

TOKDEC: A Tool for Compiler Debugging

Peter M. Maurer
Department of Computer Science
Baylor University
Waco, TX 76798

1 Abstract

TOKDEC is a tool that helps create compiler debugging code by converting numeric tokens into character strings. It is meant to be used with the output of the “yacc” preprocessor.

2 Description

The TOKDEC tool converts a token header files into a program capable of converting numeric token-codes into character strings. It is meant to be used with the yacc preprocessor. When creating yacc specifications, it is common to define a number of tokens using a yacc statement similar to the following.

```
%token VARIABLE STRING CONSTANT
```

These tokens represent numeric codes that the yacc parser expects to receive from the lexical analyzer. To use these tokens in another program, it is necessary to invoke yacc with the `-d` option as follows.

```
yacc -d MyYaccFile.y
```

This will cause yacc to produce a `y.tab.h` file in addition to the usual `y.tab.c` file.

The `y.tab.h` file will contain a `#define` statement for each declared token. The file will look something like this:

```
#define VARIABLE 257  
#define STRING 258  
#define CONSTANT 259
```

For debugging purposes it is much more useful to display the original strings “VARIABLE” “STRING” and “CONSTANT” instead of the numeric constants. The `tokdec` tool makes this possible. This tool is capable of reading the `y.tab.h` file and creating a decoder program that will accept a numeric token as input and return the corresponding string as output. To use `tokdec`, invoke it as follows.

```
tokdec y.tab.h MyDecoder >MyDecoder.cpp
```

The argument `y.tab.h` is required. Any `.h` file containing `#defines` may be used. The second argument is the name of the function. It will be defined using the following prototype.

```
char *MyDecoder(int x);
```

The output of `tokdec` will appear on `stdout`. The file will compile either as a `.c` file or a `.cpp` file. This file must be added to your project, and a suitable prototype should be placed in an appropriate `.h` file. This will enable compiler debugging messages to contain the name of the token rather than just its number.

This software is released under the GNU open source license V 2.0.