

Re: Malloc code

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.language/2008-01/msg00030.html>

- *From:* Robby <Robby@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 2 Jan 2008 10:23:03 -0800
-

Hello David Wil.

David, excuse my ignorance, but what do you mean by top-post? :-)

If you look it up in Google, Google Groups or Wikipedia you will find all you need to know. These Microsoft newsgroups are very tolerant of different posting styles (a bad thing, IMHO), but no matter which style you use you should take care to trim previous posts.

This is bottom posting right? I find it makes more sense than top posting.
IMO!
I could be wrong!

2. I did notice that there were several hard coded constants in your code (11, 12, 15, 16, 176) and wondered what their purpose was.

David: can you let me know where you see this? And its in the clearer code sample that you see this, right? I know that I assign bogus numbers in the TCP_LOAD_MCB_X() function. Is this what you mean?

In your code I see the following lines:

```
int xxx[12];
pMCB1 = malloc(176*(sizeof(struct MCB)));
for(i=0;i<11;i++)
```

Re: Malloc code

```
int xxx[16];
pCL_B = malloc(15*(sizeof(struct CL_B)));
```

There is no indication what these hard-coded numbers signify, and at least some of them are surely typos.

Yeah, the 176 was the atrocious biggy! It should of been 11! The signification of this number is that every character (ex: A-Z) is stored in an external memory. The address of where a particular character starts is denoted by the A0,A1,A2 items of an MCB structure. Also every character has extra properties such as back ground color, and text color and so forth. For now, there is 12 properties in all per character. Therefore, the 11 sets the properties for 11 subsequent characters that I wish to display. This will be usefull because, I will then be able to display, a word with each of its characters as different properties... such as different colors for every letter of the word.

And breifly, the 15 sets aside some memory to store 30 bytes for the width of 1 row of pixles of a character. Since this particular font is 15 pixels wide and the pixels in the LCD I am using requires 2 bytes per pixel, then I need 30 bytes to display a row. It does get quite complicated since I am building all the characters, icons and images from scratch.

Anyhow in the full blown program, the 11 and the 15 are not hard coded. Instead, they will come from another structure which will contain general informations about the current font being used.

As for not using the void pointer, I will have to do some further testing and programming to see if what I am doing will fit well with the rest of the programming code of the app. Thanks very much for your recomendation example. I just needed some insight on passing arrays of pointers.

```
void main()
{
int *gptr;
struct MCB *r1;
struct MCB *r2;
struct MCB *r3;

gptr = (int *) FX();

r1 = *gptr[0];
r2 = *gptr[1];
r3 = *gptr[2];

r1[0].LEADING_PIX;
r2[2].LEADING_PIX;
```

Re: Malloc code

```
r3[7].LEADING_PIX;  
  
free(*gptr[0]);  
free(*gptr[1]);  
free(*gptr[2]);  
}
```

It's int main(), not void main().

I tried it as you say:

```
int main()  
{  
  
.... other code  
  
return 0;  
}
```

and it works as well. Either way works. Is there a reason as to why

```
void main()  
{  
.... other code  
}
```

should not be used? I mean I get the same succesfull compilations???

Wish you well David!

--

Best regards
Robert

"David Wilkinson" wrote:

Robby wrote:

Replies inline:

Hello David Wilk.

David, excuse my ignorance, but what do you mean by top-post? :-)

Re: Malloc code

Re: Malloc code

If you look it up in Google, Google Groups or Wikipedia you will find all you need to know. These Microsoft newsgroups are very tolerant of different posting styles (a bad thing, IMHO), but no matter which style you use you should take care to trim previous posts.

2. I did notice that there were several hard coded constants in your code (11, 12, 15, 16, 176) and wondered what their purpose was.

David: can you let me know where you see this? And its in the clearer code sample that you see this, right? I know that I assign bogus numbers in the TCP_LOAD_MCB_X() function. Is this what you mean?

In your code I see the following lines:

```
int xxx[12];
pMCB1 = malloc(176*(sizeof(struct MCB)));
for(i=0;i<11;i++)
int xxx[16];
pCL_B = malloc(15*(sizeof(struct CL_B)));
```

There is no indication what these hard-coded numbers signify, and at least some of them are surely typos.

```
You could have gotten rid of the particular problem you had
by doing
#define NMAX 11 /* or 12 ? */
pMCB1 = malloc(NMAX*(sizeof(struct MCB)));
for (i=0; i<NMAX; i++)
```

Yes indeed I could of done this, but actually do it in the full blown project code!

