en una variable global que se utiliza durante toda la ejecución para determinar las ráfagas de tiempo.

2. **Estructura del Proceso (BCP):**
   El objeto de JavaScript que representa a cada proceso fue actualizado. Además de los campos que ya tenía como id, tme y los diferentes tiempos (tiempoLlegada, tiempoServicio, etc.), se añadió una nueva variable: tiempoEnQuantum. Este contador es clave para el algoritmo, ya que registra cuántos segundos ha estado el proceso en ejecución durante su turno actual.

3. **Estados y Colas:**
   La gestión de los estados se mantuvo igual, utilizando arreglos de JavaScript para las colas de nuevos, listos, bloqueados y terminados, y una variable procesoEnEjecucion para el proceso en el CPU. La diferencia fundamental está en cómo se gestiona la cola de listos.

4. **Lógica del Ciclo Principal (Round-Robin):**

La función mainEjecucion fue modificada para incluir la lógica de Round-Robin. En cada segundo, además de las verificaciones anteriores, ahora se realiza lo siguiente:

- Procesa el trabajo:
  Si hay un proceso en ejecución, se incrementa su tiempoServicio y también su contador tiempoEnQuantum.

- Verificación de fin de proceso:
  Primero, se comprueba si el proceso ha completado su TME (tiempoServicio >= tme). Si es así, termina normalmente.

- Verificación de fin de Quantum:
  Si el proceso no ha terminado, se comprueba si ha agotado su ráfaga de tiempo (tiempoEnQuantum >= quantum). Si esto ocurre, el proceso es retirado del CPU, su tiempoEnQuantum se reinicia a cero, y es enviado al final de la cola de listos usando el método listos.push(). El procesador queda libre para el siguiente proceso.

- Despacho:
  Si el CPU está libre y hay procesos esperando, el despachador sigue tomando el primer elemento de la cola de listos con listos.shift(), manteniendo el orden de llegada para el siguiente turno.

5. **Implementación de Teclas y Visualización:**
   Las teclas de control ('N', 'E', 'W', 'P', 'C', 'B') conservan la misma funcionalidad que en la práctica anterior. En la interfaz de usuario, se añadió un nuevo apartado visual para mostrar el valor del Quantum definido, y en la sección del "Proceso en Ejecución", ahora también se muestra el tiempo que el proceso ha consumido de su Quantum actual, permitiendo ver claramente cuándo será interrumpido.

Conclusión

Esta práctica me ayudó a comprender las diferencias entre un algoritmo no apropiativo como FCFS y uno apropiativo como Round-Robin. Implementar la lógica del Quantum me permitió ver de forma práctica cómo un sistema operativo puede ofrecer tiempos de respuesta más justos y evitar que un proceso largo monopolice el CPU. Lo más difícil fue integrar el contador del Quantum en el ciclo principal sin afectar los otros cálculos de tiempo y logrando que los procesos fueran movidos correctamente entre el estado de ejecución y la cola de listos, ahora me queda más claro su funcionamiento e importancia.

Código Fuente: https://github.com/UnluckyLuren/SisOperativos.git

Ejecutable: https://unluckyluren.github.io/SisOperativos/