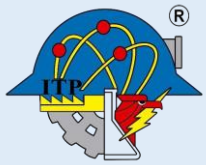


INSTITUTO TECNOLOGICO DE PACHUCA

LENGUAJES Y AUTOMATAS I

1.2 FASES DE UN COMPILADOR

PROFESOR: ING. RODOLFO LAZCANO BAUME

ALUMNO: ABISAI CALVA MELCHOR
19200190

TECNOLÓGICO NACIONAL DE MÉXICO

Fases de un compilador

En esta ocasión vamos a hablar sobre las fases del proceso de compilación, el cual es un proceso secuencial, esto es, que no se puede avanzar a la siguiente fase si no se ha aprobado satisfactoriamente la fase previa. Los compiladores son programas de computadora que traducen de un lenguaje a otro. Al igual los compiladores se componen internamente de varias etapas, o fases, que realizan operaciones lógicas.

--Análisis Léxico
– Análisis Sintáctico
– Análisis Semántico
– Generación y Optimización de código intermedio
– Generación de código objeto

Ventajas de un compilador	Desventajas de un compilador
• la ejecución del código traducido suele ser más rápida. • solo el desarrollador debe tener el compilador, el usuario final puede usar el código sin él. • el código traducido se almacena en lenguaje máquina, ya que es muy difícil de entender, es probable que tus propios inventos	• la compilación en si misma puede llevar mucho tiempo; es posible que no puedas ejecutar tu código inmediatamente después de cualquier modificación. • tienes que tener tantos compiladores como plataformas de hardware en los que deseas que se ejecute tu código

ANALISIS

ANALISIS LEXICO

"Es un conjunto de símbolos utilizados para formar palabras en un determinado idioma"

Por ejemplo: El alfabeto latino para el inglés y el español; el alfabeto cirílico para el ruso y el kanji para el japonés, etc.

Verificación de la pertenencia de cada símbolo presente en el código fuente respecto al lenguaje en el cual se presume fue hecho.
Verificación y clasificación de cada componente léxico (cada palabra) presente en el código fuente pertenezca al lenguaje y esté escrito de acuerdo a los patrones de cada uno de ellos.

Creación de la tabla de símbolos registrando en ella todos los identificadores (variables, constantes, nombres de funciones o métodos) presentes en el código fuente.

ANALISIS SINTACTICO

En esta fase de la compilación se verifica que las construcciones correspondan con la gratica del lenguaje en el cual se presume fue escrito el programa.

Ejemplo:

Si estoy realizando un lenguaje de programación donde estoy diciendo que para declarar una variable debo de escribir :

Tipo de dato+identificador+=+valor+;

int edad = 10 ; // es una estructura sintáctacticamente correcta

y

edad 10= int; // no es una estructura sintáctacticamente correcta

ANALISIS SEMANTICO

Cumple con la tarea de verificar que el código guarde coherencia en la aplicación de sus estructuras, básicamente en tres aspectos:

Utilización de tipos de datos
Controles de flujo de instrucciones
Controles de ciclos

"Una variable entera puede asignarse a otra variable entera"

"Una variable String puede asignarse a otra variable String"

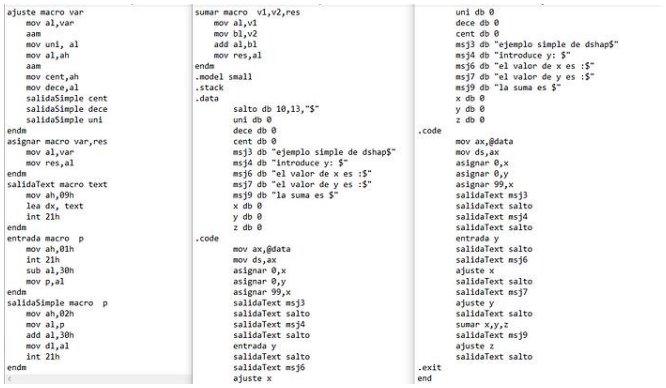
int x=10;// sintácticamente es correcta

String y =x;// sintácticamente es correcta

GENERACION DE CODIGO INTERMEDIO

```
=       0               x
=       0               y
=       99              x
>>      "ejemplo simple de dshap"
>>      "introduce y: "
=       <<              y
>>      "el valor de x es :"    x
>>      "el valor de y es :"    y
+       x       y       z
>>      "la suma es "   z
```

