Tarea 3

Compiladores

**Relación de una estructura de ejecución de tiempo en la tabla de símbolos**

Casi todos los programas de ordenador escritos están en lenguajes de alto nivel, que está un poco más cerca del inglés que una secuencia de 0 y 1 que es el lenguaje de la máquina, por lo que los humanos tenemos un tiempo más fácil de entender y trabajar con el código. Para convertir este código de programa en un archivo ejecutable que puede ser leído y ejecutado por una máquina, utilizamos un compilador.

Ahora, cuando se utilizan los términos tiempo de ejecución y tiempo de compilación, lo primero que viene a la mente son los errores de tiempo de compilación y los errores de tiempo de ejecución:

Compilar errores de tiempo: Cuando alimentamos un montón de texto al compilador para convertirlo en código máquina.

Qué puede salir mal en tiempo de compilación:

Errores de sintaxis

Errores tipográficos

(Rara vez) el compilador se bloquea

Si el compilador tiene éxito, ¿qué sabemos?

El programa estaba bien formado, un programa significativo en cualquier idioma.

Es posible iniciar la ejecución del programa ya que se encuentra en una forma legible por la máquina. (El programa puede fallar inmediatamente, pero al menos podemos intentarlo.)

¿Cuáles son las entradas y salidas?

La entrada era el programa que se estaba compilando, además de cualquier archivo de cabecera, interfaces, librerías u otro vudú que necesitara importar para ser compilado.

Se espera que la salida sea código ensamblador o código objeto reubicable o incluso un programa ejecutable. O si algo sale mal, la salida es un montón de mensajes de error.

Errores de ejecución:

Lo que puede salir mal son los errores de ejecución:

División por cero

Desfirmación de un puntero nulo

Se está quedando sin memoria

También puede haber errores que son detectados por el propio programa:

Intentar abrir un archivo que no existe

Intentar encontrar una página web y descubrir que una supuesta URL no está bien formada

Si el tiempo de ejecución tiene éxito, el programa finaliza (o continúa) sin fallar.

Otro escenario en el que se pueden utilizar los dos términos se refiere a cuándo exactamente se compila el código fuente o se convierte en código máquina: una vez justo después de escribir el código, es decir, programas compilados, o cada vez justo antes de que el programa necesite ejecutarse, es decir, programas interpretados, o compilados en tiempo de ejecución:

En un programa compilado (los ejemplos son C/C++ y Fortran):

El código fuente se introduce en otro programa (normalmente un compilador), que produce un programa ejecutable (o un error).

El ejecutable se ejecuta (haciendo doble clic en él, o escribiendo su nombre en la línea de comandos)

Las cosas que suceden en el primer paso se dicen que suceden en "tiempo de compilación", las cosas que suceden en el segundo paso se dicen que suceden en "tiempo de ejecución".

**Etapa de la construcción de una tabla de símbolos**

Una tabla de símbolos es una implementación de una estructura de datos utilizada por un traductor de idiomas, como un compilador o un intérprete, en la que cada identificador (también conocido como símbolo) en el código fuente de un programa se asocia con información relacionada con su declaración o aparición en la fuente.

Una tabla de símbolos puede existir sólo durante el proceso de traducción, o puede estar incrustada en la salida de ese proceso, como en un archivo de objeto ABI para su posterior explotación.

Existen numerosas estructuras de datos disponibles para implementar tablas. Se pueden utilizar árboles, listas lineales y listas auto-organizadas para implementar una tabla de símbolos. A la tabla de símbolos se accede por la mayoría de las fases de un compilador, comenzando con el análisis léxico y continuando con la optimización.

Un compilador puede usar una tabla de símbolos grande para todos los símbolos o usar tablas de símbolos jerárquicas separadas para diferentes alcances. El tiempo de búsqueda en tablas hash es independiente del número de elementos almacenados en la tabla, por lo que es eficiente para un gran número de elementos. También simplifica la clasificación de los literales en formato tabular.