

Project moved to slre.googlecode.com

SLRE - Super Light Regular Expression library

SLRE is an ANSI C library that implements a tiny subset of Perl regular expressions. It is primarily targeted for developers who want to parse configuration files, where speed is unimportant. It is in single .c file, easily modifiable for custom needs. For example, if one wants to introduce a new metacharacter, 'i', that means 'IP address', it is easy to do so.

Features

- Crossplatform - pure ANSI C
- Very simple API
- Light: about 5kB of code when compiled
- Uses no dynamic memory allocation
- Thread safe

Supported RE Syntax

^	Match beginning of a buffer
\$	Match end of a buffer
()	Grouping and substring capturing
[...]	Match any character from set
[^...]	Match any character but ones from set
\s	Match whitespace
\S	Match non-whitespace
\d	Match decimal digit
\r	Match carriage return
\n	Match newline
+	Match one or more times (greedy)
+	Match one or more times (non-greedy)
*	Match zero or more times (greedy)
*	Match zero or more times (non-greedy)
?	Match zero or once
\xDD	Match byte with hex value 0xDD
\meta	Match one of the meta character: ^\$().[*+?\

API

Two functions represent the API: one is used for compilations of the RE, the other for performing a match. Both functions return 0 on error, and 1 on success. If the round brackets are used in the RE, then matched substrings can be returned back to the caller in the 'struct cap **' array. Array size must be enough to hold all matches: array_size = number_of_round_bracket_pairs + 1. The first element of the array will always hold the substring matched by the whole RE.

```
/* Captured substring */
struct cap {
    const char    *ptr;          /* Pointer to the substring    */
    int          len;           /* Substring length          */
};

int slre_compile(struct slre *, const char *re);
int slre_match(const struct slre *, const char *buf, int buf_len,
               struct cap *captured_substrings);
```

Usage example

This example shows how to parse HTTP request line:

```
struct slre    slre;
struct cap     captures[4 + 1];

if (!slre_compile(&slre, "^(GET|POST) (\S+) HTTP/(\S+)\r\n") {
    printf("Error compiling RE: %s\n", slre.err_str);
} else if (!slre_match(&slre, buf, len, captures)) {
    printf("Not a valid HTTP request\n");
} else {
    printf("Request line length: %d\n", captures[0].len);
    printf("Method: %.*s\n", captures[1].len, captures[1].ptr);
    printf("URI: %.*s\n", captures[2].len, captures[2].ptr);
}
```

Download

- Version 1.0: [slre.h](#), [slre.c](#)

License

```
/*
 * -----
 * "THE BEER-WARE LICENSE" (Revision 42):
 * Sergey Lyubka wrote this file. As long as you retain this notice you
 * can do whatever you want with this stuff. If we meet some day, and you think
 * this stuff is worth it, you can buy me a beer in return.
 * -----
 */
```

Author

Copyright © by Sergey Lyubka (valenok at gmail dot com). Feel free to contact me if you want to comply with the license.

