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 5.

Antecedentes.

Esta es la ultima parte de las prácticas de algoritmos de planificación, en esta ocasión, no se agregará ningún algoritmo, simplemente se hará la modificación al código para que ahora el usuario pueda escoger que algoritmo utilizar y además de darle la opción de agregar un nuevo proceso ya sea al principio o al final.

Metodología.

En esta ocasión no se modifico ninguna de las funciones que anteriormente ya existían, solo se agregaron una nuevas, para guardar los proceso o agregarlos y otra para el menú que le da a escoger al usuario que algoritmo quiere correr.

La primera es para guardar los procesos nuevos en el archivo que se está leyendo, en este caso recibe la lista de los procesos cargados y el archivo donde están los procesos, y lo que va a hacer es actualizar toda la lista de los procesos, lo hace reescribiendo el archivo abriéndolo en formato de escritura y escribiendo línea por línea lo que tenía la lista de procesos, esto mediante un for.

```
Tabnine | Edit | Test | Explain | Document
def guardarProcesos(archivo, procesos):
    with open(archivo, 'w') as f:
        for p in procesos:
            f.write(f"{p['nombre']}, {p['llegada']}, {p['duracion']}, {p['prioridad']}\n")
```

La siguiente función es para la función en donde se agrega el proceso a la lista, se le pide al usuario los siguientes datos, que corresponden a nombre, llegada, duración, prioridad, todo esto se guarda en una variable o diccionario que es nuevo_proceso, después se le pregunta al usuario si lo quiere guardar al principio o al final y dependiendo de su respuesta es dónde lo guardará, si lo inserta en la posición 0 es al inicio, pero si lo hace con append al final.

```
Tabnine | Edit | Test | Explain | Document
111 def agregarProceso(procesos):
112     nombre = input("Nombre del proceso: ")
113     llegada = int(input("Tiempo de llegada: "))
114     duracion = int(input("Tiempo de duración: "))
115     prioridad = int(input("Prioridad (menor es mayor prioridad): "))
116     nuevo_proceso = {'nombre': nombre, 'llegada': llegada, 'duracion': duracion, 'prioridad': prioridad}
117 
118     posicion = input("¿Agregar al principio o al final? (inicio/fin): ").strip().lower()
119     if posicion == 'inicio':
120         procesos.insert(0, nuevo_proceso)
121     else:
122         procesos.append(nuevo_proceso)
123     print("Proceso agregado con éxito.")
124
```

Después se tiene un pequeño menú de opciones para los algoritmos, donde se recibe la lista de procesos, el algoritmo y el quantum, dependiendo la opción es la que se ejecutará.

```python
def ejecutar_algoritmo(procesos, algoritmo, quantum=3):
    if algoritmo == '1':
        fifo(procesos)
    elif algoritmo == '2':
        sjf(procesos)
    elif algoritmo == '3':
        prioridades(procesos)
    elif algoritmo == '4':
        round_robin(procesos, quantum)
```

Por ultimo tenemos el menú completo, que carga el archivo de los procesos, y con un while muestra las, primero le muestra al usuario la opciones para Agregar procesos, Ejecutar algoritmos o salir, dependiendo lo que escoja es lo que hará, si agregar el proceso lo mandará a las funciones anteriormente definidas, en caso de que ejecute el algoritmo, aparecerá otro menú con la lista de algoritmos, dependiendo el numero que escoja es el que se ejecutará, en caso de que se escoja el round robin se le preguntará de que tamaño quiere el quantum.

```python
def menu():
    archivo = "procesos.txt"
    procesos = cargarProcesos(archivo)

    while True:
        print("\n1. Agregar proceso")
        print("2. Ejecutar algoritmo")
        print("3. Salir")
        opcion = input("Seleccione una opción: ")

        if opcion == '1':
            agregarProceso(procesos)
            guardarProcesos(archivo, procesos)
        elif opcion == '2':
            print("\nSeleccione el algoritmo:")
            print("1. FIFO")
            print("2. SJF")
            print("3. Prioridades")
            print("4. Round Robin")
            algoritmo = input("Ingrese el número del algoritmo: ")

            if algoritmo == '4':
                quantum = int(input("Ingrese el quantum para Round Robin: "))
                ejecutar_algoritmo(procesos, algoritmo, quantum)
            else:
                ejecutar_algoritmo(procesos, algoritmo)
        elif opcion == '3':
            print("Saliendo del programa.")
            break
        else:
            print("Opción inválida. Intente de nuevo.")
```

Conclusiones.

En realidad no fue muy difícil esta práctica, ya que no se agrego ningún algoritmo nuevo, solo fue programar la forma para que se agregará un nuevo proceso y que

además diera la opción al usuario para escoger en donde quería agregar el proceso y con que algoritmo ejecutarlo.

Referencias.

GeeksforGeeks. (2024, December 19). *Writing to file in Python*. GeeksforGeeks. https://www.geeksforgeeks.org/writing-to-file-in-python/

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/

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/