Replacing Whitespaces using Lex

Irfan Sheikh

April 2021

1 The Lex File - `replace-whitespace.l`

```
%{
// C includes
#include <stdio.h>
%}

%option noyywrap

WS  [ \t]+

%%
{WS} {
        /* If a sequence of one or more whitespaces
         * is matched, then it gets replaced by a
         * single space.
         */
        printf("␣");
    }
%%

int main(void) {
    yylex();
    return 0;
}
```

2 Generating source code that implements the specified lexer and compiling `lex.yy.c`

The following sequence of commands are used:

1. Generate lexer source code:
   `lex replace-whitespace.l`

2. Compile the lexer. The executable file `./a.out` is created after running the command:
   `gcc lex.yy.c`

3. Running the lexer. The input file is `input.txt` which has a few lines with spaces and tabs. The lexer will replace one or more spaces and tabs with a single space and print the output. Redirection operators are used to input the file to the lexer and then create the output file `output.txt`
   `./a.out < input.txt > output.txt`.

3 The input file

```
Hello, this is a file with whitespaces.
Single whitespace,  double whitespace,
tabbed whitespace,
You can   find         everything here.

H a v e a n i c e  d a y!

If you run this  through the lex.yy.c program,
it will replace all whitespaces (spaces and tabs)
with a single space.
```

4 The output file

```
Hello, this is a file with whitespaces.
Single whitespace, double whitespace,
 tabbed whitespace,
You can find everything here.

H a v e a n i c e d a y!

If you run this through the lex.yy.c program,
it will replace all whitespaces (spaces and tabs)
with a single space.
```