

Programa 4 y Reporte 4

Simular el Proceso por Lotes

Salvador Castañeda Andrade
Sistemas Operativos



Introducción

Analizar el funcionamiento y la implementación del algoritmo de planificación Round Robin (RR) en un sistema de procesamiento por lotes simulado. El algoritmo RR es ampliamente utilizado en sistemas operativos para garantizar un reparto equitativo de la CPU entre los procesos, evitando el monopolio de recursos por parte de un proceso.

Desarrollo

El algoritmo de planificación Round Robin (RR) se implementa en el programa de simulación de procesamiento por lotes a través de la función `round_robin()`. A continuación, se presentan fragmentos de código clave que explican el funcionamiento del algoritmo RR:

1. Asignación de proceso actual y quantum:

Python

```
if not current_process and pending_processes:
    current_process = pending_processes.pop(0)
    current_process_number = len(finished_processes) + 1
    current_quantum = quantum
```

Este fragmento de código verifica si hay un proceso en ejecución (`current_process`) y si hay procesos pendientes en la cola (`pending_processes`). Si no hay un proceso en ejecución y hay procesos pendientes, se asigna el primer proceso pendiente a `current_process` y se inicializa el quantum con el valor especificado en la variable `quantum`.

2. Ejecución del proceso actual:

Python

```
if current_process:
    if current_quantum > 0:
        current_quantum -= 1
        current_process.max_time -= 1
    if current_process.max_time <= 0:
```

```
        current_process.finish_time = elapsed_seconds

        finished_processes.append(current_process)

        current_process = None
    else:

        pending_processes.insert(0, current_process)

        current_process = None
```

Este fragmento de código se encarga de ejecutar el proceso actual (`current_process`). Se reduce el quantum restante del proceso y el tiempo máximo de ejecución en cada iteración. Si el tiempo máximo de ejecución del proceso llega a cero, el proceso se considera finalizado y se agrega a la lista de procesos finalizados (`finished_processes`). Si el quantum actual llega a cero antes de que el proceso haya finalizado, el proceso se mueve al principio de la cola de procesos pendientes.

Estos fragmentos de código ilustran cómo se implementa el algoritmo de planificación Round Robin en el programa de simulación de procesamiento por lotes, asegurando un reparto equitativo de la CPU entre los procesos y evitando el bloqueo de procesos.



Conclusión

La implementación y análisis del algoritmo de planificación Round Robin (RR) en el contexto de un sistema de procesamiento por lotes simulado ha proporcionado una comprensión más profunda de cómo este algoritmo garantiza un reparto equitativo de la CPU entre los procesos. A través de la simulación, se ha demostrado que el algoritmo RR es efectivo para evitar el bloqueo de procesos y garantizar que todos los procesos reciban un tiempo justo de ejecución.

Una de las fortalezas clave del algoritmo RR es su simplicidad y equidad inherentes. Al asignar a cada proceso un quantum de tiempo de CPU y mover los procesos no finalizados al final de la cola, el algoritmo garantiza que ningún proceso monopolice la CPU y que todos los procesos tengan la oportunidad de ejecutarse.

En resumen, el algoritmo de planificación Round Robin es una herramienta valiosa en la gestión de procesos en sistemas operativos. Su capacidad para garantizar un reparto equitativo de la CPU y evitar el bloqueo de procesos lo convierte en una opción popular en una variedad de entornos informáticos. Sin embargo, es importante tener en cuenta las limitaciones del algoritmo y considerar otras opciones según las necesidades específicas del sistema.



Bibliografía

Bibliografía básica

Stallings, W. (2011), Operating Systems: Internals and Design Principles. Prentice Hall. 7th Edition.

Silberschatz, A., Galvin, P. B., & Gagne, G. (2010) Ingeniería del Software: Un enfoque práctico, McGraw Hill, México.

Bibliografía complementaria

Peters, James F. & Pedrycz, Witold (2008) Operating System Concepts. John Wiley & Sons Inc. 8th Edition.

Tanenbaum, A. S. (2008) Modern Operating Systems Pearson Educación, México.

