

Midterm Report

- Name: 陳偉俊
- Student ID: 61347090S

Attached Files

- 61347090S_microex.l : contains the scanner's source code
- 61347090S_script_output.txt : output of `script run`
- `exr_lex_test_data.txt` : test data provided by professor (no lexical error)
- `exr_lex_test_data_bad.txt` : test data with lexical error (checkout line 8)
- `exr_lex_test_data_terminated_string_newline.txt` : test unterminated string ("hello) caused by newline without closing quote
- `exr_lex_test_data_terminated_string_eof.txt` : test unterminated string ("hello) caused by sudden eof

Features

1. Recognition of the 38 tokens shown in the PPT slide.
2. Supports case-insensitive keywords (begin, BEGIN, etc.)
3. Handle the keyword "print" as a WRITE token (as provided in test case).
4. Handle lexical error:
 - Shows line number
 - Shows column number (char)
 - Shows illegal token
5. Identify unterminated string due to newline before closing quote (Unterminated string literal on line x column y)
6. Identify unterminated string due to EOF before closing quote (Unterminated string literal on line 3, column 9 (reached EOF))

Explanation:

From the test data provided by professor, we can see that the scanner have to be **case_insensitive**. We can also see that the string "Good!" is written using `print` instead of `write`. Therefore, we can also assume that `print` is another way to write in micro/ex. I registered `print` as another WRITE_TOKEN in the scanner

We also improve the scanner's UX by showing line number, column number, and the illegal token on error. This is done by keeping track of the current line and current column as the scanner matches the tokens.

We also give the scanner the ability to detect unterminated string (no closing quote). We did this by tracking the starting line and column numbers (`string_start_line` and `string_start_col`) when encountering an opening quote. When entering string mode using `BEGIN(String_State)`, we enter an exclusive lexer state that handles string content differently. If a newline or EOF is encountered before finding a closing quote, we report the error with the exact position where the unterminated string began.