

Universidad de Guadalajara

Centro Universitario de Ciencias Exactas e Ingenierías

Departamento de ciencias computacionales.

Ingeniería en computación



Seminario de Solución de Problemas de Sistemas
Operativos - D01.

Violeta del Rocío Becerra Velázquez

**“Programa 1. Simular el Procesamiento por
Lotes”**

2023A

08/02/2023

Flores Estrada Abraham Miguel Angel

Olguín Hernández Jair Benjamín

Índice.

Índice.	2
Objetivo.	3
Lenguaje utilizado.	3
Desarrollo del programa.	3
Conclusiones.	3

Objetivo.

El procesamiento por lotes es una forma de ejecutar tareas en un sistema informático que procesa grandes cantidades de datos en grupos o "lotes" en lugar de procesarlos individualmente. Esto permite una gestión más eficiente de los recursos del sistema y una mejor escalabilidad, ya que se pueden programar y automatizar tareas repetitivas, y procesar grandes cantidades de datos de forma más rápida y eficiente. Por lo que buscamos que el programa a desarrollar pueda simular como es el procesamiento por lotes, en este caso se agrupan en lotes distintas operaciones para simular los procesos. Donde cada lote agrupa 4 procesos, para después hacer la simulación de cómo sería el proceso de ejecución, los procesos pendientes y terminados. Para así tener un ejemplo más visual del procesamiento por lotes.

Lenguaje utilizado.

Para la elección del lenguaje de programación nos enfocamos más en las distintas herramientas o bibliotecas con las que contamos para el desarrollo de una interfaz. En nuestro caso escogimos **python**, esto porque contamos con experiencia en este lenguaje, y en la biblioteca pyqt5, que está enfocada para el desarrollo de interfaces gráficas, por lo que nos iba a facilitar un poco más todo el proceso de mostrar los distintos requisitos que debíamos mostrar en pantalla. Otra de las razones por la cual escogimos python fue por el conocimiento sobre las estructuras de datos que contamos en este lenguaje, debido a que tras investigar y conocer sobre el funcionamiento del procesamiento por lotes, sabíamos que el uso de listas nos podía ayudar al desarrollo del programa.

Desarrollo del programa.

Como se mencionó anteriormente, se escogió python para tener mayor facilidad en el desarrollo de la interfaz por lo que fue lo primero en realizarse, una vez con esto se tuvo que pensar un poco más en la lógica del programa y adaptarla a la interfaz creada.

MainWindow

Simulador de procesos

Num de procesos: 5 Aceptar Procesos por ingresar: 5 Procesos ingresados: 0

Nombre programador: Jair

Ingresa la operacion: 5 + 5

Tiempo estimado: 5 Id de proceso: 1

Agregar

Esta es nuestra primera interfaz, donde se piden los números de procesos, así como la operación a realizar, el tiempo estimado y el ID para el proceso. Cuenta con las validaciones de:

- Tener que agregar el número de procesos primero.
- Tener que ingresar todos los datos.
- Si el ID ya ha sido ingresado.
- Divisiones y residuos sobre 0.
- El tiempo no puede ser menos de 1.

Explicando un poco más del funcionamiento del programa, en esta primera parte tenemos:

```
1 self.procesos.append([self.numLote,Id,name,operacion,res,tiEs]);
2 self.leftProc -= 1;
3 self.procDone += 1;
4 self.con += 1;
```

Esta parte es el funcionamiento principal del programa, donde nosotros aprovechamos la facilidad de crear y manipular listas en python para poder guardar los procesos por lotes, ya que primeramente guardamos los datos a una lista de listas. Donde tenemos una lista principal llamada procesos y se van guardando en una lista todos los datos de cada proceso. Tenemos las variables “leftProc”, que

está simplemente nos ayuda para tener el conteo de procesos por ingresar. Después la variable “procDone”, para contar los procesos ingresados y por la variable “con”, esta es un contador que es la que nos ayuda a separar el número de lotes. Ya que está válida que cada 4 procesos se aumente en 1 el número de lotes, si representamos la lista de listas quedaría de la siguiente forma:

[1,1,Jair,5+5,10,5]
...
[2,5,Miguel,10/2,5,5]

Si vemos cuando agregamos los datos a la lista, tenemos que primero se agrega el número de lote, después el Id del proceso, nombre, operación, el resultado de la operación y el tiempo estimado. Donde cada 4 procesos el número de lote cambia y así sucesivamente hasta agregar todos los procesos. Una vez ingresemos nuestros procesos tendremos la siguiente interfaz:

The screenshot shows a software window titled "MainWindow" with a dark theme. The main title is "Simulador de procesos".

