

## Re: malloc + 4??

**Source:** [http://coding.derkeiler.com/Archive/C\\_CPP/comp.lang.c/2004-04/0375.html](http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2004-04/0375.html)

---

**From:** Chris Torek (*nospam\_at\_torek.net*)

**Date:** 04/04/04

Date: 4 Apr 2004 16:41:14 GMT

In article <4070182a\$0\$27645\$61ce578d@news.syd.swiftdsl.com.au>  
Kevin Torr <kevintorr@hotmail.com> writes:  
><http://www.yep-mm.com/res/soCrypt.c>

In general, it is better to post the actual problematic code (preferably after shrinking it down to a "problematic nub", as it were), but a URL reference can work if the one reading netnews bothers to follow the link. :-)

>*I have 2 malloc's in my program, and when I write the contents of them to the screen or to a file, there aren't addition 4 characters.*

>*As far as I can tell, both the code to register the malloc and to write information into the malloc is solid. Why then is my program returning an additional 4 characters?*

It is not quite as solid as one might hope, although the problem has nothing to do with malloc() per se. Here are excerpts from the code (quoted with ">" as usual, although I had to insert the markers myself):

```
>// soCrypt 1.0  
>  
>#include <stdio.h>  
>#include <stdlib.h>  
>#include <string.h>  
>#include <time.h>  
>// #include <md5.h>
```

OK so far, although `//`-comments are specific to C99. You have included necessary headers, so you will not need to cast malloc()'s return value.

```
>// Global variables  
>  
>int statCode = 0; // Status code  
>int mode; // Mode variable (1 = enc, 2 = dec)  
>int i; // Looper variable
```

Re: malloc + 4??

```
>int inSize = 0; // Input filesize  
>int intRand; // Random int  
>char tmp_char; // Temporary char
```

Many of these should not be file-scope external-linkage ("global") variables, although this is mostly a style issue (at least in a program this small).

Note that tmp\_char has type "char"; on a typical PowerPC, it would hold values between 0 and 255 inclusive, because there plain "char" is unsigned. The variable inSize is a plain (signed) int and has at least the range [-32767..+32767] (although most systems, today, have an even wider range, about +/- 2 billion).