Introducción

El código presentado corresponde a un sistema de simulación que gestiona y visualiza la planificación de procesos en un entorno de CPU. Este sistema utiliza una interfaz gráfica para representar diversas métricas de rendimiento y estados de los procesos, permitiendo a los usuarios interactuar con los datos de manera intuitiva.

Estructura del Código

1. **Clases y Paquetes**:
    - El sistema está organizado en varias clases, cada una responsable de diferentes aspectos de la simulación. Las clases principales incluyen:
        - MainWindow: Controla la interfaz principal y la configuración del sistema.
        - CPU: Representa las unidades de procesamiento y su estado.
        - Proceso: Modela los procesos que se ejecutan en las CPUs.
        - Grafico Torta y Gráfico Barras Tiempo: Se encargan de la visualización de datos mediante gráficos.
        - Cola: Implementa una estructura de datos para manejar las colas de procesos.
        - Nodo: Representa un nodo en la lista enlazada que almacena procesos.
        - Manejo Txt: Maneja la lectura de configuraciones desde un archivo.
2. **Interfaz Gráfica**:
    - Se utiliza Java Swing para crear la interfaz gráfica. La clase MainWindow inicializa varios componentes, como paneles para cada CPU, contenedores para las colas de procesos (listos, bloqueados, terminados) y gráficos para visualizar el rendimiento.
    - Los usuarios pueden crear procesos nuevos, cambiar la política de planificación y ajustar la configuración del sistema a través de la interfaz.
3. **Manejo de Procesos**:
    - El sistema permite la creación de procesos con diferentes características (CPU-bound o I/O-bound) y los gestiona en colas según la política de planificación seleccionada (FCFS, Round Robin, SPN, SRT, HRRN).
    - Cada proceso tiene atributos como nombre, cantidad de instrucciones, estado y ciclos de CPU, lo que permite simular su ejecución y manejo de excepciones.

4. **Estructura de Cola**:
    - La clase Cola implementa una cola genérica utilizando nodos, permitiendo operaciones como encolar (enqueue), desencolar (dequeue), y buscar procesos no tomados. Esto es fundamental para gestionar el flujo de procesos en el sistema.
    - La clase Nodo almacena el dato del proceso y una referencia al siguiente nodo, facilitando la creación de una lista enlazada.
5. **Visualización de Datos**:
    - Los gráficos creados con JFreeChart permiten visualizar la distribución de procesos y el rendimiento a través de gráficos de torta y de barras, actualizándose dinámicamente para reflejar el estado actual del sistema.
6. **Gestión de Configuraciones**:
    - La clase Manejotxt se encarga de leer configuraciones desde un archivo de texto, permitiendo que el sistema se inicie con configuraciones específicas para el número de CPUs activas y la duración de los ciclos de ejecución.

Funcionalidad General

- **Creación de Procesos**: Los usuarios pueden ingresar datos para crear nuevos procesos, que se encolan para su ejecución.
- **Ejecución Simulada**: La simulación de la ejecución de procesos se maneja a través de hilos que representan las CPUs, permitiendo observar cómo se gestionan los procesos en tiempo real.
- **Actualización de Estadísticas**: Se actualizan constantemente las métricas de rendimiento, permitiendo a los usuarios observar el comportamiento del sistema bajo diferentes condiciones de carga.
- **Configuraciones Interactivas**: El sistema permite a los usuarios ajustar configuraciones como el número de CPUs activas y la duración de los ciclos de ejecución, lo que impacta directamente en la simulación.

Conclusiones

Este sistema de simulación proporciona una herramienta efectiva para estudiar y entender los conceptos de planificación de procesos en sistemas operativos. La interfaz gráfica facilita la interacción y la visualización de datos, mientras que la lógica subyacente permite simular comportamientos complejos de la CPU y los procesos. La implementación de la clase Cola, junto con la estructura de Nodo, es crucial para el

manejo eficiente de los procesos y su estado en el sistema. Además, la clase Manejotxt permite configurar el sistema de manera flexible. Esta implementación puede servir como base para estudios más avanzados en el área de sistemas operativos y gestión de recursos computacionales.