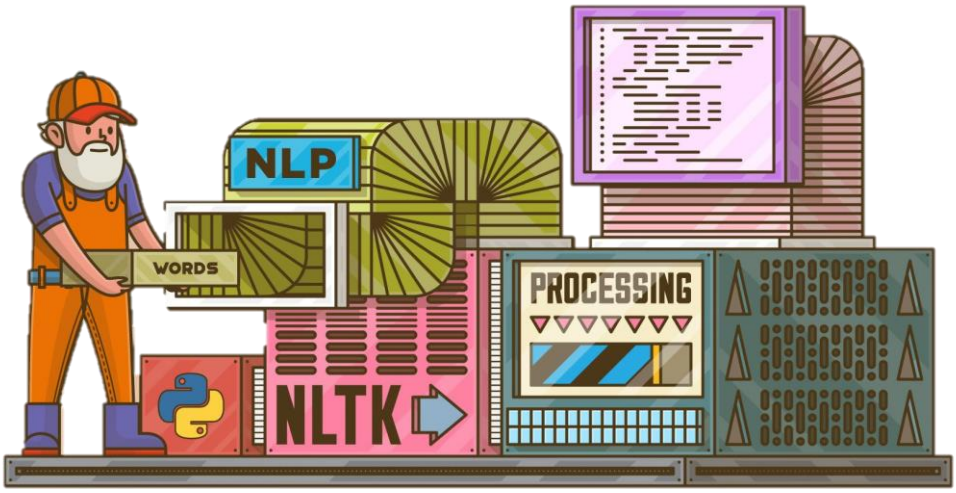
UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Escuela de Ingeniería Informática

Curso: 21-22

22/02/2022

Procesadores de Lenguajes



1 – DEFINICIÓN DEL LENGUAJE

Fabián Alfonso Beirutti Pérez - fabian.beirutti101@alu.ulpgc.es

Índice

INTRODUCCIÓN

Esta primera entrega de la práctica de curso de la asignatura consiste en la definición de un lenguaje de programación de alto nivel cuyo compilador se implementará por partes en las siguientes fases y/o entregas. Como mínimo debe incorporar declaraciones de variables numéricas, instrucciones de asignación de expresiones, estructuras de control, rutinas recursivas, así como operaciones de salida para variables y ristras.

IMPORTANTE: De momento, como el trabajo será desarrollado por un único integrante, el lenguaje será limitado al trabajo de operaciones de números enteros. Además, se permitirá escribir por pantalla el valor de una variable o ristra de caracteres. Una vez desarrollado lo descrito aquí, es posible que se implemente funciones adicionales.

DEFINICIÓN

El lenguaje a definir estará basado en C. Toda línea, a excepción de la apertura y cierre de funciones, acabarán en ';', de esta manera será más fácil identificar cuando acaba una instrucción.

```
instruction;
```

El código desarrollado se detallará en el **método principal** o ***main***. En él, se permiten las llamadas a otros fragmentos de código que realicen diferentes operaciones o ***function***. La finalidad de esta decisión es permitir desarrollar un código limpio, legible, modular y de fácil mantenimiento.

El ***main*** acaba con la última instrucción que se encuentre en él. En cambio, una ***function*** siempre devuelve un entero, por tanto, su última línea y/o instrucción será un ***return***.

```
function funcion() {
    instruction;
    return 0;
}

main() {
    instruction;
    funcion();
    instruction;
}
```

Como se puede ver, las **funciones** y/o conjunto de instrucciones externas al *main*, han de ser declaradas y/o implementadas antes de él.

Existe la posibilidad de guardar valores enteros, esto es, **variables**. Solo se permite la declaración de variables locales en todo el programa. Para declarar **variables numéricas** y/o guardar valores enteros se puede seguir el ejemplo a continuación:

```
int a = 1;
```

Los **operadores** y **símbolos** aceptados por el lenguaje son los siguientes:

- **Asignación '='** -> Asigna un valor numérico entero a una variable.
- **Suma '+'** -> Suma entre dos valores enteros o variables.
- **Resta '-'** -> Resta entre dos valores enteros o variables.
- **Multiplicación '*'** -> Multiplica dos valores enteros o variables.
- **División '/'** -> Divide dos valores enteros o variables. El resultado devuelto es el cociente de la división.
- **Resto '%'** -> Calcula el módulo de dos valores enteros o variables. El resultado devuelto es el resto de la división entre dichos valores.

```
int a = 1 operator 3;
```

Para realizar **comparaciones**, se pueden combinar diferentes símbolos tal que:

- **Igualdad '=='** -> Comprueba si el valor de dos números o variables es el **mismo**.
- **Distinto '!='** -> Comprueba si dos valores son **diferentes**.
- **Menor que '<'** -> Comprueba si el valor de la **izquierda** es **menor** que el de la **derecha**.
- **Mayor que '>'** -> Comprueba si el valor de la **izquierda** es **mayor** que el de la **derecha**.
- **Menor o igual que '<='** -> Comprueba si el valor de la **izquierda** es **menor o igual** que el de la **derecha.**
- **Mayor o igual que '>='** -> Comprueba si el valor de la **izquierda** es **mayor o igual** que el de la **derecha.**

Las **estructuras de control** son:

- **If – else** -> Si se cumple **condition**, se ejecutan las instrucciones a continuación, si no, las de debajo del **else**.

```
if (condition) {
    instruction;
} else {
    instruction;
}
```

- **<u>While</u>** -> Mientras se cumpla **condition**, se ejecutan las instrucciones de dentro.

```
while (condition) {
        instruction;
}
```

Para **mostrar por pantalla** el valor de una variable o una pequeña cadena de caracteres, se utiliza la instrucción **print**():

```
print("Hola mundo");
```

Para añadir **comentarios**, se añade lo siguiente '//'. Todo lo que se esté a la derecha en la misma línea será omitido por el compilador.

```
// Comment
```

EJEMPLO DE APLICACIÓN

```
function sum(x, y) {
        int res = x + y;
        return res;
}

// Programa que suma dos variables

main() {
        int x = 2;
        int y = 3;
        int res = sum(x, y);
        if (res == 5) {
                print("Resultado correcto");
        } else {
                print("Resultado incorrecto");
        }
}
```