Planificación de Procesos Utilizando Algoritmos FCFS, RR, MLQ y MLFQ

Estudiante

Marco Antonio Riascos Salguero

Profesor:
Jefferson Peña Torres

Sistemas Operativos

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería y ciencias
Programa de Ingeniería de Sistemas y computación

Santiago de Cali

6 de septiembre de 2024

1. Introducción al Programa

El código es un simulador de planificación de procesos que implementa diferentes algoritmos de planificación en sistemas operativos. Las políticas incluidas son:

- **FCFS (First-Come, First-Served)**: Los procesos se ejecutan en el orden en que llegan.
- **Round Robin (RR)**: Cada proceso recibe una pequeña cantidad de tiempo (quantum) y luego se alternan.
- **MLQ (Multi-Level Queue)**: Los procesos se clasifican en múltiples colas basadas en prioridades.
- **MLFQ (Multi-Level Feedback Queue)**: Similar a MLQ, pero los procesos pueden moverse entre colas según su comportamiento.

El input del programa proviene de un archivo de texto que contiene los detalles de cada proceso (ID, tiempo de llegada, tiempo de ráfaga, prioridad, y cola a la que pertenece). El objetivo es simular cómo se ejecutan los procesos bajo una política de planificación específica.

2. Clases y Estructuras Principales

Clase Process

La clase Process representa cada proceso en el sistema. Contiene los siguientes atributos:

- **id**: ID del proceso.
- **at (Arrival Time)**: Tiempo en que el proceso llega al sistema.
- **bt (Burst Time)**: Tiempo que el proceso necesita para ejecutarse.
- **ct (Completion Time)**: Tiempo en que el proceso finaliza.
- **rt (Response Time)**: Tiempo de respuesta en la primera ejecución.
- **wt (Waiting Time)**: Tiempo que el proceso pasa esperando en la cola antes de ejecutarse.
- **priority**: Prioridad del proceso.
- **queuelevel**: Cola a la que pertenece (importante para MLQ y MLFQ).
- **tat (Turnaround Time)**: Tiempo total desde la llegada del proceso hasta su finalización.

La clase tiene varias funciones **get** y **set** para acceder y modificar los atributos de los procesos.

Aquí tienes una explicación detallada del código que has proporcionado, el cual implementa un sistema de planificación de procesos con diferentes políticas como FCFS, Round Robin, MLQ y MLFQ:

3. Lectura del Input

La función readInput se encarga de leer el archivo que contiene la lista de procesos. Este archivo debe tener el siguiente formato:

1. Número de procesos.
2. Número de colas.
3. Política de planificación (FCFS, RR, MLQ o MLFQ).
4. Prioridad por defecto.
5. Detalles de cada proceso: ID, tiempo de llegada (AT), tiempo de ráfaga (BT), prioridad y el nivel de cola.

4. Algoritmos de Planificación

FCFS (First-Come, First-Served)

La función firstComeFirstServed implementa el algoritmo FCFS. Procesa los procesos en el orden en que llegan (primero en entrar, primero en salir). Calcula el **Completion Time (CT)**, el **Turnaround Time (TAT)** y el **Waiting Time (WT)** para cada proceso.

Round Robin (RR)

La función roundRobin implementa el algoritmo Round Robin, que asigna un quantum de tiempo a cada proceso. Si un proceso no termina dentro de su quantum, se mueve al final de la cola y se sigue ejecutando más tarde. El proceso continúa hasta que todos los procesos terminan.

MLQ (Multi-Level Queue)

En la política MLQ, los procesos están organizados en diferentes colas, y cada cola puede tener su propia política de planificación (en este caso, se utiliza FCFS). Los procesos son ejecutados según la prioridad de la cola.

MLFQ (Multi-Level Feedback Queue)

La función mlfq implementa la política MLFQ. En este enfoque, los procesos pueden moverse entre colas. Si un proceso no se completa en su quantum, se degrada a una cola de menor prioridad. Esto permite una mayor flexibilidad en la ejecución de los procesos.

5. Función Principal os

La función os es el corazón del sistema de planificación. Utiliza una lista de procesos y los organiza en diferentes colas basadas en su política de planificación:

1. **FCFS**: Los procesos se ejecutan en el orden en que llegan.
2. **Round Robin**: Se utiliza un quantum para alternar entre los procesos.
3. **MLFQ**: Los procesos pueden moverse entre colas si no terminan dentro de su quantum asignado.

Cada vez que el sistema tiene procesos listos para ejecutarse, la función revisa las colas y selecciona el siguiente proceso a ejecutar según la política seleccionada.

6. Ejecución del Programa

El programa comienza leyendo los detalles de los procesos desde un archivo usando la función readInput. Luego, según la política de planificación especificada en el archivo, ejecuta los procesos usando la función os. El contador de tiempo se incrementa a medida que los procesos se ejecutan, y al finalizar se imprimen los detalles de cada proceso, como el tiempo de finalización, turnaround y tiempo de espera.

7. Ejemplo de Input

Aquí se tiene un ejemplo de cómo podría verse el archivo input.txt:

4	Número de procesos
3	Número de colas
MLFQ	Política de planificación
1	Prioridad por defecto
1 0 10 2 1	Proceso 1: ID=1, AT=0, BT=10, Prioridad=2, Cola=1
2 2 5 1 2	Proceso 2: ID=2, AT=2, BT=5, Prioridad=1, Cola=2
3 4 8 3 1	Proceso 3: ID=3, AT=4, BT=8, Prioridad=3, Cola=1
4 6 6 2 3	Proceso 4: ID=4, AT=6, BT=6, Prioridad=2, Cola=3

8. Video explicativo

https://youtu.be/paDyq-M8YFo

Conclusión

Este código es un simulador de planificación de procesos que soporta múltiples políticas de planificación y ofrece una visión clara de cómo se ejecutan los procesos en diferentes colas. Está diseñado para ser flexible y puede manejar sistemas más complejos al permitir la planificación basada en múltiples niveles de prioridad, además de ofrecer la capacidad de mover procesos entre colas (en el caso de MLFQ).