StackExchange ▾ sign up log in tour help ▾ careers 2.0

stackoverflow

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required. Take the 2-minute tour ×

# c regex code not working?

I have this code below which checks whether the user has entered a syntactically correct url. Regex code was got from Regular expressions in C: examples?

```
printf("Enter the website URL:\n");
fgets(str, 100, stdin);
if (!strcmp(str, "\n")) {
    printf("Empty URL ");
    exit(2);
}

regex_t regex;
int reti;
char msgbuf[100];

/* Compile regular expression */
reti = regcomp(&regex, "[a-zA-Z0-9\\-\\.]+\\.[a-zA-Z]{2,3}(/\\S*)?$", 0);
if (reti) {
    fprintf(stderr, "Could not compile regex\n");
    exit(3);
}

/* Execute regular expression */
reti = regexec(&regex, str, 0, NULL, 0);
if (!reti) {
    puts("Match");
} else if (reti == REG_NOMATCH) {      //This else if always executes.
    puts("No match");
    exit(4);
} else {
    regerror(reti, &regex, msgbuf, sizeof (msgbuf));
    fprintf(stderr, "Regex match failed: %s\n", msgbuf);
    exit(5);
}

/* Free compiled regular expression if you want to use the regex_t again */
regfree(&regex);
```

However the regex always fails, even if the url entered is correct. I know the regex is correct but for some reason it fails on the 'Execute regular expression' part. Even if the user enters a syntactically correct URL the else if always executes.

What could be the reason for the else if always executing?

c regex url

asked Mar 13 '12 at 9:14

user667430
251 ●5 ●29

What is the output of your program? In particular, what's in `msgbuf` after the `regerror()` call? – Ferdinand Beyer Mar 13 '12 at 9:22

There is no error it always executes the else if (reti == REG_NOMATCH) – user667430 Mar 13 '12 at 9:28

add comment

## 1 Answer

Your pattern is *not* valid!

Note that POSIX defines two flavors of Regex: Basic (BRE) and extended (ERE) (see Wikipedia). Since you want to use the "extended" flavor, pass the `REG_EXTENDED` flag to `regcomp()`.

Here are (some of?) the problems with your pattern:

`[a-zA-Z0-9\\-\\.]+\\.[a-zA-Z]{2,3}(/\\S*)`

- Within brackets ( `[]` ), you don't need to escape special characters. In fact, you cannot escape them and `[a-zA-Z0-9\-\.]` will match backslashes, but not the hyphen, since `\-\` is interpreted as the range from `\` to `\`. If you want to match the hyphen, place it first or last in the character list: `[a-zA-Z0-9.-]`
- The Perl-style character class `\S` is not supported by POSIX. Use `[^[:space:]]` instead.
- Quantifiers `{}` need to be written as `\{\}` with BRE
- The `+` and `?` quantifiers are only supported by ERE

To summarize, replace the call to `regcomp()` with this one:

```
reti = regcomp(&regex, "[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,3}(/[^[:space:]]*)?$", REG_EXT
```

edited Mar 13 '12 at 10:06
Tim Pietzcker
138k ●11 ●136 ●222

answered Mar 13 '12 at 9:28
Ferdinand Beyer
26.2k ●4 ●54 ●83

Thanks, I have tried replacing my regcomp with yours but i am still getting the same result. – user667430 Mar 13 '12 at 9:43

1 Maybe it's because your input ends with `\n` and your regex pattern does not allow trailing whitespace? – Ferdinand Beyer Mar 13 '12 at 10:44

In short, using libpcre instead of regcomp is recommended to the OP :) – jørgensen Mar 13 '12 at 10:59

add comment

**Not the answer you're looking for? Browse other questions tagged c regex url or ask your own question.**