Tarea Semana 7 – Compiladores

# Tema: Entorno de Ejecución en los Compiladores

## Entorno de Ejecución

El entorno de ejecución en un compilador es el conjunto de mecanismos que se encargan de gestionar cómo se ejecutan los programas en tiempo de ejecución. Incluye aspectos como la asignación de memoria, el acceso a variables, y la forma en que se pasan los parámetros. Este entorno permite que el código que fue traducido por el compilador se ejecute correctamente sobre una máquina real o virtual.

## Organización de la Memoria en Tiempo de Ejecución

Durante la ejecución de un programa, la memoria se organiza en distintas áreas que cumplen funciones específicas:

- Código (Text Segment): Contiene las instrucciones del programa que fueron compiladas. Es de solo lectura.

- Heap: Se utiliza para almacenar datos que se asignan dinámicamente durante la ejecución. Su tamaño puede crecer o reducirse.

- Stack: Guarda las variables locales, los parámetros de funciones y las direcciones de retorno. Su crecimiento es hacia abajo (de direcciones altas a bajas).

- Datos Globales (Data Segment): Guarda las variables globales e información inicializada antes de la ejecución.

- Datos No Inicializados (BSS Segment): Se usa para variables globales que no tienen un valor inicial definido.

## Estrategias de Asignación de Memoria

Existen diferentes estrategias para manejar la memoria durante la ejecución:

- Asignación Estática: Se determina en tiempo de compilación. Las variables globales o locales que no cambian de tamaño suelen usar esta estrategia.

- Asignación Automática (o Dinámica en el Stack): Se realiza cuando una función es llamada. Las variables locales se crean y destruyen automáticamente al entrar y salir de la función.

- Asignación Dinámica (en el Heap): Permite asignar memoria de forma flexible, por ejemplo, usando funciones como malloc o new en algunos lenguajes. Es el programador quien se encarga de liberar esa memoria cuando ya no se necesita.

## Acceso a Variables Locales, No Locales y Globales

- Variables Locales: Son aquellas declaradas dentro de una función o bloque. Se accede a ellas usando el puntero del stack, ya que su posición está relacionada con la activación actual.

- Variables No Locales: Son variables que no están en el bloque actual, pero pertenecen a funciones que envolvieron la actual. En lenguajes como Pascal o funciones anidadas en otros lenguajes, se accede a ellas usando enlaces estáticos o dinámicos.

- Variables Globales: Están disponibles en todo el programa. Se ubican en una región fija de memoria y se accede a ellas directamente mediante su dirección global.

## Paso de Parámetros

Existen varias formas de pasar parámetros a funciones:

- Por valor: Se copia el valor del argumento a una nueva variable local. Lo que se cambie en la función no afecta al valor original.

- Por referencia: Se pasa la dirección de la variable, permitiendo que los cambios dentro de la función afecten al valor original.

- Por valor-resultante: Se copia el valor al inicio y se devuelve el resultado al final (usado en algunos lenguajes como Ada).

- Por nombre: Se sustituye cada uso del parámetro en la función por el argumento original. Es una técnica menos común y más costosa.

## Conclusión

El entorno de ejecución es esencial para que los programas se comporten correctamente al ejecutarse. Desde la asignación de memoria hasta la forma en que se accede a las variables y se pasan los parámetros, todo debe estar cuidadosamente organizado por el compilador y soportado por el entorno en tiempo de ejecución. Entender estos conceptos permite escribir código más eficiente y entender mejor cómo funciona internamente un programa.