

“Caso de Estudio 2 – Memoria Virtual”

Luis D. Castro Gonzalez, Andrés M. Ochoa Toro

201720389 - 201913554

Caso 2 – Infraestructura Computacional

Universidad de los Andes, Bogotá, Colombia

Fecha de presentación: abril 12 de 2020

1 Descripción de la solución

En el desarrollo de la solución de este caso se implementó un esquema similar a lo que sería la memoria de un computador. Se tienen una clase tablas de páginas compuestas por páginas y una clase memoria real compuesta por marcos de página. Además, el proyecto tiene dos clases que representan cada uno de los threads de actualizar y envejecer.

El thread actualizar se encarga de leer las referencias ingresadas y buscar si las tablas de páginas tienen una asociación con un marco de página. Y, de ser necesario se resuelve el fallo de página de acuerdo con el envejecimiento de los marcos de página. Por otro lado, el thread envejecer, se encarga de ir actualizando el contador de cada uno de los marcos de página que se encuentran en la memoria real.

2 Estructuras de datos

En este caso, se usaron las siguientes estructuras de datos para representar cada uno de los elementos que se pensaron necesarios para implementar la solución.

Tabla de Página (TablaPagina): Clase que representa el arreglo de páginas por medio de un arreglo de tamaño fijo del tamaño definido a partir del archivo de configuración. Esta clase tiene la responsabilidad de inicializar todas las páginas, dar si una página está relacionada con una página y relacionar una página con un marco de página.

Página (Pagina): Clase que representa una página y la información de qué marco de página está relacionado con ella. Las responsabilidades de esta clase es dar qué marco de página está asociado con ella y actualizarlo cuando sea necesario.

Memoria Real(Memoria Real): Clase que representa la memoria RAM y la sección que se le va a asignar a los marcos de página. Los marcos de página se representan a través de un arreglo de tamaño fijo. Esta clase es responsable de solucionar los fallos de página originados tanto porque se está iniciando el proceso o porque ya está llena y es necesario cambiar un marco de página.

Marco de Página(MarcoPagina): Clase que representa los marcos de página. Las responsabilidades de esta clase es actualizar su contador y de ser necesario agregar un bit en la sección más significativa para aplicar el proceso de envejecimiento.

3 Esquema de sincronización

La implementación de threads se realizó a través de dos clases, las cuales fueron el envejecimiento y actualización. Cada una de estas clases se encargan de realizar la función definida por su nombre. Por otro lado, la sincronización se realiza en un método de cada clase.

En el caso de la actualización, se sincroniza sobre el método de leer las referencias, especialmente sobre el objeto que representa la memoria real. Este método se implementa a partir de un llamado en el run con el objetivo de facilitar la sincronización y el entendimiento de la funcionalidad. Y este llamado se realiza por cada una de las referencias ingresadas en el archivo de configuración.

```
private void leerReferencia(int posTablaPagina, int i) {
    synchronized (memoriaReal) {
        int posMarcoPagina = tablaPagina.obtenerRefMarcoPagina(posTablaPagina);
        if( posMarcoPagina != -1) {
            bitsPaginas[posMarcoPagina] = 1;
        } else {
            fallosPagina++;
            posMarcoPagina = memoriaReal.falloPagina();
            tablaPagina.relacionar(posTablaPagina, posMarcoPagina);
            bitsPaginas[posMarcoPagina] = 1;
        }
    }
}

public void run() {
    int posTablaPagina;
    for (int i = 0; i < referencias.size(); i++) {
        try {
            posTablaPagina = referencias.get(i);
            leerReferencia(posTablaPagina, i);
            Thread.sleep(5);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
    memoriaReal.cambiarTerminar();
    System.out.println("Número de Fallo de Páginas:" + fallosPagina);
}
```