Realiza una primera traduccion del código en el cual elimina aspectos característicos del lenguaje de alto nivel pero sin entrar en compilaciones de un lenguaje de bajo nivel. En este punto, el código sigue teniendo independencia de la arquitectura de un hardware sobre el cial se ejecutará.



```
ajuste macro var
    mov al,var
    aam
    mov uni, al
    mov al,ah
    aam
    mov cent,ah
    mov dece,al
    salidaSimple cent
    salidaSimple dece
    salidaSimple uni
endm
asignar macro var,res
    mov al,var
    mov res,al
endm
salidaText macro text
    mov ah,09h
    lea dx, text
    int 21h
endm
entrada macro  p
    mov ah,01h
    int 21h
    sub al,30h
    mov p,al
endm
salidaSimple macro  p
    mov ah,02h
    mov al,p
    add al,30h
    mov dl,al
    int 21h
endm
```

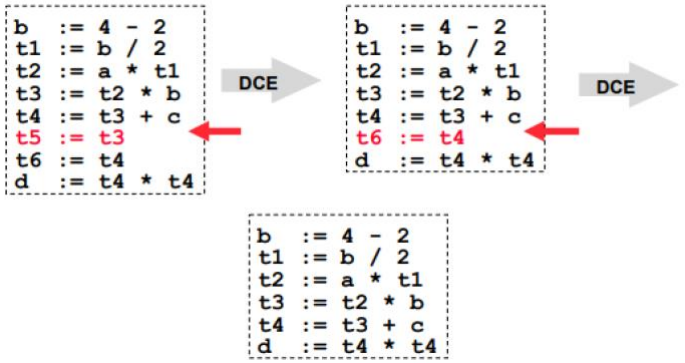
```
sumar macro  v1,v2,res
    mov al,v1
    mov bl,v2
    add al,bl
    mov res,al
endm
.model small
.stack
.data
        salto db 10,13,"$"
        uni db 0
        dece db 0
        cent db 0
        msj3 db "ejemplo simple de dshap$"
        msj4 db "introduce y: $"
        msj6 db "el valor de x es :$"
        msj7 db "el valor de y es :$"
        msj9 db "la suma es $"
        x db 0
        y db 0
        z db 0
.code
        mov ax,@data
        mov ds,ax
        asignar 0,x
        asignar 0,y
        asignar 99,x
        salidaText msj3
        salidaText salto
        salidaText msj4
        salidaText salto
        entrada y
        salidaText salto
        salidaText msj6
        ajuste x
        salidaText salto
        salidaText msj7
        ajuste y
        salidaText salto
        sumar x,y,z
        salidaText msj9
        ajuste z
        salidaText salto
.exit
end
```

Consiste en la última transformación que se realiza sobre el código. El resultado de esta fase es un programa ejecutable totalmente dependiente de la arquitectura del hardware sobre el cual se ejecutara.

Optimización de Código

En esta fase se optiene una versión mejorada pero equivalente del código, es decir, logrando en la medida de lo posible mejorar el desempeño del programa y el suso óptimo de los recursos como la memeria del CPU, pero sin alterar la funcionalidad del mismo.

CODIGO MAQUINA



CONCLUSIONES

Una conclusión sobre las fases de los compiladores es que, aunque pueden variar en número y en la forma en que se organizan, típicamente siguen un conjunto de pasos que incluyen análisis léxico, análisis sintáctico, análisis semántico, generación de código intermedio, optimización de código y generación de código final. Cada fase cumple un propósito específico para transformar el código fuente en un programa ejecutable, lo que permite a los desarrolladores escribir código en lenguajes de alto nivel y luego ejecutarlo en computadoras.

Referencias

https://vicente-aguilera-perez.medium.com/estructura-general-de-un-compilador-df97892f91c3

https://lenguajesyautomatasitsh.blogspot.com/2015/02/17-fases-de-un-compilador.html