



15 DE AGOSTO DE 2017

EMULADOR DE PLANIFICADOR DE DISCO SECUNDARIO A NIVEL DE SISTEMA OPERATIVO

INFORME DEL PROYECTO

ABARCA VIQUEZ ANDRES

Solución planteada

Para este proyecto se optó por hacer uso del lenguaje orientado a objetos C#, usando el framework de Microsoft .NET, para así lograr un solución con un tiempo de respuesta alto, fácil acceso por medio de Windows Forms.

La solución consta de dos clases: objeto disco de almacenamiento secundario con los atributos de la cola de peticiones de acceso a los cilindros del disco, el tamaño del disco y la posición del cabezal para emular la lectura. La segunda clase contiene todas las funciones necesarias para obtener los datos del objeto anteriormente definido y las funciones para ordenar los valores para cada algoritmo de planificación y finalmente, las funciones para graficar el desplazamiento por algoritmo.

La solución planteada hace uso de algoritmos creados empíricamente para la planificación variada, FIFO, SSF, SCAN, CSCAN, y se grafican mediante la clase "Charts" del framework.

Resultados obtenidos

La solución planteada cumple los objetivos del problema, por medio de la clase chart, se insertó en un Windows Form, tres objetos (text box) para la inserción y obtención de los datos del disco, en donde se obtiene el tamaño del disco, la cola de peticiones de acceso al disco y el punto de inicio. La solución además cuenta con un objeto tipo chart para la graficación de los movimientos de la planificación de los distintos algoritmos.

Para la graficación de los algoritmos de planificación, se planteó una organización de los valores de la cola de peticiones. Por ejemplo, para el algoritmo SSJ, se creó un nuevo arreglo de los valores de la cola de peticiones que son ordenados por medio de la sintaxis lambda de c#, los valores más cercanos al punto de partida. Para el algoritmo FIFO, no se cambió el orden de los valores de la cola de peticiones. Para el algoritmo SCAN, se implementó una función que ordena los valores menores al punto de inicio de manera descendente hasta llegar a cero, y los valores mayores al punto de inicio de manera ascendente hasta llegar al último punto de la cola de peticiones. Para el algoritmo CSCAN, se implementó una función que ordena los valores mayores al punto de inicio de manera ascendente hasta llegar al valor mayor del disco, y después brincar al punto cero, después se ordenaron los valores menores al punto de inicio de manera ascendente hasta llegar al último punto de la cola de peticiones restante.

Todos los algoritmos fueron graficados en el mismo chart del Windows form.

Benchmarking

FIFO

Es el algoritmo más simple de todos los algoritmos de planificación de disco, las solicitudes se direccionan en el orden en el que llegan a la cola de peticiones.

Ventajas:

- Cada solicitud tiene una oportunidad justa
- Sin aplazamiento indefinido

Desventajas:

- No intenta optimizar el tiempo de búsqueda
- Puede no proporcionar el mejor servicio posible

SSF

En SSF (Tiempo de búsqueda más corto primero), las solicitudes que tienen el tiempo de búsqueda más corto se ejecutan primero. Por lo tanto, el tiempo de búsqueda de cada solicitud se calcula por adelantado en cola y luego se programan de acuerdo con su tiempo de búsqueda calculado. Como resultado, la solicitud cerca del brazo del disco se ejecutará primero. SSF es ciertamente una mejora sobre FIFO, ya que disminuye el tiempo de respuesta promedio y aumenta el rendimiento del sistema.

Ventajas:

- El tiempo medio de respuesta disminuye
- Aumento del rendimiento

Desventajas:

- Gastos generales de tiempo extra para calcular el tiempo de búsqueda por adelantado
- Puede provocar problemas de respuesta para una solicitud si tiene mayor tiempo de búsqueda en comparación con las solicitudes entrantes
- Alta variación del tiempo de respuesta ya que SSF sólo favorece algunas peticiones

SCAN

En el algoritmo SCAN el brazo del disco se mueve en una dirección descendente al punto de inicio y después de llegar al final del disco, invierte su dirección y de nuevo prosigue con las solicitudes restantes ascendentemente. Como resultado, las peticiones en el rango medio se reparten más y las que llegan detrás del brazo del disco tendrán que esperar.

Ventajas:

- Alto rendimiento.
- Baja variación del tiempo de respuesta.
- Tiempo medio de respuesta.

Desventajas:

- Tiempo de espera largo para las solicitudes de ubicaciones próximas a ubicaciones que acaba de visitar el brazo del disco.

CSCAN

En el algoritmo CSCAN el brazo del disco en lugar de invertir su dirección va al otro extremo del disco y comienza a atender las solicitudes desde allí. Por lo tanto, el brazo del disco se mueve de forma circular.

Ventajas

- Proporciona un tiempo de espera más uniforme en comparación al SCAN.

LOOK

Es similar al algoritmo de programación de disco SCAN, excepto que el brazo de disco a pesar de ir al final del disco, va sólo a la última solicitud para ser colocado en frente de la cabeza y luego invierte su dirección desde allí. De este modo se evita el retraso adicional que se produjo debido a un desplazamiento innecesario hasta el final del disco.

CLOOK

Como LOOK es similar al algoritmo CSCAN. En CLOOK, el brazo del disco a pesar de ir hasta el final sólo va a la última solicitud para ser colocado en frente de la cabeza y luego de allí va a la última petición del otro extremo. Por lo tanto, también evita el retraso adicional que se produjo debido a un desplazamiento innecesario al final del disco.

Cada algoritmo es único y el rendimiento general es variado ya que depende del número y tipo de solicitudes.