UNIVERSIDAD DE GUADALAJARA

Centro Universitario de Ciencias Exactas e Ingenierías

Ingeniería en Computación

6to semestre

Tema: Simulación Procesamiento por Lotes

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:
Miércoles, 08 de febrero de 2023

Act de aprendizaje 2 - Simulación de procesamiento por lotes

Objetivo

Desarrollar un programa que simule el procesamiento por lotes y que satisfaga los requerimientos planteados en Moodle.

Desarrollo

Los requerimientos planteados para la actividad son:

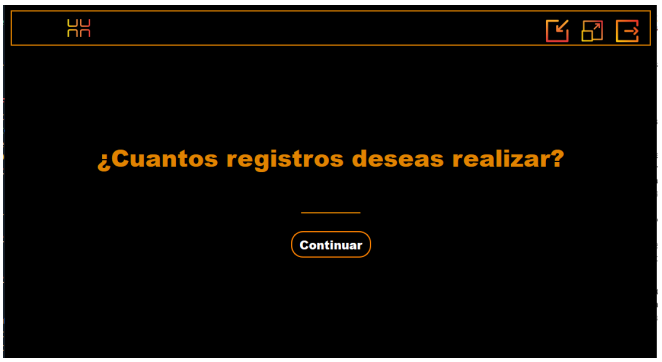
Requerimiento 1

Introducir desde teclado N procesos, estos serán los que conformen los lotes, la capacidad máxima de un lote es de 4, si el número de procesos que se introduce excede esta cantidad, se conformará otro lote. Al terminar un lote automáticamente se ejecuta el siguiente lote en espera. Información a Capturar por proceso:

1. Nombre de Programador
2. Operación a realizar (+, -, *, /, residuo) con sus respectivos datos (validar operaciones)
3. Tiempo Máximo Estimado (validar, debe ser mayor a 0)
4. Número de Programa (validar que sea Único, ID)

Solución

Realizamos una interfaz gráfica en Python y PyQt con una pestaña para captar la cantidad de procesos, la cual se somete a ciertas evaluaciones para determinar si el valor ingresado es aceptado (que el valor sea mayor a 0, que no sea una letra ni un número flotante)



Posteriormente, se abre una nueva pestaña para el registro de los procesos indicados. Dentro de esta pestaña encontramos un espacio para la entrada de texto, cada una correspondiente a uno de los campos solicitados por cada proceso:

- ID: Corresponde a un valor numérico entero. Se busca el identificador registrado en los registros ya captados para verificar que se trata de un id único.
- Nombre: Corresponde a un valor de tipo cadena. Se verifica a través de una expresión regular que la cadena ingresada se componga únicamente de letras del abecedario, de vocales con acento o espacios en blanco (espacio entre nombres-apellidos).
- Operación: Corresponde a un valor de tipo cadena. Se verifica que se trate de una operación válida siguiendo dos pasos:
    - Usando una expresión regular, se revisa que la operación cuente una sintaxis correcta (sus operandos sean enteros y su operador sea +-/%*).

- En caso de que se respete el formato de una operación, se trata de resolver el cálculo con la función *eval*, cuyo funcionamiento es el siguiente...

  eval(expresión) -> retorna el valor resultante de convertir y evaluar la operación expresada como cadena

- Si se completan ambos pasos con éxito, se marca como válida la operación propuesta.
- Por el contrario, si la cadena no es contemplada en la expresión regular o si la función *eval* sufre algún error en su ejecución, se expresa que la operación es inválida.

- Tiempo Máximo Estimado (TME): Corresponde a un valor numérico entero mayor a 0.

Valores válidos

Valores inválidos


En esta parte del programa, solo se almacenan los procesos en una lista de datos. No se efectúa ninguna clasificación por lotes en este punto, debido a que los lotes se forman según la manera que se tiene para extraer los datos de la lista.

Requerimiento 2

Atender los procesos en el orden que llegaron.

Solución

Se empleó una TDA de tipo lista para los lotes, la cual a su vez almacena listas más pequeñas con los componentes pertenecientes a un proceso:

Lotes <- [Proceso1 [ID, Nombre, Operación, TME], Proceso2 [ID2, Nombre2, Operación2, TME2], ...]

Aprovechamos la capacidad de Python para construir listas compuestas de diferentes tipos de datos, supliendo así el uso de objetos.

