

Compiladores e Intérpretes

Manual de Desarrollo

Diego Marcovecchio (LU: 83815) Leonardo Molas (LU: 82498)

2 de Septiembre de 2010

Índice general

1. Implementación	2
2. Diseño general	3
2.1. Programa principal	3
2.2. LexAn	3
2.3. shlex	4

Capítulo 1

Implementación

El analizador léxico fue desarrollado utilizando la release 2.7 del lenguaje interpretado y multiplataforma **Python**¹. Se aprovecharon las herramientas propias de dicho lenguaje para el reconocimiento de expresiones regulares; en particular, fue utilizada la librería oficial **re**².

Para procesar el stream de caracteres del archivo de entrada, se utilizó como base el código fuente de la librería **shlex**³ de Python, realizándole todas las modificaciones necesarias para la eliminación de comentarios e interpretación del alfabeto propio de mini-pascal. El código de esta versión modificada se encuentra en el archivo **shlex.py** de la carpeta */src*.

El reconocimiento de los argumentos de entrada está provisto por la librería **argparse**⁴, y el esquema de lectura con buffering es soportado por Python mediante la librería oficial **io**⁵.

Por último, la generación del archivo ejecutable fue realizada mediante gracias a **py2exe**⁶, una extensión creada a partir del módulo distutils de Python que permite generar archivos binarios para Windows.

¹Para más información, dirigirse a la página oficial: <http://www.python.org/>

²La documentación oficial de **re** puede ser vista en: <http://docs.python.org/library/re.html>

³**shlex** es una librería de Python para procesar comandos de consola. La documentación de la versión original puede ser encontrada en: <http://docs.python.org/library/shlex.html>

⁴La documentación oficial de **argparse** puede ser encontrada en <http://docs.python.org/dev/library/argparse.html>

⁵La documentación oficial de **io** puede ser encontrada en <http://docs.python.org/tutorial/inputoutput.html>

⁶Para más información, dirigirse a la página oficial de py2exe: <http://www.py2exe.org/>

Capítulo 2

Diseño general

Detallaremos a continuación la arquitectura general del analizador léxico, describiendo brevemente cada una de las clases que lo componen. Además de estas clases, existe un archivo **setup.py** que no contiene código propio de la aplicación, y es simplemente utilizado para generar el ejecutable de Windows.

2.1. Programa principal

El programa principal, cuyo código fuente puede ser encontrado en el archivo **main.py** de la carpeta */src*, no contiene ninguna clase, y es simplemente un script que reconoce los argumentos de entrada utilizando la librería *argparse*, abre los archivos para lectura y escritura, y crea una instancia de la clase **LexAn**, el analizador léxico implementado.

A medida que **LexAn** devuelve los tokens uno a uno, el programa principal los imprime al archivo correspondiente (si no se especifica ninguno se utiliza *stdout*, por lo que la salida se imprime por pantalla), y en caso de encontrar algún error (es decir, si se atrapa una excepción *LexError*), el programa aborta inmediatamente la ejecución, detallando el error.

2.2. LexAn

La clase **LexAn**, cuyo código fuente puede ser encontrado en el archivo **lexan.py** de la carpeta */src*, procesa los lexemas provistos por **shlex** para transformarlos en tokens, que finalmente serán volcados al archivo de salida. Para reconocer los lexemas, la clase cuenta con dos pilares: una estructura de diccionario de Python (similar al hash de Nokia Qt, un arreglo asociativo en el que se mapean pares de valores (id, valor) y se accede mediante sus identificadores), y el reconocimiento de expresiones regulares utilizando la librería **re**. La clase permite además saber qué lexema y número de línea del archivo fuente está siendo analizando.

Los operadores y las palabras reservadas que cuentan con un token propio son cargadas inicialmente en el diccionario. Al momento de analizar un lexema, se realiza un chequeo para ver si dicho lexema es uno de los id del diccionario. En ese caso, se retorna inmediatamente el token asociado. En caso contrario, se compara el lexema con las expresiones regulares de identificador, número, y caracter (devolviendo el respectivo token en caso de match). Si ninguna de estas opciones tiene éxito, se dispara la excepción *LexError*, indicando que el lexema no pudo ser reconocido.

Por último, si al intentar obtener el siguiente lexema de **shlex** se captura la excepción *EOFError*, dicha excepción es propagada, informando que un comentario se mantuvo abierto hasta el final del archivo.

2.3. shlex

Ésta clase contiene el código de la librería **shlex** de Python (creada originalmente para procesar scripts de consola) modificado para que elimine los comentarios del archivo fuente a medida que lee los caracteres, y adaptado para los separadores y el alfabeto propio de mini-pascal. El procedimiento más importante de la clase devuelve un lexema del archivo fuente, y permite además saber en qué línea se encuentra dicho lexema.

Además de las modificaciones mencionadas, se agregó la excepción *EOFError*, que es lanzada cuando se encuentra el caracter indicando el fin de archivo mientras aún se esperaba procesar. Ésta excepción es aprovechada por LexAn para saber cuándo un comentario multilínea se mantiene abierto de manera errónea. Su código fuente puede ser encontrado en el archivo **shlex.py** de la carpeta */src*.