

Tarea 1

Cabrera López Oscar Emilio

2019-03-10

1. Seleccione un compilador junto con un IDE y haga una lista de los programas auxiliares al compilador y una breve descripción de lo que hacen.

- IDE: Neovim
- Compilador: Clang
- Programas auxiliares:
 - clang-cpp: El preprocesador de clang
 - clang-format: Da formato al código fuente
 - clang-tidy: Da sugerencias para corregir y prevenir comportamientos no deseados en los programas
 - clangd: Servidor que provee análisis estático del código fuente a los IDEs

2. Dada la siguiente instrucción en C, realice el proceso de compilación dando una breve descripción de cada fase de compilación.

`a[i + 1] = a[i] + a[j + i]`

1. Análisis léxico: generación de tokens, creación de la tabla de símbolos

<id, 0>
<op, [>
<id, 1>
<op, +>
<num, 1>
<op,]>
<op, =>
<id, 0>
<op, [>
<id, 1>
<op,]>

```

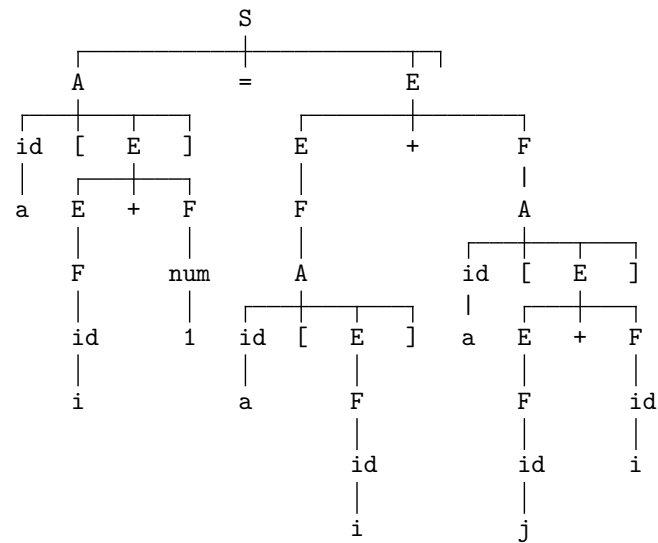
<op, +>
<id, 0>
<op, [>
<id, 2>
<op, +>
<id, 1>
<op, ]>

```

Tabla de símbolos

Pos	ID	Tipo
0	a	
1	i	
2	j	

2. Análisis sintáctico: creación del árbol sintáctico

$$\begin{aligned} S &\rightarrow A = E; \\ A &\rightarrow A[E] \mid \text{id}[E] \\ E &\rightarrow E + F \mid F \\ F &\rightarrow \text{id} \mid \text{num} \mid A \end{aligned}$$


falta un ; error

3. ¿Qué es un compilador?

Un compilador es un programa que traduce un programa fuente escrito en un lenguaje de alto nivel a otro lenguaje.

4. ¿Qué es un intérprete?

Un intérprete es un programa que traduce un programa fuente directamente a código máquina cada vez que se ejecuta.

5. ¿Qué es la etapa de síntesis y qué otro nombre recibe?

La etapa de síntesis toma el árbol sintáctico anotado y lo transforma en código objeto.

6. ¿Qué es la etapa de análisis y qué otro nombre recibe?

La etapa de análisis se encarga de revisar que el programa tenga la estructura adecuada y pueda ser traducido a código objeto.

7. ¿Qué es una máquina virtual?

Es un tipo de interprete que realiza previamente una compilación del programa fuente en código objeto intermedio, que luego es ejecutado en un interprete que traduce el código intermedio en código máquina.

8. Diferencias entre un intérprete, un compilador y una máquina virtual.

Compilador	Intérprete	Máquina virtual
- Se ejecuta 1 vez - Genera código objeto	- Se ejecuta cada vez - Genera código máquina	- Se ejecuta cada vez - El compilador genera código intermedio y el intérprete código máquina
- Detecta errores en tiempo de compilación	- Detecta errores en tiempo de ejecución	- Detecta errores en tiempo de compilación y de ejecución

9. Mencione algunos tipos de compiladores y sus características.

- Compilador *source to source*: Recibe un programa fuente escrito en un lenguaje de alto nivel y lo traduce en un programa escrito en otro lenguaje de alto nivel.
- Compilador *cruzado*: Es un compilador que toma un programa fuente en un lenguaje de alto nivel y genera código objeto para una arquitectura distinta de la arquitectura en la que se ejecuta.
- Metacompilador: Es un compilador de compiladores, toma la definición de un lenguaje y usualmente genera los analizadores léxicos y sintácticos.

10. Describa el proceso que sigue un sistema de procesamiento de lenguaje.

Un sistema de procesamiento de lenguaje se divide en 3 o 4 etapas:

1. Preprocesado: Si el lenguaje cuenta con un metalenguaje para creación de macros e inclusión de archivos, en esta etapa las macros son expandidas, los archivos necesarios son añadidos al programa fuente y los comentarios son eliminados.
2. Compilación: En esta etapa se recibe el programa fuente ya preprocesado y se realiza el análisis léxico, sintáctico, semántico, se realizan optimizaciones y se traduce a un programa objeto en lenguaje ensamblador.
3. Ensamblador: En esta etapa se traduce el lenguaje ensamblador en código máquina relocizable.
4. Enlazador/Cargador: En esta etapa el objeto en código máquina es unido a todas las utilidades necesarias para ser cargado y ejecutado por el sistema operativo en el que se ejecutará.

11. De algunos ejemplos de errores: léxico, sintácticos y semánticos.

- Léxicos:

```
/* comentario sin terminar
1e+$
0x99g
char c = 'hola';
```

- Sintácticos:

```
int entero = 3;
if {entero} { // se esperaba '('
    bar() // punto y coma faltante
    return;
}
```

- Semánticos:

```
if (a < b) { // variables no definidas
    x++ += f(&x) + g(&x); // comportamiento sin definir
}
```