

Re: Crash by allocationg small blocks

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.language/2005-01/0709.html>

From: Doug Harrison [MVP] (dsh_at_mvps.org)

Date: 01/15/05

Date: Sat, 15 Jan 2005 07:00:14 -0600

"Sebastien LEGUET" <Sebastien.LEGUET@discussions.microsoft.com> wrote:

```
>Hello,
>
>I try to exeute this code and I have a crash during the execution. The
>realloc function return NULL.
>#include "stdio.h"
>#include "stdlib.h"
>
>int main()
>{
> char** pData = NULL;
> char *pszFicDescName = NULL;
>
> for( int i=0; i< 7000000; i++ )
> {
> pszFicDescName = (char*)malloc(1041);
> strcpy(pszFicDescName,"Ceci est un test");
> if (pData == NULL)
> pData = (char**)malloc( sizeof(char* ) );
> pData = (char**)realloc( (void*)pData, (i+1)*sizeof(char* ) );
> //access violation in small-block allocator
> pData[i] = pszFicDescName;
> printf("Cpt = %d\n",i);
> }
>
> return(0);
> } //end main
>
>Could you tell me if you know about that problem and how can I resolved that
>because I need to use standard C code.
>I tried this code on VC6 and VC7 and I have the same result. I know that
>there is a resolution by declaring _set_sbh_threshold(0); but it doesn't work
>and I have the SP5 of VC6 so the realloc function has been corrected.
>
>In fact if I comment the line
> pData[i] = pszFicDescName;
```

>the program is correctly executed to the end.

I don't see anything illegal, so if it crashes, I'd call it a bug in the library. I tried it in VC.NET 2003 and it failed without crashing after some 80,000 iterations. I do think your allocation pattern is somewhat pathological, looping tightly around the following, essentially forever:

```
pszFicDescName = (char*)malloc(1041);  
pData = (char**)realloc( (void*)pData, (i+1)*sizeof(char*) );
```

Unless you get lucky, you can assume realloc is allocating and copying to a new buffer each time you call it, possibly fragmenting your heap in the process. In addition, when the buffer cannot be extended in-place, growing the buffer linearly is $O(N^2)$ in the size of the buffer, which can be glacial even on a fast machine when the buffer size grows large. I would recommend growing the buffer exponentially, say by a factor of two when you need to reallocate. Making that simple change caused the program to run *_much_* faster while also allowing the loop to run 425,000 times, at which point, I killed the program, as it began slowing down, and I heard my pagefile growing big-time.

Unfortunately, I still don't fully understand why your original approach failed. As Walter mentioned, it appears to fail in the Windows function HeapReAlloc, so perhaps if you have the time, you could explore whatever HeapXXX analysis functions there are.

P.S. To #include a standard header, use angle brackets, not quotes. To understand VC's rules regarding the two forms, see "The #include directive" in the help.

```
--  
Doug Harrison  
Microsoft MVP - Visual C++
```