



TECNOLÓGICO NACIONAL DE MÉXICO

CAMPUS JEREZ

MATERIA: ESTRUCTURA DE DATOS

DOCENTE: ISC SALVADOR ACEVEDO SANDOVAL

3° SEMESTRE

TEMA 1: INTRODUCCIÓN A LAS ESTRUCTURAS DE
DATOS

ACTIVIDAD 5: MAPA CONCEPTUAL - MANEJO DE
MEMORIA

ALUMNA: LIZA AREMY SANTANA CONTRERAS

NO. CONTROL: 16070005

18 DE DICIEMBRE DE 2020

JEREZ DE GARCIA SALINAS

RESUMEN

FACTORES DE DESEMPEÑO

Una influencia importante en el rendimiento de un programa java se puede dividir en dos partes principales:

- Consumo de memoria.
- Tiempo de ejecución total de un programa.

En caso de que el programa es en algún caso un programa que interactúa con otros también el tiempo de respuesta es un hecho muy importante de la ejecución de un programa.

Un CPU promedio puede hacer aproximadamente mil millones (10^9) operaciones por segundo.

Este artículo no cubre la concurrencia. Si desea leer acerca de la simultaneidad / multihilo, vea Concurrency / Multithreading en java.

MANEJO DE MEMORIA EN JAVA

Java maneja su memoria en dos áreas. El montón y la pila. Comenzaremos con un breve resumen de la memoria en general en una computadora. A continuación, se explica el montón java y la pila.

MEMORIA NATIVA

La memoria nativa es la memoria que está disponible para un proceso en java. La memoria nativa es controlada por el sistema operativo (OS) y basada en la memoria física y otros dispositivos físicos, p. discos, memoria flash, etc.

El procesador (CPU) de la computadora calcula las instrucciones para ejecutar y almacena sus resultados de cálculo en registros. Estos registros son elementos de memoria rápida que almacena el resultado de la CPU.

El procesador puede acceder a la memoria normal a través del bus de memoria. La cantidad de memoria que puede acceder una CPU se basa en el tamaño de la dirección física que utiliza la CPU para identificar la memoria física. Una dirección de 16 bits puede tener acceso a 2×16 (= 65.536) ubicaciones de memoria. Una dirección de 32 bits puede tener acceso a 2×32 (= 4.294.967.296) ubicaciones de memoria. Si cada área de memoria consta de 8 bytes, entonces un sistema de 16 bits puede acceder a 64 KB de memoria y el sistema de 32 bits puede acceder a 4 GB de memoria.

Normalmente, un sistema operativo (SO) utiliza la memoria virtual para asignar la memoria física a la memoria que cada proceso puede ver. El sistema operativo asigna entonces memoria a cada proceso en un espacio de memoria virtual para este proceso y asigna el acceso a esta memoria virtual a la memoria física real.

El sistema actual de 32 bits utiliza una extensión (Extensión de dirección física (PAE)) que extiende el espacio físico a 36 bits del sistema operativo. Esto permite que el OS tenga acceso a 64GB. El sistema operativo utiliza entonces la memoria virtual para permitir que el proceso individual de 4 GB de memoria. Incluso con PAE activado un proceso no puede acceder a tan más de 4 GB de memoria. Por supuesto, con un sistema operativo de 54 bits esta limitación de 4GB ya no existe.

MEMORIA EN JAVA

Java administra la memoria para su uso. Nuevos objetos creados y colocados en el montón. Una vez que su aplicación ya no tiene referencia a un objeto, el recolector de elementos no deseados de Java puede eliminar este objeto y eliminar la memoria para que la aplicación pueda volver a usarla.

MONTÓN DE JAVA

En el montón, Java Virtual Machine (JVM) almacena todos los objetos creados por la aplicación, p. utilizando el operador "nuevo". El recolector de basura de Java (gc) puede separar lógicamente el montón en diferentes áreas, de manera que el gc pueda identificar más rápidamente los objetos que se pueden quitar. La memoria para objetos nuevos se asigna en el montón en tiempo de ejecución. Las variables de instancia viven dentro del objeto en el que se declaran.

JAVA STACK

La pila es donde se almacenan las invocaciones de método y las variables locales. Si se llama a un método, se coloca el marco de pila en la parte superior de la pila de llamadas. El marco de pila contiene el estado del método que incluye la línea de código que se está ejecutando y los valores de todas las variables locales. El método en la parte superior de la pila es siempre el método de ejecución actual para esa pila. Los hilos tienen su propia pila de llamadas.

