ESAT INNOVATION SCHOOL | ESCUELA SUPERIOR DE ARTE Y TECNOLOGÍA

MEMORY STACK

Nombre Asignatura: Algoritmos y Estructuras de datos
Módulo Asociado: 2 VGP
Profesor: Iván Sancho
Curso: 2023/2024
Autor/es: Carlos Mazcuñan y Lucas Calatayud

Índice/Index

1.- Definición

Una “stack” o pila de memoria es una región de la memoria de un programa de computadora que se organiza como una estructura de datos de tipo pila. Es un área de memoria que se utiliza para almacenar datos temporalmente y gestionar las llamadas a funciones dentro de un programa.

La pila funciona según el principio de “último en entrar, primero en salir” (LIFO, por sus siglas en inglés). Esto significa que el último elemento que se añade a la pila es el primero en ser retirado. La pila es utilizada para almacenar direcciones de retorno, variables locales y otros datos temporales durante la ejecución de un programa.

Cuando se llama a una función, se reserva un bloque de memoria en la pila para almacenar las variables locales y otros datos asociados con esa función. Estos datos se eliminan de la pila cuando la función ha terminado de ejecutarse. Además, la dirección de retorno (la dirección de la instrucción a la que debe regresar el programa después de que se complete la función) también se almacena en la pila.

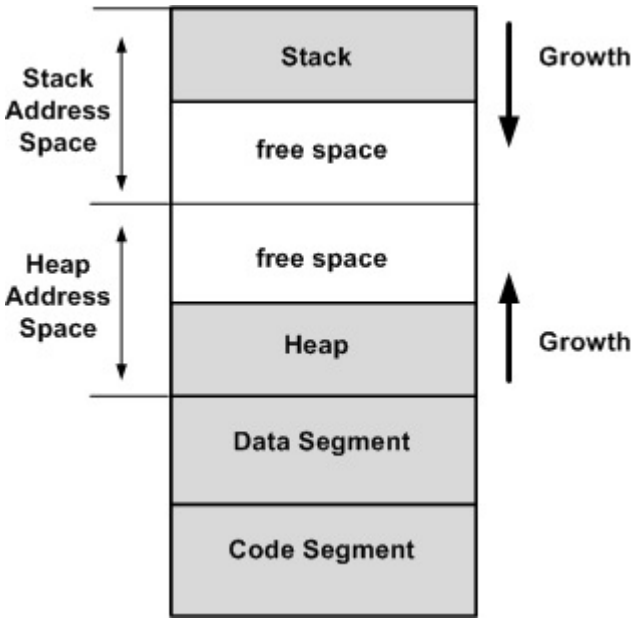

Fig. 01

2.- Operaciones

Una memory stack típicamente admite dos operaciones principales: “push” y “pop”. Estas operaciones se utilizan para agregar datos a la pila y eliminar datos de la pila, respectivamente.

Push

Se utiliza para agregar un nuevo elemento a la parte superior de la pila. Este nuevo elemento se coloca en la posición de memoria que actualmente es la parte superior de la pila. Después de realizar la operación “push”, la pila se expande hacia arriba para incluir el nuevo elemento.

Pop

Se utiliza para eliminar el elemento que está en la parte superior de la pila. Esto implica eliminar el elemento de la posición de memoria actualmente en la parte superior de la pila y ajustar la pila hacia abajo para excluir ese elemento. Después de realizar la operación “pop”, la parte superior de la pila se refiere al elemento que estaba debajo eliminado,

Estas operaciones se utilizan en conjunto para gestionar la pila de memoria de manera eficiente. Por lo general, se combinan con la gestión de punteros para realizar un seguimiento adecuado de la parte superior de la pila y garantizar que las operaciones “push” y “pop” se realicen en el lugar correcto.

3.- Ventajas

- Eficiencia en la gestión de memoria: La pila permite una asignación y liberación rápida de memoria, ya que simplemente se agrega o elimina elementos al principio de la pila.
- Facilita la gestión de llamadas a funciones: La pila es esencial para gestionar llamadas a funciones y proporcionar el flujo de control adecuado al regresar de las funciones.
- Conservación automática de memoria: No es necesario que el programador gestione manualmente la asignación y liberación de memoria para las variables locales.
- Soporte para recursividad: La pila es crucial para gestionar las llamadas recursivas de funciones, ya que cada llamada a una función se gestiona mediante la reserva de espacio en la pila.

4. BIBLIOGRAFÍA

Listado de imágenes:

Img. 01 - The Concept of Heap and Its Usage in Embedded Systems - Open4Tech

Bibliografía:

1. *GeeksforGeeks.. Memory Stack Organization in Computer Architecture. Extraido de Memory Stack Organization in Computer Architecture - GeeksforGeeks*

2. *EximiaCo. Qué es y cómo funciona la memoria "Stack" en .NET (incluida una descripción general sobre StackOverflowException). Extraido de Qué es y cómo funciona la memoria "Stack" en .net (incluida una descripción general sobre StackOverflowException) - EximiaCo*

3. *Hektor Profe. . Gestión de la memoria (Stack y Heap) | Apuntes lenguaje C++. Extraido de Gestión de la memoria (Stack y Heap) | Apuntes lenguaje C++ | Hektor Profe*