

Re: can I know how to write a html parser in C

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2005-02/3738.html

From: Walter Roberson (*roberson_at_ibd.nrc-cnrc.gc.ca*)

Date: 02/28/05

Date: 28 Feb 2005 03:50:22 GMT

In article <1109560168.939988.157590@z14g2000cwz.googlegroups.com>, WUV999U <usbharath.ganesh@gmail.com> wrote:
: op = fopen(argv[1], "r");

argv[1] might be NULL. You should be checking that you have the right number of parameters before you use any of them.

```
:void htmlparse(FILE * op)
: {
: char line[81];
```

Are the lines truly limited to 80 characters of text? It is not at all uncommon to encounter HTML in which the lines go on for several hundred characters.

```
: char images[250];
```

That declares a single character array named 'images' with a maximum null-terminated character string size of 249 characters. However, since you are only fetching 80 characters per line, the maximum image file name you are going to be able to extract is about 68 characters (once you remove the tag and quotes.)

If you want to allow for 250 images, then you should be declaring either an array of char * pointers or else a "two dimensional" array of characters.

```
: if (fgets(line,81,op) == NULL)
```

There's that magic number again, 81. Any time you have a number whose meaning is not obvious and which is repeated, you should either use a #define or store the value in a variable [which would have implications on how you would write the code.]

```
: {
: printf("Error reading data");
: exit(0);
: }
```

Eventually you are going to run out of input and get NULL returned. That isn't an error: it is a signal that your function should finish up and return. As you have named the function 'htmlparse', the reader would tend to assume that –all– the function does is parse the input and extract certain information from it, but would not act upon that information, so the reader would tend to assume that you would return the list of images to the calling routine and let it do whatever should be done with the list.

```
: puts(line);
```

Why do you need to output the line at that point? The input file isn't going anywhere, so you are unlikely to need to duplicate the input.

```
: if(line == "<img src")
```

That is never going to be true. That is going to compare the *address* of the string "<img src" to the address of the character array 'line'. Since "<img src" is a literal string, it is not going to have the same address as your buffer.

You also cannot fix this just by using strcmp() instead of testing the pointer: you need to be looking inside the line to find a place on the line (not necessarily at the beginning) where the string "<img src" occurs. Try strstr(). But watch out for comments and for the possibility that you might be within a quoted string...

Note too that in the general case it is perfectly acceptable in HTML for there to be a linebreak between the "<img" and "src". Are you working with a very restricted subset of HTML? If so then it would help a lot to describe what the subset is. Some HTML subsets are very easy to parse, whereas HTML in general is fairly complex to parse.

```
:well,, thats all i hav..... and m stuck here...
```

Ekkk!

No offense intended but you really haven't gotten very far at all and have made a number of mistakes in what you posted. Looking at this, we would tend to conclude that you are very much a beginner at C (and possibly a beginner at programming in general). Parsing general HTML is something that requires a fair bit of experience to program correctly; if what you posted is indeed representative of your C skills then you have no hope of writing a generalized HTML img file name extractor in any reasonable amount of time. Even a well–experienced programmer would take more than "a day or two" to write a proper HTML parser from scratch.

comp.lang.c: Re: can I know how to write a html parser in C

[Of course, a well-experience programmer would know to *not* write it from scratch if it could be avoided: there are a number of already-written HTML parser libraries out there, and there are programs such as "lynx" which could be canablized. Writing from scratch would usually be reserved for instances in which there were notable copyright or patent issues at stake.]

--

IEA408I: GETMAIN cannot provide buffer for WATLIB.