



UNIVERSIDAD DE LA SIERRA SUR

Licenciatura en Informática

Programación de Sistemas

Tema: Traductores e intérpretes

Profesor: Oswaldo Rey Ávila Barrón

Grupo: 406

Fecha de entrega: 4 marzo del 2021

Índice de contenido

1	Traductores e interpretes.....	1
1.1	Traductor.....	1
1.2	Tipos de traductores.....	1
1.2.1	Compiladores.....	1
1.2.2	Intérprete.....	2

Índice de figuras

Figura 1: Esquema básico de un traductor.....	1
Figura 2: Esquemá de traducción /ejecución de un programa interpretado.....	2

1 Traductores e intérpretes

En los siguientes párrafos se describirán las definiciones de traductor y de intérprete.

1.1 Traductor.

Es un programa que traduce o convierte desde un texto o programa escrito en un lenguaje fuente hasta un texto o programa equivalente escrito en un lenguaje destino produciendo, si es necesario, un mensaje de error. Los traductores engloban tanto a los compiladores (en los que el lenguaje destino suele ser código máquina) como a los intérpretes (en los que el lenguaje destino está constituido por las acciones atómicas que puede ejecutar el intérprete). (Gálvez Rojas, 2005)

En la Figura 1 se muestra el esquema básico que compone un compilador/intérprete.

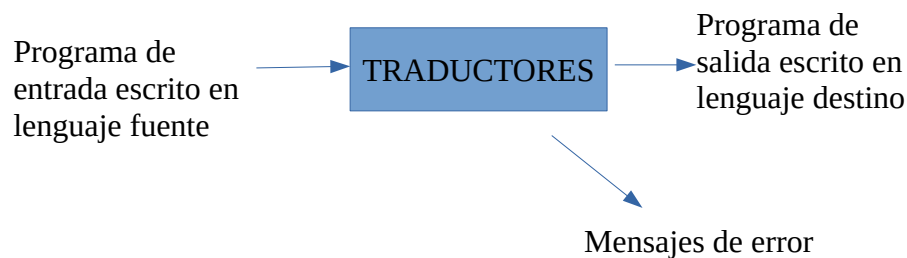


Figura 1: Esquema básico de un traductor

1.2 Tipos de traductores

Desde el origen de la computación, ha existido una diferencia muy grande entre la forma en que las personas expresan sus necesidades y la forma en que una computadora es capaz de interpretar sus instrucciones. Los traductores han intentado salvar este abismo para facilitarles el trabajo a los humanos, lo que ha llevado a aplicar la teoría de autómatas a diferentes campos y áreas concretas de la informática, provocando los distintos tipos de traductores. (Gálvez Rojas, 2005)

1.2.1 Compiladores

Es aquel traductor que tiene como entrada una sentencia en lenguaje formal y como salida tiene un archivo ejecutable, es decir, realiza una traducción de un código de alto nivel a código máquina (también se entiende por compilado aquel programa que proporciona un archivo objeto en lugar del ejecutable final). (Gálvez Rojas, 2005)

1.2.2 Intérprete

Es como un compilador, solo que la salida es una ejecución. El programa de entrada se reconoce y ejecuta a la vez. No se produce un resultado físico (código máquina) sino lógico (una ejecución). Algunos ejemplos de lenguajes interpretados son SNOBOL (StriNg Oriented SimBOlyc), LISP (LISt Processing), BASIC (Beginner's), etc.

Su principal ventaja es que permiten una fácil depuración. Entre las desventajas se puede mencionar la lentitud de ejecución, ya que al ejecutar a la vez que se traduce no puede aplicarse un alto grado de optimización; por ejemplo, si el programa entra en un bucle y la optimización no esta afinada, las mismas instrucciones se interpretarán y ejecutarán una y otra vez, enlenteciendo la ejecución del programa. Otra desventaja es que durante la ejecución, el interprete debe residir en memoria, por lo que consumen mas recursos.

Además de que la traducción optimiza el programa acercándolo a la máquina los lenguajes interpretados tienen las características de que permiten construir programas que se pueden modificar a sí mismos.

Algunos lenguajes intentan juntar las ventajas de los compiladores y de los interpretes y evitar sus desventajas; son los lenguajes pseudointerpretados. En estos, el programa fuente pasa por un pseudocompilador que genera pseudoejecutable. Para ejecutar este pseudoejecutable se le hace pasar por un motor de ejecución que lo interpreta de manera relativamente eficiente. Esto tiene la ventaja de la portabilidad, ya que el pseudoejecutable es independiente de la maquina en que vaya a ejecutarse, y vasta con que en dicha máquina se disponga del motor de ejecución apropiado para poder interpretar cualquier pseudoejecutable. El ejemplo actual más conocido lo constituye el lenguaje Java; también son pseudointerpretadas algunas versiones de pascal y de COBOL (Common Bussiness Oriented Language) (Gálvez Rojas, 2005)

La figura 2 muestra los pasos a seguir en estos lenguajes para obtener una ejecución.

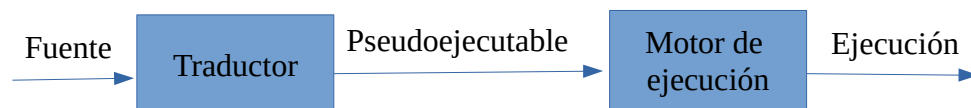


Figura 2: Esquema de traducción /ejecución de un programa interpretado

1.2.3 Intérprete de comandos

Un intérprete de comandos traduce las sentencias simples a invocaciones a programas de una biblioteca. Se utilizan especialmente en los sistemas operativos (la shell de Unix es un intérprete de comandos). Los programas invocados pueden residir en el kernel (núcleo) del

sistema o estar almacenado en algún dispositivo externo como rutinas ejecutables que se traen a memoria bajo demanda.

En el interprete de comandos Bash de Linux, cuando el usuario ejecuta el comando `ls`, la shell pide al núcleo de Linux cargar en memoria el archivo `/bin/ls`.

Bibliografía

Gálvez Rojas, S. (2005). *Java a Tope: Traductores Y Compiladores Con Lex/yacc, Jflex/cup Y Javacc.*