

Universidad de Guadalajara

Centro Universitario de Ciencias Exactas e Ingenierías

Ingeniería en computación

Seminario de solución de problemas de sistemas operativos

NRC 164138

Sección D07

Profesor Javier Rosales Martínez

Diego Armando Sánchez Rubio 217570609

Reporte Práctica 4.

Antecedentes.

En esta ocasión se seguirán revisando algunos algoritmos para la administración de procesos y cómo los va ejecutando, en este caso se revisarán dos en particular, Round Robin y Algoritmo en base a prioridades, para esto tenemos que establecer prioridades para los procesos dentro del txt que los contiene, ya que si no el algoritmo no funcionará.

Metodología.

En esta ocasión se volvió a usar el mismo método para leer el archivo que en la práctica pasada, primero se inicializa un lista vacía donde se guardarán los procesos, se abre el archivo y se lee línea por línea, de lo que se lea va a dividir el archivo por nombre, llegada, duración y prioridad, esto cada vez que se encuentre una coma, ya que así está marcado en el archivo, después se hace un casteo a entero y se va a devolver la lista con los datos ya cargados.

```
1 def cargarProcesos(archivo):
2     procesos = []
3     with open(archivo, 'r') as f:
4         for linea in f:
5             nombre, llegada, duracion, prioridad = linea.strip().split(',')
6             procesos.append({
7                 'nombre': nombre,
8                 'llegada': int(llegada),
9                 'duracion': int(duracion),
10                'prioridad': int(prioridad)
11            })
12     return procesos
```

Ahora para el algoritmo basado en prioridades, lo primero que se hace es ordenar los procesos por el tiempo de llegada, ya con esos se inicializan variables para controlar el tiempo actual, los procesos que faltan y lista de los procesos que ya pasaron, con un while se va a iterar en los procesos restantes y no haya más procesos en la lista, después de esto va a procesar la lista y los procesos que ya estén listo los va a añadir a la lista de procesos listos, así como se borrará de la lista de procesos restantes, con un if se va a hacer una validación para ver si hay algo en la lista de procesos listos, en caso de que no seguirá avanzando el tiempo hasta el siguiente proceso, ahora con esto se ordenarán mediante a la prioridad que tenga cada proceso, dependiendo del procesos con mayor prioridad, será el primero que

entre a ejecutarse, una vez que se ejecute se le sumará al tiempo actual la duración del ultimo proceso.

```
tabnine | Edit | Test | Explain | Document
def prioridades(procesos):
    procesos_ordenados = sorted(procesos, key=lambda x: x['llegada'])
    tiempo_actual = 0
    procesos_restantes = procesos_ordenados.copy()
    lista_listos = []

    print("\nAlgoritmo de Prioridades:")

    while procesos_restantes or lista_listos:
        while procesos_restantes and procesos_restantes[0]['llegada'] <= tiempo_actual:
            lista_listos.append(procesos_restantes.pop(0))

        if not lista_listos:
            tiempo_actual = procesos_restantes[0]['llegada']
            continue

        lista_listos.sort(key=lambda x: x['prioridad'])
        proceso = lista_listos.pop(0)
        print(f"Ejecutando {proceso['nombre']} (Duración: {proceso['duracion']}, Prioridad: {proceso['prioridad']})")
        tiempo_actual += proceso['duracion']
        print(f"Tiempo total transcurrido: {tiempo_actual}")
```

Para el round robin también se ordenan los procesos por tiempo de llegada, también están las variables de tiempo actual y la lista de procesos restantes y tenemos una cola de procesos listos para ejecutarse, lo primero que se hará es iterar mientras haya procesos pendientes o en cola, esto seguirá hasta que no haya más procesos en ninguna de las dos, con otro while se realizará que si un proceso ha llegado se va a mover a la cola de listos y se va a eliminar de procesos restantes, en caso de que no haya procesos listos el tiempo va a avanzar hasta que llegue el siguiente proceso, después de esto se sacará el primer proceso para ejecutarlo, ahora este proceso se ejecuta con el quantum definido, si el proceso no termina dentro del quantum, se ejecuta solo por quantum unidades, se reduce su tiempo restante y se reenvía al final de la cola, si el proceso si termina dentro del quantum lo completa.

```
def round_robin(procesos, quantum=3):
    procesos_ordenados = sorted(procesos, key=lambda x: x['llegada'])
    tiempo_actual = 0
    cola = []
    procesos_restantes = procesos_ordenados.copy()

    print("\nAlgoritmo Round Robin:")

    while procesos_restantes or cola:
        while procesos_restantes and procesos_restantes[0]['llegada'] <= tiempo_actual:
            cola.append(procesos_restantes.pop(0))

        if not cola:
            tiempo_actual = procesos_restantes[0]['llegada']
            continue

        proceso = cola.pop(0)
        if proceso['duracion'] > quantum:
            print(f"Ejecutando {proceso['nombre']} (Ejecutando {quantum} unidades de tiempo)")
            tiempo_actual += quantum
            proceso['duracion'] -= quantum
            cola.append(proceso) # Se vuelve a poner al final de la cola
        else:
            print(f"Ejecutando {proceso['nombre']} (Duración: {proceso['duracion']})")
            tiempo_actual += proceso['duracion']

    print(f"Tiempo total transcurrido: {tiempo_actual}")
```

Por ultimo se carga el archivo y se manda a llamar a las funciones.

```
66 # Cargar los procesos desde un archivo
67 archivo = "procesos.txt" # Nombre del archivo de entrada
68 procesos = cargarProcesos(archivo)
69
70 prioridades(procesos)
71 round_robin(procesos)
```

Conclusiones.

Comparado con los otros algoritmos, creo que estos dos podrían ser lo más eficientes para la mayoría de tareas que se hacen en un sistema normalmente, ya que basarse en las prioridades es algo muy importante en un sistema así como también darle un cierto tiempo al proceso con el quantum es útil para no atrasar otros procesos.

Referencias.

TecnoDigital. (2024a, May 20). *Algoritmo de planificación por prioridad en procesos: La guía definitiva*. Informática Y Tecnología Digital.

<https://informatecdigital.com/algoritmo-de-planificacion-por-prioridad/>

TecnoDigital. (2024b, November 3). *Planificación de Round Robin: Definición y Ejemplos*.

Informática Y Tecnología Digital. <https://informatecdigital.com/planificacion-de-round-robin-definicion-y-ejemplos/>