- Lotes pendientes:** A label followed by a blue box containing the number "2".
- Procesos Terminados:** A label above a table with 5 columns: Lote, Id, Name, Op, and Res. The table is currently empty.
- Lote en Ejecución:** A label above a table with 2 columns: Nombre and Tiempo. The table is empty.
- Proceso en Ejecución:** A label above a form with 5 input fields: Id, Nombre, Operacion, Tiempo Ejecucion, and Tiempo restante.
- Buttons:** A blue "Iniciar" button is centered below the input fields. At the bottom center, there is a "Contador Global:" label followed by a blue box containing the number "0".

En este caso aún no comenzamos la simulación de los procesos, pero se agregaron 6 procesos, por lo que se separaron en 2 lotes. Una vez que demos iniciar, se nos mostrará el lote en ejecución, es decir que mostrará todos los procesos pendientes y

los irá ejecutando de uno por uno para después agregarlos en los procesos terminados.

- Primer lote:

MainWindow

Simulador de procesos

Lotes pendientes: 1

Lote en Ejecución

	Nombre	Tiempo
1	Jair	5
2	Jair	5
3	Benjamin	17
4	Abraham	4

Proceso en Ejecución

	1
Id	1
Nombre	Jair
Operacion	4 + 5
Tiempo Ejecucion	2
Tiempo restante	3

Iniciar

Contador Global: 2

Procesos Terminados

Lote	Id	Name	Op	Res
------	----	------	----	-----

- Segundo lote:

MainWindow

Simulador de procesos

Lotes pendientes: 0

Lote en Ejecución

	Nombre	Tiempo
1	Miguel	15
2	6	4

Proceso en Ejecución

	1
Id	5
Nombre	Miguel
Operacion	56 / 8
Tiempo Ejecucion	4
Tiempo restante	11

Iniciar

Contador Global: 35

Procesos Terminados

	Lote	Id	Name	Op	Res
1	1	1	Jair	4 + 5	9
2	1	2	Jair	5 - 15	-10
3	1	3	Benja...	90 / 10	9.0
4	1	4	Abraha...	3 * 3	9

- Procesamiento terminado:



Ya en esta segunda parte del programa, el funcionamiento se basa en los siguiente:

```

1  done = []
2      for i in range(self.numLote):
3          aux = []
4          tablerow = 0
5          self.label_11.setText(str(self.numLote - (i+1)))
6          for row in self.procesos:
7              if row[0] == i+1:
8                  aux.append(row);

```

En este caso creamos una lista para los procesos terminados y una lista auxiliar, lo que hacemos es tener 2 for anidados que nos ayudaran para ir recorriendo nuestra lista de procesos, el primer for nos ayudará para recorrer los diferentes lotes, mientras que el segundo irá por cada uno de los procesos. Es decir el primer for es el que nos ayuda a identificar los procesos con el lote 1 hasta n, con eso se nos facilitó más el poder mostrar los procesos. Mientras que el segundo for nos ayuda para recorrer cada uno de los procesos del lote con el que estamos trabajando, los estará agregando en la lista aux, donde su función es solo guardar los procesos del

lote en ejecución, por eso mismo se inicializa por cada vuelta. Después de todo esto solo queda mostrarlos en la tabla de Lote en ejecución y Proceso en ejecución, y una vez que se muestran, se agregan a la lista done, que como se mencionó anteriormente es la lista que se encarga de guardar los procesos terminados.

Conclusiones.

Como pudimos ver el desarrollo del programa simula el como es el procesamiento por lotes, en este caso son operaciones básicas, pero se logra entender mucho más el cómo es que se agrupan estas operaciones en lotes y va realizando cada una de ellas de manera grupal. Y si lo vemos de una manera más práctica, podemos ver que el procesamiento por lotes es útil para tareas que no requieren una respuesta en tiempo real o que sea dependiente del usuario, lo que permite a los sistemas dedicarse a otras tareas mientras se ejecutan los procesos en segundo plano, en este caso podemos dejar que las operaciones se resuelvan y al final nos mostrará el resultado de todas y el tiempo que tardó. A Pesar de que fue una actividad sencilla si se necesitaba de pensar acerca de la lógica del como podíamos realizarla, ya que también se nos ocurrió hacerla con programación orientada a objetos donde en la lista guardará objetos, pero nos pareció mejor la idea por la que optamos, y gracias a la propia interfaz es más fácil apreciar del procesamiento y de entender el programa.