

Universidad Rey Juan Carlos

Procesadores de
lenguajes



Práctica Obligatoria. Parte 1: Análisis léxico y
sintáctico

Autores: Sergio Paniego Blanco

Héctor Domínguez Diego

Descripción de la práctica

En la práctica hemos realizado la especificación léxica, en la que hemos detectado los identificadores, las constantes numéricas, que pueden ser del tipo real o decimal y con base decimal, octal o hexadecimal, las constantes literales y los comentarios.

Para la sintaxis, hemos definido las declaraciones de funciones y procedimientos, declaraciones de variables, asignaciones y llamadas a las funciones y procedimientos. Además de esto, hemos definido las sentencias de control de flujo if, for y while, y los tipos de datos para distintas estructuras y matrices.

Para el control de errores, hemos detectado si no se ha escrito un principio de comentario y se ha puesto un final de comentario, si se ha declarado mal un número hexadecimal, tanto dejándolo a medias, como si se ha puesto más de un signo o más de un punto y si se ha abierto de forma incorrecta un comentario o si se ha cerrado de forma incorrecta.

Casos de uso

Caso correcto 1

Declaramos un registro, una función con instrucciones simples y un procedimiento con instrucciones de flujo y en la función main hacemos uso de estos. Estas declaraciones se pueden observar en el fichero prueba1.c.

La salida tras compilar este fichero se encuentra en el fichero resultado1.txt, en él mostramos la ristra de tokens que se han detectado en el fichero prueba1.c.

Caso correcto 2

Declaramos una función en la que realizamos operaciones con las distintas bases y tipos de números y la función main donde hacemos uso de la función anterior.

La salida tras compilar este fichero se encuentra en el fichero resultado2.txt, en él mostramos la ristra de tokens que se han detectado en el fichero prueba2.c.

Caso correcto 3

Declaramos un registro, una función con instrucciones simples y un procedimiento con instrucciones de flujo y finalmente hacemos las llamadas a estos métodos en la función main.

La salida tras compilar este fichero se encuentra en el fichero resultado3.txt, en él mostramos la ristra de tokens que se han detectado en el fichero prueba3.c.

Caso correcto 4

Declaramos un registro, una función con instrucciones simples y un procedimiento con instrucciones de flujo que hace uso de las estructuras que hemos declarado y un procedimiento con instrucciones simples y finalmente hacemos las llamadas a estos métodos en la función main.

La salida tras compilar este fichero se encuentra en el fichero resultado4.txt, en él mostramos la ristra de tokens que se han detectado en el fichero prueba4.c.

Caso Error 1

Al asignar un valor a un número hexadecimal se escribe mal el valor del número y al compilar se muestra por pantalla el siguiente error:

```
\Error in line:21 Column: 12. 0x Hexadecimal number not completed.\
```

La salida tras compilar el fichero se encuentra en el fichero resultado5.txt, donde mostramos la ristra de tokens detectados, además de este error del fichero pruebaerror1.c

Caso Error 2

En un determinado comentario solo se ha introducido el cierre de comentario por lo que se avisa de que falta abrirlo.

```
\Error in line:33 Column: 4. */ Start part of comment missing.\
```

La salida tras compilar el fichero se encuentra en el fichero resultado6.txt, donde mostramos la ristra de tokens detectados, además de este error del fichero pruebaerror2.c

Caso Error 3

La apertura de comentario se ha declarado de forma incorrecta e indicamos que el símbolo detectado no coincide con la apertura de un comentario mientras que el cierre sí.

```
\Error in line:34 Column: 4. \ Starting of comment symbol not matching.\
```

La salida tras compilar el fichero se encuentra en el fichero resultado7.txt, donde mostramos la ristra de tokens detectados, además de este error del fichero pruebaerror3.c

Caso Error 4

Al asignar un valor a un número hexadecimal se han usado más de un signo a la vez.

```
\Error in line:21 Column: 12. 0x++34 More than one addition sign added.\
```

La salida tras compilar el fichero se encuentra en el fichero resultado8.txt, donde mostramos la ristra de tokens detectados, además de este error del fichero pruebaerror4.c