Task documentation SYN: Syntax highlighting in Perl for IPP 2011/2012
Name and surname: Fridolín Pokorný
Login: xpokor32

1 Introduction

This document is a technical brief of Syntax highlighting Perl script `syn.pl`. User is allowed to define syntax which is highlighted using HTML tags.

2 Implementation

Problem is abstracted into four main functions.

Function `check_opt()` processes argument vector into hash which controls program flow. Parsed arguments are checked for errors (unknown arguments) and duplicity is not allowed.

Depending on passed arguments, function `open_files()` opens files, if neccessary. If no input or output files are given, `stdin` and `stdout` is used.

Parsing and highlighting is done in `syntax()` function, which uses its subroutines. First of all, format file is loaded and parsed line by line in `syntax_format()` function. Implementation specified regular expressions are translated into Perl regular expressions in `syntax_format2perlre()` rutine and tags are precomputed in correct order in `syntax_get_tags()` function.

Program has to load whole file from specified input file because of `%n` regular expression (new line). It is easier and transparent to match regular expression rules on whole input, but HTML tags are not directly inserted. Instead, it is better to store position on which HTML tags will be placed (with precomputed HTML tags itself). This computation is done in `syntax_indexes()` subroutine. Indexes[1] have to be stored in right order, thus an array is used. Note that ending HTML tags have to be concatenated in reverse order to match corresponding opening tags. Every new record is placed at the end of index array so it is preserved to insert HTML tags into the input text in correct order.

Last called subroutine in `syntax()` function is called `syntax_print()`. It simple browse input character by character and checks whether some HTML tags should not be printed before browsed character. On end of line symbol, `<br>` tag is printed if neccessary.

Last function `close_files()` closes opened files.

2.1 Regular expression translation

Main part of script consist of translation patterns described by regular expressions to Perl regular expressions. Every special character (such as `?`, `.` etc.), which has different meaning in Perl regular expressions than regular expressions described in format file, needs to be escaped or substituted. Patterns describing this substitution needs to obey different enquoting in format file than in Perl.

3 Source code metrics

Number of source files:	1
Number of functions implemented:	10
Line count (comments included):	317

[1] Implementation uses composite type to group indexes and computed HTML tags together.