

Re: Data Structure

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2005-06/msg03043.html

- *From:* Roger Leigh <\${rleigh}@invalid.whinlatter.ukfsn.org.invalid>
 - *Date:* Sat, 25 Jun 2005 18:58:02 +0100
-

-----BEGIN PGP SIGNED MESSAGE-----

Hash: SHA1

"Malcolm" <regniztar@xxxxxxxxxxxxxxxx> writes:

> "Roger Leigh" <\${rleigh}@invalid.whinlatter.ukfsn.org.invalid> wrote

>>> typedef struct

>>> {

>>> char *Data;

>>> char *Parent;

>>> int Size;

>>> int_err (*Open) (struct_ObjectRecord *Self,

>>> const char *Filename);

>>>

>>> int_err (*close)(struct_ObjectRecord *Self);

>>>

>>> int_err (*destroy)(struct_ObjectRecord *Self);

>>>

>>> int_err (*YourFile)(struct_ObjectRecord *self,

>>> const char *filename,

>>> int *outFile);

>>>

>>> } struct_ObjectRecord;

>>

>> It's an attempt to fake objects with virtual functions in C. However,

>> it's grossly inefficient. The virtual functions should have been

>> placed in a separate "class" or "vtable" structure, because there's

>> only one instance required for each class. You have a pointer to the

>> vtable as part of each object struct.

>>

> It's not grossly inefficient. It avoid a layer of indirection for the cost

> of three pointers per structure. Unless the number of struct_ObjectRecords

> is very large this is not unreasonable.

It's still (IMO, of course) suboptimal. You waste the space taken up by three pointers in every instance. Depending on the number of objects you need to instantiate, this could be a very large amount of waste. In this case it's 3/7 of the total size (assuming int has the same size as a pointer).

Re: Data Structure

More importantly, it prevents new member variables being added, so you can't derive new instance types. If the instance and class structures are separate, you can extend either at will.

Regards,
Roger

--

Roger Leigh

Printing on GNU/Linux? <http://gimp-print.sourceforge.net/>

Debian GNU/Linux <http://www.debian.org/>

GPG Public Key: 0x25BFB848. Please sign and encrypt your mail.

-----BEGIN PGP SIGNATURE-----

Version: GnuPG v1.4.1 (GNU/Linux)

Comment: Processed by Mailcrypt 3.5.8 <<http://mailcrypt.sourceforge.net/>>

iD8DBQFCvZsoVcFcaSW/uEgRAtrcAJ9w37PX67B9H16Vt7SvzEFzsakB0gCg3W/X
KXfLLe8oDkl/mlw6iXFuZoA=
=n7Eb

-----END PGP SIGNATURE-----

.

• **References:**

- ◆ **Data Structure**
◇ From: ranjeet . gupta
- ◆ **Re: Data Structure**
◇ From: Roger Leigh
- ◆ **Re: Data Structure**
◇ From: Malcolm

- Prev by Date: **Re: Implementing my own memcpy**
- Next by Date: **Re: Implementing my own memcpy**
- Previous by thread: **Re: Data Structure**
- Next by thread: **Re: Data Structure**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**