Requerimiento 3

En pantalla deberá mostrarse lo siguiente:

- Número de lotes pendientes: Con el total de procesos que se capturaron se conformarán uno o más lotes, dependiendo del número de procesos. En un principio el primer Lote es al que denominaremos Lote en Ejecución, si hay más de un Lote se anotará el número de lotes pendientes por ejecutar en este espacio (solo número).
- Lote en Ejecución: Por cada proceso que conforma el lote en ejecución se listarán los siguientes datos antes capturados:
    - Nombre del programador.
    - Tiempo Máximo Estimado.
- Proceso en Ejecución:
    - Se deberán mostrar todos los datos del proceso (Nombre, Operación, Tiempo Máximo Estimado, Número de Programa)
    - Tiempo transcurrido en ejecución.
    - Tiempo restante por ejecutar.
- Procesos Terminados:
    - Número de Programa.
    - Operación y datos.
    - Resultado de la operación.
- Contador Global: Desde que inicia la simulación, se desplegará un reloj global, es decir, un contador que lleve el tiempo desde el inicio del programa hasta que termine.

Solución

Diseñamos otra pestaña donde se van mostrando los datos sobre el lote actual, el proceso en ejecución y los procesos terminados.

- Número de lotes pendientes: Establecemos el valor inicial de lotes pendientes con el cociente resultante de dividir el número de procesos entre 4, corregimos sumando 1 si el número de procesos no es múltiplo de 4. Se decrementa en uno ese contador cada que se procesan 4 lotes, en caso de un lote incompleto, se decrementa cuando ya no quedan procesos en la lista.
    - Ejemplo
    - procesos = 7
    - Lotes =int(procesos/4) -> 1.75 -> 1
    - ¿Procesos%4!=0?, si, entonces corregimos el número de lotes sumando 1
    - Lotes = 2 (uno completo y uno de tres procesos)
- Lote actual: Se sacan de 4 en cuatro los procesos de la lista de procesos, y se muestran solo el nombre del programador a cargo y el tiempo máximo estimado para ese proceso. Se toman los procesos de arriba hacia abajo, despejando la línea en la que estaba el proceso. Posteriormente se muestra este proceso en el campo de ejecución. No es necesario usar alguna estructura para almacenar los procesos del lote actual
- Proceso en ejecución: Se extrae un proceso de la lista de procesos y se traslada a la sección de proceso en ejecución. Todos los registros permanecen en pantalla el mismo tiempo, por lo que los contadores adaptan su velocidad de actualización para respetar ese tiempo (un conteo de proceso se tarda lo mismo en llegar de 1 a 10 que otro en llegar de 1 a 1000).
- Procesos terminados: Finalmente cuando se termina el tiempo de un proceso en la sección de proceso en ejecución, este es trasladado a la tabla de terminados. En esta tabla se muestra solo el identificador, la operación y su resultado. para marcar cada lote se imprime una fila separadora que indica el número de lote
- Contador global: Es una variable en la cual se va sumando uno, considerando el tiempo máximo estimado de cada proceso. El valor final de este contador es la suma de todos los TME. La actualización de esta variable está adaptada para satisfacer un tiempo fijado.

Requerimiento 5

Al terminar un Lote automáticamente se continuará con el siguiente lote pendiente, actualizándose los datos en pantalla, el número de lotes pendientes disminuye, en el Lote en Ejecución ahora se listarán los procesos del lote que se trabaja, y en procesos terminados se marcará el fin de un lote y el comienzo del nuevo lote.

- Continuar con el siguiente lote: cuando es momento de trabajar con otro lote, los espacios para procesos en la sección de lote actual ya están limpios, así que solo se traen de la lista los cuatro procesos siguientes (o los que queden). Se resta uno al contador de lotes pendientes.
- El contador global permanece con la cuenta previa, esperando a registrar los tiempos pendientes.

- Procesos terminados seguirá mostrando las tuplas correspondientes a los procesos que ya fueron atendidos.
  Cuando acabe el procesamiento de todos los lotes, la tabla de terminados estará llena con los resultados de todos los procesos. El programa se quedará en pausa esperando a que se cierre la ventana o que se reinicie el procesamiento por lotes.

¿Por qué elegimos Python?

Algunas de las razones por las que elegimos Python son:

- La implementación de listas compuestas de datos abstractos es muy sencilla e intuitiva.
- Su forma de trabajar con interfaz gráfica a través de PyQt.
- Cuenta con una gran cantidad de funciones inteligentes para procesar datos, tales como las expresiones regulares que usamos, la búsqueda de un elemento en una lista, la función que usamos para evaluar las operaciones, etc.
- El IDE Spyder es bonito y cómodo.

Conclusión

La dificultad de esta actividad radicó en la gran cantidad de elementos que teníamos que controlar y actualizar de forma sincronizada. La solución a esto fue el uso de la interfaz gráfica, ya que facilitó mucho el acomodo de los datos de forma tal que fuera posible visualizar la forma en la que se procesaban las tareas.

En consola hubiera sido más difícil implementar todos los apoyos visuales que usamos, tales como tablas, colores, títulos, etc. Aunque hay que destacar que el código estaría bastante más limpio sin todas las sentencias usadas para crear los elementos de la interfaz.