

UNIVERSIDAD DE GUADALAJARA

Centro Universitario de Ciencias Exactas e Ingenierías

Ingeniería en Computación

6^{to} semestre

Tema: Algoritmo de planificación RR (Round-Robin)



Materia:
Seminario de Sistemas Operativos

Sección:
D02

NRC:
103845

PRESENTA:
Saúl Alejandro Castañeda Pérez
Daniel Martínez Martínez

Docente:
Violeta del Rocio Becerra Velazquez

Fecha de entrega:
Domingo, 02 de Abril de 2023

Simulación del Algoritmo de planificación Round Robin

Objetivo

El programa de esta semana contempla la realización de mejoras al programa 4, además de la implementación del algoritmo Round Robin para atender a los procesos en estado de listo.

El algoritmo Round Robin funciona mediante la asignación de un pequeño intervalo de tiempo, llamado "quantum" a cada proceso en secuencia circular. Cuando el quantum de un proceso se agota, el sistema operativo interrumpe el proceso y lo pone de regreso en la cola de listos para el siguiente ciclo de asignación de quantum.

Este proceso de asignación de quantum a cada proceso se repite continuamente de manera cíclica, lo que garantiza que cada proceso tenga una oportunidad justa y equitativa de utilizar el procesador (entrar en estado de ejecución).

Desarrollo

Implementación del uso de Quantum: la implementación del Quantum se realiza a través de una variable contadora en la función encargada de la ejecución de los procesos. Esta variable se va decrementando constantemente hasta llegar a 0, lo cual indica que al proceso dentro del procesador ya se le acabó su tiempo de ejecución. Ante esto, el proceso que estaba en ejecución es enviado de regreso a su lugar dentro de la lista de procesos listos, teniendo reflejado el intervalo Quantum en su tiempo transcurrido, esto si se completó una ronda quantum con éxito, ya que podría no completarse porque el proceso tenía un tiempo restante menor al quantum, por que el proceso fue interrumpido, etc.

Este proceso se realiza hasta que todas las tareas son completadas.

CÓDIGO RELEVANTE

```
def ProcesarDatos(self,IDLote):
    Lotes[IDLote][10]='Ejecucion'
    QuantumAUX=-1
    for i in range(Lotes[IDLote][7],Lotes[IDLote][1]):
        QuantumAUX+=1
        if Quantum-QuantumAUX==0:
            return True
        if Bandera_Error:
            break
        if Bandera_P_C:
            while True:
                if not Bandera_P_C:
                    break
```

```

if BanderaInterrupcion:
    Lotes[IDLote][10]='Bloqueado'
    return True
self.lineEdit_11.setText('{}'.format(i+1))
self.lineEdit_12.setText('{}'.format(Lotes[IDLote][1]-i-1))
self.lineEdit_27.setText('{}'.format(Quantum-QuantumAUX-1))
self.IncrementoTiempoEspera(x)
self.IncrementoContadorGlobal()
self.DesbloquearProcesoBloqueado()
self.LlenarTablaBCP()
if Bandera_Error:
    Lotes[IDLote][9]='ERROR'
    Lotes[IDLote][10]='Finalizado'
    self.ProcesosFinalizados(Lotes[IDLote])
    self.LlenarTablaBCP()
return False

```

The interface displays the following components:

- Top Bar:** Contains window control icons (minimize, maximize, close) on the right and a 2x2 grid icon on the left.
- Nuevos:** A box showing the number 6.
- Listos Panel:** A table with columns ID, TME, and TT.

ID	TME	TT
2	11	5
3	6	5
4	9	5
- Bloqueados Panel:** A table with columns ID and TTB, currently empty.

ID	TTB
----	-----
- Proceso en ejecución Panel:** Displays details for process ID 1.

ID:	1
Operacion:	70-93
TME:	9
TT:	9
TR:	0
Quantum:	1
- Contador global:** A box showing the value 24.
- Terminados Panel:** A table with columns ID, Operacion, and Resultado, currently empty.

ID	Operacion	Resultado
----	-----------	-----------
- Continuar Button:** A rounded button at the bottom center.

Ejemplo de ejecución con un quantum de 5

Conclusiones

Esta tarea fue menos complicada que los dos programas anteriores, ya que muchas de las bases necesarias ya estaban establecidas y solo se necesitó hacer ajustes para implementar el algoritmo Round Robin. Sin embargo, lo más desafiante fue asegurarse de que no hubiera desfase entre los tiempos transcurridos y el tiempo actual del Quantum, debido a que una implementación inadecuada de los incrementos y decrementos en los tiempos transcurridos y en el quantum podría fácilmente desencadenar un incremento continuo en el error entre el valor esperado y el valor real en los contadores, algo que ante nuestros ojos puede pasar desapercibido o puede ser irrelevante.

En conclusión podemos decir que el programa cumple de manera satisfactoria con los requerimientos impuestos en la realización de este programa, siendo los requerimientos anteriores los más desafiantes de manera indirecta, debido a que tenemos que seguir implementando mejoras sin averiar alguna de las funciones que ya estaban previamente implementadas.

Con respecto a la naturaleza del algoritmo, podemos decir que el algoritmo funciona de una manera bastante eficiente, ya que permite la ejecución de los procesos de manera más equitativa, evitando que un proceso acapare el procesador por una gran cantidad de tiempo (cosa que sí podía suceder en el algoritmo FCFS), esto claro si se escoge un intervalo quantum adecuado, por que de no ser así, todas las tareas podrían pasar mucho tiempo sin completarse.