En el caso del envejecimiento se sincroniza sobre el método de actualizar, y al igual que el caso anterior, se sincroniza sobre la memoria real, que en este caso se llama marcos de página en memoria. Este método se realiza mientras el proceso se encuentra procesando referencias, y se termina cuando la actualización termine de procesar la última referencia.

```
public void actualizar() {  
    synchronized (marcosPaginaMemoria) {  
        MarcoPagina[] marcosPagina = marcosPaginaMemoria.darMarcosPagina();  
        for (int i = 0; i < bitsPaginas.length; i++) {  
            MarcoPagina mPagina = marcosPagina[i];  
            if(marcosPagina[i] != null)  
            {  
                if(bitsPaginas[i] == 1)  
                {  
                    mPagina.sumar();  
                    bitsPaginas[i] = 0;  
                } else {  
                    mPagina.actualizar();  
                }  
            }  
        }  
    }  
}  
  
public void run() {  
    while(marcosPaginaMemoria.darTerminar() != 5) {  
        try {  
            actualizar();  
            Thread.sleep(1);  
        } catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

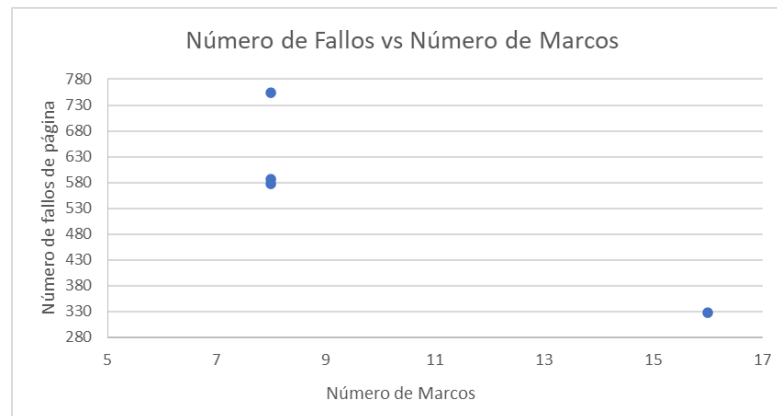
La sincronización sobre estos dos métodos se aplica a través de la referencia a la memoria real, la cual es la encargada de contener los marcos de página. Este objeto se comparte entre los dos threads, lo que permite la sincronización.

4 Recopilación de datos

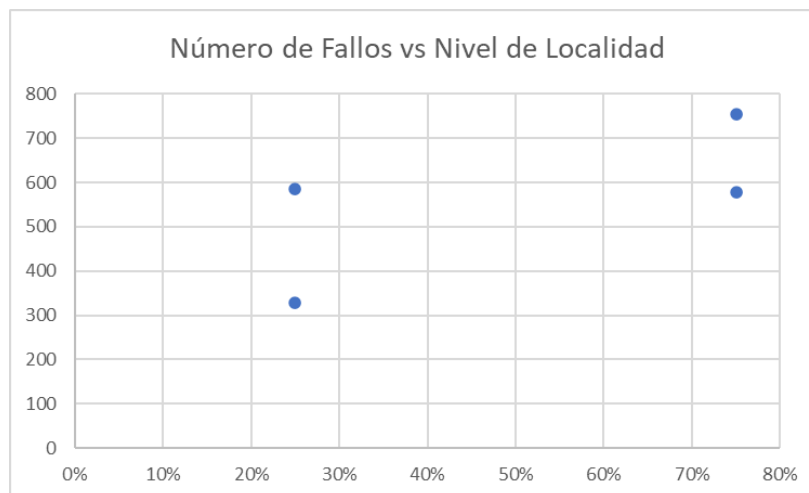
Después de la ejecución de los diferentes casos de prueba, se obtuvieron los siguientes datos representados a continuación:

Caso	Número de marcos de página	Número de páginas del proceso	Nivel de Localidad	Número de Referencias	Número Fallo de página
1	8	40	25%	1000	591
2	8	60	75%	1000	580
3	8	100	75%	1000	756
4	16	40	25%	1000	323

A partir de los casos suministrados y los resultados de correr las pruebas que se demuestran en la tabla anterior, se construyen las siguientes gráficas.



Número de fallos vs Número de Marcos



Número de Fallos vs Nivel de Localidad

Por último, a partir de los datos obtenidos y las gráficas se puede concluir que los datos si concuerdan con lo esperado al tener en cuenta cada uno de los casos. Entre mayor sea la cantidad de marcos de página o más cercano sea el valor de la cantidad de tablas de página a los marcos de página menor va a ser la cantidad de fallos de página que van a ocurrir. Así mismo, de acuerdo con la segunda gráfica se puede entender que entre menor sea el porcentaje de localidad el porcentaje de fallos puede ser menor.

Estos resultados sí concuerdan con lo esperado, ya que entre mayor sea la cantidad de marcos de página, mayor será la cantidad de información que se tendrá y menos fallos se van a tener.