



*Universidad Nacional
Autónoma de México*

Facultad de Ingeniería

Sistemas operativos



Profesor(a):
Gunnar Wolf

Práctica II

Planificación por Retroalimentación Multinivel y por Lotería

ROJAS TERRAZAS LAYLET

RUÍZ SÁNCHEZ MIGUEL ÁNGEL

GRUPO: 6

Planificador de Lotería

```
import random
import queue
# Creamos los procesos
num_procesos = random.randint(5, 8)
procesos = []

for i in range(num_procesos):
    duracion = random.randint(80, 120)
    tickets = random.randint(1, 4)
    proceso = {"id": i+1, "duracion": duracion, "tickets": tickets}
    procesos.append(proceso)

# Imprime Tabla
print("Planteamiento de Loteria:")
print("ID\tDuración\tBoletos")
for proceso in procesos:
    print("{}\t{}\t{}".format(proceso['id'], proceso['duracion'],
    proceso['tickets']))

# Crea la cola de procesos
cola = queue.Queue()

# Agrega procesos a la cola de procesos
for proceso in procesos:
    cola.put(proceso)

# Simular ejecución de procesos
while not cola.empty():
    # Obtener el proceso ganador de la lotería
    winner = None
    tickets_totales = 0
    for proceso in cola.queue:
        tickets_totales += proceso["tickets"]
    if tickets_totales > 0:
        num_loteria = random.randint(1, tickets_totales)
        for proceso in cola.queue:
            if num_loteria <= proceso["tickets"]:
                winner = proceso
                break
        num_loteria -= proceso["tickets"]

    if winner is None:
        continue

    # Ejecuta el proceso ganador por un quantum
```

```

winner["duracion"] -= 1
print(f"Tick {winner['id']}: Proceso {winner['id']} ejecutándose")

# Si el proceso aún no ha terminado, volver a agregarlo a la cola de procesos
if winner["duracion"] > 0:
    cola.put(winner)

print("Todos los procesos han terminado.")

```

Codigo Round Robin:

```

import random
import queue

# Creamos los procesos
num_procesos = random.randint(5, 8)
procesos = []

for i in range(num_procesos):
    duracion = random.randint(80, 120) # Asignar una duración aleatoria entre 80 y 120 ticks
    proceso = {"id": i+1, "duracion": duracion, "tick": 0}
    procesos.append(proceso)

# Imprime Tabla
print("Planteamiento por Round Robin:")
print("ID\tDuración")
for proceso in procesos:
    print("{}\t{}".format(proceso['id'], proceso['duracion']))

# Crea cola de procesos
cola = queue.Queue()

# Agrega procesos a la cola de procesos
for proceso in procesos:
    cola.put(proceso)

# Definir el rango del quantum de tiempo
quantum_min = 2
quantum_max = 12

# Simular ejecución de procesos
print("\nProcesos de planificación:")
while not cola.empty():
    # Obtener el proceso actual de la cola

```

```

proceso_actual = cola.get()

# Genera un quantum aleatorio
quantum = random.randint(quantum_min, quantum_max)

# Ejecuta el proceso actual por el quantum generado
for _ in range(quantum):
    proceso_actual["duracion"] -= 1
    proceso_actual["tick"] += 1
    print(f"Tick {proceso_actual['tick']}: Proceso {proceso_actual['id']} ejecutándose")

    # Verificar si el proceso ha terminado
    if proceso_actual["duracion"] == 0:
        print(f"\n\t\t Proceso {proceso_actual['id']} ha terminado \n")
        break

    # Si el proceso aún no ha terminado, volver a agregarlo a la cola de procesos
    if proceso_actual["duracion"] > 0:
        cola.put(proceso_actual)

print("\n Todos los procesos han terminado.\n")

```

¿Puede presentarse la situación de tener huecos sin ningún proceso listo para ejecutar? ¿Cómo lidiarían con ellos?

Para el código de la lotería, no se pueden presentar huecos debido a que en la cola generan a los procesos ganadores.

Sin embargo, para el código de Round Robin si hay una posibilidad de que los procesos pasen a los huecos, para lidiar con ello podremos realizar una condición donde en determinado tiempo verifica si hay un proceso en ejecución, en caso de que no hayan procesos este se puede esperar o puede tener un valor especial asignado el cual representara el intervalo de tiempo sin ejecución.

Al planificar estos procesos, se toman varias decisiones. ¿Les están asignando determinada prioridad? ¿Están subiendo o bajando de cola?

Se están subiendo

Preguntas para el Código de Round Robin

¿Cómo se compararía este método con los otros abordados?

Se le asignan tiempos determinados en Quantum (de 2 a 12 Ticks) a cada proceso que realiza, mientras que en el código de Planificación de lotería lo asigna de manera aleatoria (probabilística).

¿Para qué tipo de carga es más apto y menos apto?

Este algoritmo es más adecuado donde cada proceso posee una carga de trabajo más equitativa, sin embargo es menos efectivo en situaciones donde se tengan procesos de alta prioridad que requieran una ejecución más rápida.

¿Qué tan susceptible resulta a producir inanición?

Este algoritmo muy poco susceptible a producir inanición por el hecho de que cada proceso recibe su propio tiempo de ejecución, asignándoles una proporción correspondiente

¿Qué tan justa sería su ejecución?

ningún proceso está favorecido o perjudicado debido a su proporcionalidad

¿Qué modificaciones requeriría para planificar procesos con necesidades de tiempo real? (aunque sea tiempo real suave)

Podríamos asignar una prioridad para los procesos, de modo que los procesos con requisitos de tiempo real más estrictos tengan una mayor prioridad y estos reciban un tiempo de ejecución más rápido. Esto implicaría ajustar la lógica de asignación de tiempos de ejecución en función de las prioridades requeridas

¿Alguna otra reflexión o inquietud que les provoque?

El algoritmo Round Robin es popular por su sencillez y equidad al asignar tiempos de ejecución. No obstante, presenta limitaciones y puede no ser la elección óptima para todas las situaciones.