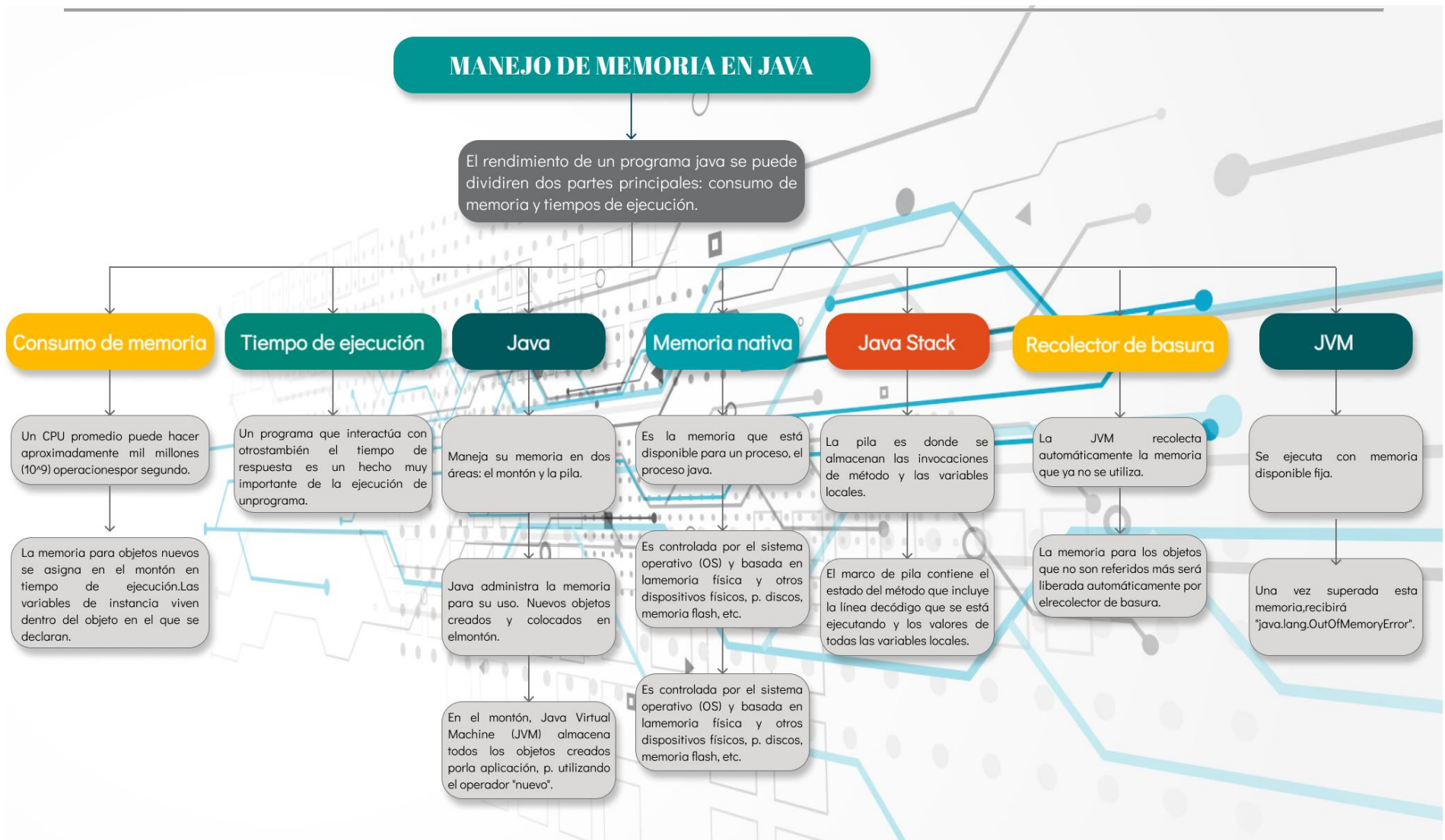
RECOLECTOR DE BASURA

La JVM recolecta automáticamente la memoria que ya no se utiliza. La memoria para los objetos que no son referidos más será liberada automáticamente por el recolector de basura. Para ver que el recolector de basura comienza a trabajar, agregue el argumento de línea de comandos "-verbose: gc" a su máquina virtual. Un artículo detallado sobre el recolector de basura se puede encontrar aquí: [Tuning Garbage Collection con la máquina virtual Java](#).

AJUSTES DE MEMORIA PARA LA MÁQUINA VIRTUAL JAVA

La JVM se ejecuta con memoria disponible fija. Una vez superada esta memoria, recibirá "java.lang.OutOfMemoryError". La JVM intenta hacer una elección inteligente sobre la memoria disponible en el inicio (consulte la configuración de Java para obtener detalles) pero puede sobrescribir la configuración predeterminada con la siguiente configuración. Para dar vuelta al funcionamiento usted puede utilizar ciertos parámetros en la JVM.

MAPA CONCEPTUAL



REFERENCIAS

- [1] Carballeira, F. G. Arquitectura de la Máquina Virtual Java.