Re: xmalloc

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2007-06/msg03369.html

- *From*: Eric Sosman <esosman@xxxxxxxxxxxxxxxxxx>
- *Date*: Sun, 24 Jun 2007 09:12:06 -0400

Roland Pibinger wrote:

> On Sat, 23 Jun 2007 16:30:12 -0400, Eric Sosman wrote:
>
> > Not all that strange. Half the programmers in the world
> > are below-average.
>
> which is at least half-true.
>
> > But if you follow Malcolm's Maxims, even the above-average
> > programmers will be helpless. Programs that deal with malloc()
> > failure in a significant way may be a minority, but those that
> > do it *need* to to do it. Malcolm argues, in essence, that
> > such programs should not be written in C.
>
> The discussion basically boils down to one question: Is OOM an error
> that reasonably can and should be handled by the application or is it
> a fatal error? The 'fatal error' advocates have already shown how they tackle the
> problem. Now it's time for the other camp to demonstrate how OOM can
> be consistently _handled_ throughout the program ('return NULL;' is
> not enough).

Already mentioned a few times in this thread:

```
buff = malloc(image_size);
if (buff == NULL) {
fprintf (stderr, "Image too large (%lu) to paste\n",
(unsigned long)image_size);
return;
}
/* read image into buff, insert in current document */
```

Here's another I think has been referred to:

```
must_have_mem = malloc(how_much);
```

```
if (must_have_mem == NULL) {
fprintf (stderr, "Out of memory; shutting down\n");
save_snapshot(snapshot_file);
exit (EXIT_FAILURE);
}
/* store "must have" data in allocated memory */
```

And here's still another (I don't remember whether it's
cropped up in this thread yet):

```
void *getmem(size_t bytes) {
void *new = malloc(bytes);
if (new == NULL && bytes > 0) {
fprintf (stderr, "Failed to allocate %lu bytes\n",
(unsigned long)bytes);
free (emergency_stash);
emergency_stash = NULL;
new = malloc(bytes);
if (new == NULL) {
fprintf (stderr, "You were warned ...!\n");
exit (EXIT_FAILURE);
}
fprintf (stderr, "Running on fumes: save your work "
"and exit soon!\n");
}
return NULL;
}
```

Even if out−of−memory is a "fatal error," it does not follow
that the program should have no opportunity to "die with dignity."
Have you made a will, Roland? If so, should the fact that many
people die intestate invalidate your will? If not, I certainly
don't want your intestacy to invalidate my will!

−−
Eric Sosman
esosman@xxxxxxxxxxxxxxxxxx
.