Task documentation SYN: Syntax Highlight in PHP for IPP 2015/2016
Name and surname: Matej Marušák
Login: xmarus06

1 Task

The goal was to create a script for automatic highlighting of parts of text. The script takes a table of regular expressions, to which output format is assigned.

2 Solution

Firstly, a simple error handler was created. After calling this function, a adequate error message is printed to stderr and the whole script is exited with a suitable exit code. In the following text, *ERROR* means that this function was called.
The main script runs following 5 steps in the given order.

1. **Parse command line options**
   For this task a simple command line parser was written. Firstly, the number of parameters is counted. If the number is not in range $\langle 1, 4 \rangle$, ERROR is called. Next, each argument is compared with possible arguments, that means, is checked if starts with one from this list ("--br", "--format=","--input=","--output=", "--help"). If so, this fact is marked down and possible file names were parsed and saved as well.
   This part also checks for duplicity arguments and prohibited combinations.

2. **Open and process format file**
   Format file obtained from previous step is opened and the content is read into a variable. If the file name was not specified, a empty file is considered. Next, two checks and edits must be performed.

   - **Check validity of regular expressions and transform them into PCRE format**
     Validity checking of regular expressions was done by a finit state machine. In total 9 states were needed plus a "trap" state. When checking, a PCRE regular expressions were generated alongside. States were created for checking of possible succeeding characters, for control of opened parenthesis a counter was used.
   - **Check validity of format entries and convert them into html tags**
     Every entry is confronted with list of accepted formats. If corresponding html tag exists, this entry is substituted by pair of beginning and ending tag. If no tag is found, or another error is detected (mainly wrong size of text or color format in a wrong format) ERROR is called.

3. **Open input file and read content**
   Input file obtained from step 1 is opened and the content is read into a variable. If the file name was not specified a stdin is considered. When the file cannot be opened or read, ERROR is called.

4. **Apply rules on input text**
   For each, now edited, regular expression with corresponding beginning and ending tag, all

occurrences in the input text are found. For each occurrence a HTML tags are added. Adding these tags can cause some issues and for this reason a merger was created. This merger takes an old version of text and a original text with one pair of tags added. A new version is created, by merging this strings together. Merging process compares two strings character by character. When no tag is actually in neither string, a current character is copied into result. When in one string tag is present, firstly this tag is copied. If a tags are in both strings, by simple rules tags are copied one after another. Result is then used as an old version. The last output from merger is a final product.

5. **Open output file and write content** In this step, firstly a --br options is checked. If selected, all newline characters $\backslash n$ are replaced
   with $<br \backslash >$.
   Output file name obtained from step 1 is opened and the result from previous step is written into this file. If the file name was not specified a stdout is considered. When the file cannot be opened or written into, ERROR is called.

3 Extensions

A **NQS** extension was implemented. A simple function before converting to PCRE in step 2 is called. This function replaces all sequences consisting only from characters "*" and "+" into one character "*". If only more occurrences of character "+" are in a row, this sequence is replaced by character "+".

4 Conclusion

Created script was tested by official test. Another tests were created for testing more complex regular expressions. All tests passed successfully. A script was tested on school server merlin and problem with older version of PHP was removed.