Universidad de Guadalajara

CUCEI (Ciencias Exactas e Ingenierías)

Departamento de ciencias computacionales

Materia: Sistemas operativos

Profesor: Violeta del Rocío Becerra Velázquez

Gómez Rubio Alexia

Rico Morones Denice Estefania

Carrera: Ingeniería en computación

Sección: D04

Programa 2

Tema: Procesamiento por lotes con multiprogramación

Fecha:15 septiembre 2024

Contenido

Objetivo

Crear un programa, en el que se registren “n” procesos, los cuales conformarán un lote cuando se tengan cinco procesos, en cada proceso se generarán los datos aleatorios para el nombre, ID, operación a realizar (suma, resta, multiplicación, división, módulo), el tiempo máximo estimado que el programa tardará en ejecutar esa operación.

Este programa se controlará con algunas teclas, con la letra “e” terminamos el programa forzándolo como un error, con la letra “i” interrumpimos la ejecución del programa y lo ponemos al final de la lista de los programas que están esperando ser ejecutados, con la letra “p” se detiene la ejecución del programa momentáneamente, hasta que se presione la letra “c”, la cual reanudará el programa que se pauso.

Los procesos se van a ejecutar en el orden en que se registraron a menos que sean interrumpidos por el usuario, y tendrán que irse actualizando cuando cambien de módulo (lotes pendientes, procesos en ejecución, procesos terminados). Con ayuda de un contador global, se sumarán los tiempos estimados de todos los procesos registrados hasta terminar todos.

Descripción del programa (qué lenguaje utilizamos, por qué lo utilizamos y no elegimos otro lenguaje para programar, estructura que manejamos en el programa)

<u>Lenguaje utilizado</u>

Decidimos seguir utilizando JavaScript y HTML para **facilitar** la demostración gráfica del programa, y porque solo le cambiamos/agregamos al programa anterior unas funciones que nos hacían falta.

Una de las ventajas del uso de JavaScript junto con HTML para mostrar los resultados gráficamente dentro de nuestro servidor local es que a comparación de un lenguaje que se maneja en consola, es mucho más fácil de manipular y utilizar para acomodar gráficamente los datos de una manera muy sencilla sin la necesidad de recurrir a librerías para el manejo de las coordenadas en consola; no se está utilizando ningún framework.

Estructura del programa:

Al igual que en anterior dividimos el código HTML y el código de JavaScript para tener una mejor organización en nuestro programa.

En la parte de HTML nos apoyamos de las etiquetas para representar gráficamente la captura de los procesos como la ejecución de cada uno de ellos, además, también nos ayuda a ejecutar y mandar llamar a la función que dentro del .js nos da la funcionalidad del programa, esto, gracias a los <form>.

Para la parte del JavaScript agregamos

Al inicio del código declaramos alguna variables globales, como: procesos el cual almacena los procesos generados, lotes guarda los procesos en grupos de hasta cinco procesos, loteEnEjecucion contiene el lote actual que está siendo procesado, el procesoActual guarda el proceso específico que se está ejecutando, contadorGlobal lleva la cuenta del tiempo total desde el inicio de la ejecución, intervaloGlobal es el temporizador que incrementa el contadorGlobal cada segundo, intervaloProceso controla la ejecución de cada proceso, idContador es un contador que asigna un identificador único a cada proceso, isPaused es una bandera que indica si la ejecución está pausada o no.

El código incluye un mecanismo para pausar y continuar la ejecución de los procesos. Cuando el usuario presiona la tecla "p", la función pausar() detiene ambos intervalos de tiempo y con una alerta se le informa al usuario de su estado. Si el usuario presiona "c", la función continuar() reanuda los intervalos y retoma la ejecución del proceso actual.. La función Procesos() crea una cantidad de procesos basada en la entrada del usuario, asignando nombres, operaciones y tiempos aleatorios. Estos procesos se agrupan en lotes mediante la función actualizarLotes(), que mueve los procesos a lotes y actualiza el contador de lotes pendientes.

