



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

Ingeniería en sistemas computacionales

FECHA: 11/10/2020

TÍTULO: Practica 4

ASIGNATURA: COMPILADORES

PROFESOR: Rafael Norman Soucedo Delgado

GRUPO: 3CV5

NOMBRE:

Torres Moreno David

Introducción:

Hoy en día comprender cómo fue la creación de compiladores es de vital importancia ya que este es una pieza clave en el desarrollo de software y también una gran brecha de seguridad para los sistemas. Es tan difícil optimizar los compiladores en forma apropiada, que nos atrevemos a decir que ¡ningún compilador optimizador está libre de errores! (Aho, A. V., Lam, M. S., & Ullman, R. S. J. D., 2008, p. 16)
Esta es una razón para aprender sobre cómo funcionan los compiladores.

La intención de la práctica es desarrollar un analizador léxico de lenguaje con la finalidad de aprender cómo funciona internamente el lenguaje, para esto se utilizarán varias herramientas que nos facilitarán la construcción del analizador léxico. La herramienta a utilizar en esta práctica será flex 2.6.4. cuyo generador de código será compilado por gcc 9.3.0.

Desarrollo:

Ejemplos del lenguaje.

```
int main(){  
    printf("hola mundo");  
    return 0;  
}
```

```
extern char **environ;  
int  
main(int argc, char *argv[])  
{  
    int j;  
    char **ep;  
    clearenv(); /* Erase entire environment */  
    for (j = 1; j < argc; j++)  
        if (putenv(argv[j]) != 0)  
            errExit("putenv: %s", argv[j]);  
    if (setenv("GREET", "Hello world", 0) == -1)  
        errExit("setenv");  
    unsetenv("BYE");  
    for (ep = environ; *ep != NULL; ep++)  
        puts(*ep);  
    exit(EXIT_SUCCESS);  
}
```

```

void menu() {
    printf ("\nIntroduce una opción de las siguientes:\n");
    printf ("1.- Sumar\n");
    printf ("2.- Restar\n");
    printf ("3.- Multiplicar\n");
    printf ("4.- Dividir\n");
    printf ("0.- Salir\n");
    printf ("Opcion: ");
}

float suma(float a, float b) {
    return a+b;
}

float resta(float a, float b) {
    return a-b;
}

float multiplicar(float a, float b) {
    return a*b;
}

float dividir(float a, float b) {
    return a/b;
}

```

Clases léxicas:

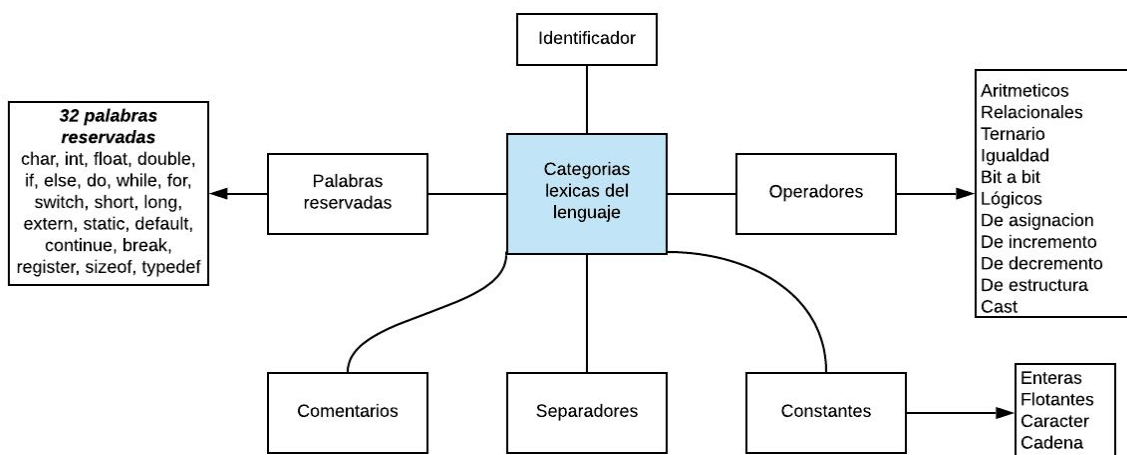


Imagen 1: Clases léxicas obtenidas de Kernighan, B. W., & Ritchie, D. M., 1991

Expresiones regulares y codificación.

En esta parte se obtuvieron las expresiones regulares de ANSI C., 1995 el archivo con las expresiones regulares obtenidas está en lexico.l

Pruebas:

```
float x = 3.14f;  
char a = '\n';  
int y = (int) x;  
int anio=1992;  
int is21= anio>2000 ? 1 : 0 ;
```

Test1.c

```
yo@yo-X556UQKyo:~/Documents/ESCUELA/compiladores/practica4$ cat test1.c | ./a.out  
<Palabra_reservada> <Identificador> <Operador-ASIGNACION> <Constante-Flotante><Punto-Coma>  
<Palabra_reservada> <Identificador> <Operador-ASIGNACION> <Constante-Cadena><Punto-Coma>  
<Palabra_reservada> <Identificador> <Operador-ASIGNACION> <Operador-CAST><Palabra_reservada><Operador-CAST> <Identificador><Punto-Coma>  
<Palabra_reservada> <Identificador><Operador-ASIGNACION><Constante-Entera-DECIMAL><Punto-Coma>  
<Palabra_reservada> <Identificador><Operador-ASIGNACION> <Identificador><Operador-RELACION><Constante-Entera-DECIMAL> <Operador-TERNARIO> <Constante-Entera-DECIMAL> <Operador-TERNARIO> <Constante-Entera-DECIMAL> <Punto-Coma>
```

Lanzando prueba Test1.c

Conclusiones:

Concluyó que esta práctica fue muy interesante y sería útil para hacer un sitio de mecanografía para programadores online, algo parecido al sitio <https://typing.io> ya que se podría usar flex para resaltar los colores del código fuente a teclear y que el código fuente sea importado por el usuario.

Referencias

Aho, A. V., Lam, M. S., & Ullman, R. S. J. D. (2008). *Compiladores: Principios, técnicas y herramientas* segunda edición Ed.

ANSI C. (1995) ANSI C grammar, Lex specification. Retrieved November 13, 2020, from <https://www.lysator.liu.se/c/ANSI-C-grammar-l.html>

Kernighan, B. W., & Ritchie, D. M. (1991). *El lenguaje de programación C*. Pearson Educación.