Then why not do it in the example code? You should make your example code as clear as you can possibly make it. Your posted code had so many things wrong that it was very difficult to see what the real problem was.

```
Although I did guess that xxx was being used only for
debugging
purposes, it would have been nice if this had been indicated
in the code:
int xxx[12]; /* for debugging only */
```

Re: Malloc code

You bet I will indicate this next time! I think I confused alot of people with that !

I didn't know that we can insert a file in our posts! I will need to figure out how to do this.

I was not suggesting that you attach a file (this is generally frowned on in usenet). Rather you should combine all your code into a single file, check that it still compiles and illustrates your problem, and then paste the whole file into your post. As it was, if a reader wanted to test your code he/she had to pick different parts of your post out into separate files, and add them all to a vc project. Too much trouble.

6. I understand that you are using some embedded compiler, but this is a vc group. It would be god if you were to try your code in vc, and see if the problem is still there. If not, it is unlikely that anybody here can help you with it; but if so, this may be a big clue.

I thought of doing that, but you see, its 2 years I have not done any vc++ because, I really had to get the embeded stuff of my project done first (man, I didn't think it would take me so long). So, when I posted this post a couple of days ago, I did try it in vc++ by reproducing the three files and copying the code, but it gave me over 150 errors. And then with all of the discouragment I was going through with this post, I just gave up and closed vc++. But, I really can't wait to pick it up again in the near future... theres so much I want to learn.

If you combined all your code into a single file, then you would just create an empty console project and add this file to it. Remember, someone who is really going to test your code is going to have to do this, so shouldn't you do it also?

Thanks for the clue... But this site is very good and there is alot of resourcefull help.

You guys also helped me alot with vc++. I apologize if this post was so unclear but this was a nasty one.

Also, since David Web. has reccomended to not declare pointers in my header, this sort of complicated things where I must now assign all the pointers returned from malloc to an array of pointers, and then return this array of pointers back to main, which I have never done before. I did start coding it.

Re: Malloc code

I know how to pass in an array of pointers. But don't know how to return an array of pointers. So I just returned a void pointer and assign the value to another pointer declared in main. I read up on some samples which showed some examples but didn't show how to do this exactly the way I needed it. This is what I came up with and it compiles but I did not test it yet. This will be done tomorrow. I was also curious, is there another way where I can pass back the actual array of pointers from the function instead of a void pointer? Anyways here is what I have done.

```
void *FX()
{
    struct MCB *MCBptrs[3]; //Array of pointers
    MCBptrs[0] = malloc(10*(sizeof(struct MCB)));
    MCBptrs[1] = malloc(20*(sizeof(struct MCB)));
    MCBptrs[2] = malloc(30*(sizeof(struct MCB)));
    return MCBptrs;
}
```

The reason that I suggested to you (some time back) to return void* was that you were returning pointers to three different structs from the same function (a bad design, IMHO). You are not doing that any more. If you want to assign a whole array of pointers, you can do

```
void FX(struct MCB* MCBptrs[3], int flag)
{
    if(flag == 0)
    {
        MCBptrs[0] = malloc(10*(sizeof(struct MCB)));
        MCBptrs[1] = malloc(20*(sizeof(struct MCB)));
        MCBptrs[2] = malloc(30*(sizeof(struct MCB)));
    }
    else
    {
        free (MCBptrs[0]);
        free (MCBptrs[1]);
        free (MCBptrs[2]);
    }
}
```

```
void main()
{
    int *gptra;
    struct MCB *r1;
    struct MCB *r2;
    struct MCB *r3;
```

Re: Malloc code

```
gptr = (int *) FX();

r1 = *gptr[0];
r2 = *gptr[1];
r3 = *gptr[2];

r1[0].LEADING_PIX;
r2[2].LEADING_PIX;
r3[7].LEADING_PIX;

free(*gptr[0]);
free(*gptr[1]);
free(*gptr[2]);
}
```

It's int main(), not void main().

```
int main()
{
struct MCB* MCBptrs[3];
FX(MCBptrs, 0); /* allocate */
/* do work here */
FX(MCBptrs, 1); /* free */
return 0;
}
```

I never really coded in C, but in C++ I never use global variables. It is (almost) always possible to pass required information in function arguments.

Thankyou for your post, hope you had a nice new year's eve!

A good year to you also, Robbie.

--

David Wilkinson
Visual C++ MVP