Para iniciar la simulación, la función Iniciar() se encarga de arrancar el contador global y llamar a ejecutar(), que gestiona la ejecución de los lotes. La función ejecutar() toma el primer lote de la lista de lotes y lo coloca en ejecución. A través de la función mostrarLote(), se actualiza la interfaz visual para mostrar los procesos del lote en ejecución. Luego, la función ejecutarProceso() comienza a procesar los elementos del lote uno a uno, decrementando su tiempo restante en

intervalos de un segundo. Si el tiempo de un proceso llega a cero, se calcula el resultado de su operación y se mueve al siguiente proceso o lote.

La función de handleInterruption() se activa cuando el usuario presiona la tecla "i". Esto provoca que el proceso actual, que está en ejecución, sea interrumpido y clonado. El proceso clonado conserva el tiempo restante de ejecución del proceso original y se insertará al final del lote actual para ser ejecutado más adelante. El proceso sigue ejecutándose desde donde fue interrumpido.

De igual forma la función handleError() se activa cuando el usuario presiona la tecla "e". El proceso actual es finalizado repentinamente y se registra en la interfaz como "terminado con error". Luego, se pasa al siguiente proceso del lote o, si el lote ha terminado, se continúa con el siguiente lote. Esta función representa una falla crítica en la ejecución de un proceso, que no puede completarse de forma correcta.

El procesamiento de cada operación matemática es realizado por la función Operacion(), que ejecuta la operación correspondiente al tipo de proceso, como suma, resta, multiplicación, división o módulo. Una vez que un proceso finaliza su ejecución, el resultado es mostrado en la interfaz mediante la función procesarResultado(), la cual asegura que no se dupliquen los resultados y que los procesos ya terminados se registren correctamente. El ciclo continúa hasta que todos los lotes han sido procesados y la simulación concluye.

Problemas presentados y su solución

Entre los problemas que llegamos a presentar están las funciones de las teclas; había dos opciones, una implementarlas mediante eventos (keydown) o migrar el programa a react, nosotras nos decidimos por continuar con el mismo programa que teníamos y hacer las debidas pruebas para ver la dificultad del uso de los eventos y cómo se maneja en conjunto con HTML. Al momento de hacer las distintas funciones hubo problemas en cuanto la interrupción de los procesos ya que el programa muchas veces se nos llegó a comportar de una manera descontrolada por lo cual tuvimos que hacer arreglos a las funciones que ya se tenían como lo es ejecutarProceso(), además de los ajustes en los contadores e ir clonando y apilando los procesos interrumpidos para que no tuvieran error en la ejecución después de su interrupción, en cuanto a los tiempo existieron problemas con la sincronización de los contadores ya que con cada interrupción el programa

se ejecutaba más rápido lo cual ocasionaba problemas al final ya que el contador continuaba cuando ya no había ningún lote pendiente.

Conclusiones

Denice Estefania Rico Morones:

Como conclusión puedo decir que fue una práctica interesante de realizar ya que considero que obtuvimos más conocimientos sobre cómo trabaja la multiprogramación, además, el uso del lenguaje de programación JavaScript es sencillo de comprender en este caso el uso de eventos y cómo estos pueden acceder al DOM y hacer el trabajo mucho más sencillo gracias a la unión de HTML que utilizamos en este código; pese a las dificultades pudimos concluir con el programa de una manera eficaz, y un trabajo en conjunto que convirtió de esta actividad algo mucho más fácil de implementar.

Alexia Gómez Rubio:

Por mi parte tuve la oportunidad de investigar sobre la programación por lotes con multiprogramación, lo cual me ayudó a comprender mejor el tema. Además, las explicaciones proporcionadas por la maestra fueron clave para aclarar algunos conceptos. Durante el desarrollo del programa, enfrentamos algunos inconvenientes, especialmente relacionados con la funcionalidad de las teclas. Sin embargo, logramos resolverlos de manera efectiva, lo que nos permitió finalizar el proyecto de forma satisfactoria, gracias al trabajo colaborativo.