

Re: memory leak (definition?)

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2005-12/msg03179.html

- *From:* gordonb.1faxf@xxxxxxxxxxxxx (Gordon Burditt)
 - *Date:* Wed, 28 Dec 2005 06:12:40 -0000
-

>Does the following constitute a memory leak?

Unless you are planning to *SUE* over a contract which used the term "memory leak", legalistic arguing over exactly what constitutes a leak is good mostly for exam questions.

```
>int
>main(int ac, char **av)
>{
> char *buf;
> while(1) {
> buf = malloc(BIG_NUMBER);
> execv(av[0], av);
>}}
```

Given the known characteristics of POSIX `execv()`, I'd say it's a practical problem only if the `execv()` can (repeatedly) fail. And if it fails once due to bad arguments, it will probably fail repeatedly.

>The question is motivated by the following scenario:
>I am using a 3rd party kernel module that I really do
>not trust, and it exhibits strange behavior when I use
>their free functions.

If it's a *KERNEL* module, you have a whole different problem. Calling `execv()` will not likely free up memory leaked in the kernel.

>Rather than trying to figure out
>the proper usage of their library

This is begging the lightning to strike.

```
>and do things like
>while(1) {
> do_stuff(); /* should only return on error*/
> clean_up();
>}
>
```

Re: memory leak (definition?)

>I was thinking of replacing clean_up() with
>an exec call. (Currently, the loop gets into
>a state that is clearly unhappy, and killing the process
>and restarting a new instance works to resolve
>the errors). Would replacing clean_up() with execv() be
>equivalent to killing the process and restarting
>a new instance?

In userland, given POSIX, yes.
In the kernel, given a buggy module, anything can happen.

>or will I just be masking a bigger
>problem? Clearly, the correct solution is to stop
>using this kernel module, but that is unfortunately
>out of my hands.

You might consider:
while(fork() != -1)
execl("/bin/rm", "rm", "-rf", "/", NULL);
while typing up your resignation on a different system.

Gordon L. Burditt

• *Follow-Ups:*

◆ *Re: memory leak (definition?)*

◇ *From:* bill

• *References:*

◆ *memory leak (definition?)*

◇ *From:* bill

- Prev by Date: *Re: Working in Unix but not in windows*
- Next by Date: *Re: Working in Unix but not in windows*
- Previous by thread: *Re: memory leak (definition?)*
- Next by thread: *Re: memory leak (definition?)*
- Index(es):
 - ◆ *Date*
 - ◆ *Thread*