

1 Assignment

The task was to create a script in PHP 5.6, that highlights input text according to given regular expressions and formatting HTML tags from formatting file.

2 Design description

2.1 Argument parsing

Arguments of command line are processed with function `getopt`, which simplified argument processing. In addition it was necessary to manage situations as duplicate arguments or unsupported combinations of arguments. In each situations script exits with appropriate message for user.

2.2 Format file analysis

As PHP offers pretty good regular expression functions, I decided to use it for parsing format file. Every line of format file is checked, whether it does not contain unsupported characters and also if it meets formatting requirements. After that function `preg_split` splits each line to an array, which is appended to the array of format file.

2.3 Regular expressions analysis

Due to specific syntax of regular expressions in this assignment, it was necessary to translate them. I have chosen finite state machine to split regular expressions to tokens. Each token is analyzed and translated to PCRE regular expression and then appended to the array for further analysis. After complete translation, every member of the array is inspected, whether whole translated regular expression is valid. If there is unsupported combination in array (e.g. `".."`, `".+"`, `"("`, etc.), script exits and warn user.

2.4 Inserting HTML tags

At first it seemed, inserting tags could be done easily with regular expressions, but unfortunately it would not lead to proper solution. The problem was, closing tags must be put before opening tags, which in my opinion can not be achieved with regular expressions.

Description of my solution:

1. at the begining an `$tag_array` with size of input file is created
2. `preg_match` matches string with regex and save `$offset` from begining
3. function `generate_tags` is called to generate opening HTML tags
4. opening tags are inserted to `$tag_array` to index which is equal to `$offset` and concatenate with existing tags on same index
5. function `generate_tags` is called to generate closing HTML tags
6. closing tags are inserted to `$tag_array` to index which is equal to `$offset` and length of matched string, eventually concatenate with existing tags on same index
7. when all HTML tags are inserted to `$tag_array`, input string is concatenate with `$tag_array`

2.5 NQS extension

One of optional extensions consists of implementing correct evaluation of regular expressions quantifiers. As each of PCRE symbol is stored as string in array of PCRE strings, it was not hard to process multiple quantifiers.

Description of my solution:

- check if previous string in array of strings is "+" and if current string in array of strings is "+" or "*", delete previous string in array of strings
- check if previous string in array of strings is "*" and if current string in array of strings is "+" or "*", copy previous string to index of current string and delete previous string

2.6 Error handling

If an error occurred, the function `error_msg($errno, $str)` is called. First parameter, which is passed to function is exit code. Second parameter is string, which contains what type of error has occurred, in which part of processing and eventually which part of format file caused it.