Proyecto Final de la clase de Compiladores
Manual de Usuario

Lenguaje LEAD 2.0

Profesores:
Ing. Elda Guadalupe Quiroga González
Dr. Héctor Gibrán Ceballos Cancino

Joel Hiram Chávez Verástegui

Monterrey, Nuevo León, México
22 de Noviembre del 2022

Visión General del Lenguaje de Programación

Usar el lenguaje de programación LEAD 2.0 es muy sencillo, no se necesita nada especial fuera de python instalado en la computadora.

Crear un archivo para programar en LEAD 2.0 es tan sencillo como crear un archivo .txt y empezar a programar ahí. Un ejemplo de un programa en LEAD es el siguiente:

```
program LEAD;
var
    int: g1;
function float test(float x)
{
    write(x);
    return(2.1);
}
main()
var
    int: main1;
    float: mf1;
{
    main1 = 4;
    call(mf1 = test(mf1));
    write(mf1);
}
```

En este programa se pueden observar ciertos aspectos a considerar cuando se programe en el lenguaje de programación LEAD.

- Al iniciar la programación, se tiene que escribir program y el id del programa seguido de un punto y coma
- Las variables globales se definen se definen después de lo explicado en el enunciado anterior
- Al crear una función es necesario escribir la palabra function, seguido del tipo de función void, int o float y los parámetros de la misma
- Las variables locales se definen después de cerrar los paréntesis de los parámetros de una función y antes de abrir los corchetes
- Para hacer una llamada a una función, es necesario usar call() y dentro del paréntesis hacer la asignación o la llamada a la función
- Los tipos de variables soportadas para definir son int y float

Vectores

Para revisar la creación de vectores en LEAD veamos el siguiente ejemplo:

```
program LEAD;
var
    int: t1;

main()
var
    int: prob1, vec[5];
{
    prob1 = 10;
    vec[0] = 5;
    vec[4] = prob1 + 15;
    vec[1] = vec[0] * vec[4];
    write(vec[1]);
}
```

Los vectores se definen en LEAD como un id, seguido de medio cuadro ([ ), un número y el cierre del cuadro ( ]) y se usa un índice con base 0 (esto en el ejemplo anterior de 5 sería: 0, 1, 2, 3 y 4). Se pueden hacer expresiones con los arreglos. Sin embargo, no se puede usar una expresión para acceder a un índice de un arreglo.

Llamadas a funciones

Para revisar las llamadas a funciones en LEAD veamos el siguiente ejemplo:

```
program LEAD;
var
    int: g1;

function float test(float x)
{
    return(2.1);
}

main()
var
    float: mf1;

{
    call(mf1 = test(mf1));
    write(mf1);
}
```

Para hacer el llamado a una función se tiene que usar la palabra reservada call. En caso de que sea una función que regrese un valor, se asigna a una variable dentro del alcance ("scope"). La definición de las funciones se hace primeramente tecleando function, seguido del tipo de función que se quiera declarar, el ID y los parámetros. Estos últimos pueden ser ninguno, uno o varios parámetros. Se deben definir con su tipo.

Condicionales y Ciclos

Para revisar los condicionales y los ciclos en LEAD veamos el siguiente ejemplo:

```
program LEAD;
var
    int: t1;

function void fact(int num)
var
    int: fact;
{
    write("La respuesta al factorial cíclico es: ");
    if(num < 0){
        write("No existe el factorial negativo");
    }

    if(num == 0){
        write(1);
    }else{
        fact = 1;
        while(num > 1){
            fact = fact * num;
            num = num - 1;
        }

    write(fact);
    }
}

main()
var
    int: main1;
{
    read(main1);
    call(fact(main1));
}
```

En este ejemplo se puede ver todo lo visto anteriormente, con el añadido de que ahora se usa un if, if-else y un while. La estructura es muy similar entre estos estatutos, se usa la palabra reservada if o while seguido de una expresión booleana. Esta expresión booleana no soporta and's y or's. Después de esto se abren corchetes y se pueden poner estatutos dentro de los mismos. En caso de que haya necesidad de tener un else, al cerrar el corchete del if se escribe el else y se vuelven a abrir corchetes, dentro de los cuales se pueden hacer estatutos