

Re: [newbie] _lfind syntax problem

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.language/2004-12/1191.html>

From: Victor Bazarov (v.Abazarov_at_comAcast.net)

Date: 12/23/04

Date: Thu, 23 Dec 2004 11:57:45 -0500

Peter Schmitz wrote:

> *this time, I want to do some binary searching.*

Do you know what 'binary searching' means?

> *Therefore I have declared the*

> *following:*

>

> *BYTE *buffer; //holds pointer to data to search in*

> *UINT bufferlen; //length in bytes of buffer*

> *BYTE *pattern; //holds pointer to pattern to search for*

> *UINT patternlen; //length of pattern in bytes*

>

> *Now, I've decided to use _lfind for this searching (I'm not using bsearch,*

'_lfind' does NOT perform binary search. It performs a _linear_ search.

> *because AFAIK I'd have to sort the buffer I search in – and as I normally*

> *search in rather small buffers this would decrease my speed significantly –*

> *or am I wrong?).*

No, it's a valid reason.

> *But somehow I don't get the correct syntax to call _lfind (perhaps because*

> *I'm a newbie to C). So, could someone show me how to call _lfind with the*

> *parameters from above?*

```
#include <iostream>
using namespace std;
#include <search.h>
```

```
struct patternmatching {
    char const *pattern;
    int len;
};
```

```
int patternmatch(void const *pat, void const *str) {
    patternmatching const *p =
```

```
    reinterpret_cast<patternmatching const*>(pat);
char const *s =
    reinterpret_cast<char const*>(str);

    return strcmp(s, p->pattern, p->len);
}

int main()
{
    const char *pattern = "abc";
    int patternlen = 3;
    const char *buffer = "123abc123";
    int bufferlen = 9;

    patternmatching pat = { pattern, patternlen };
    unsigned searchlen = bufferlen - patternlen;
    const char * matched =
        (const char*) _lfind(&pat, buffer,
            &searchlen, 1, patternmatch);
    if (matched) {
        cout << matched << endl;
    }
    else {
        cout << "didn't find\n";
    }
}
```

This is a very convoluted way, of course. Use 'std::find_if', it should be easier, actually.

Victor