

Bitwise Flags abandoned? [closed]

Asked 16 years, 2 months ago Modified 13 years, 6 months ago


Viewed 3k times  Part of [PHP](#) Collective



19



Closed. This question is [opinion-based](#). It is not currently accepting answers.

 **Want to improve this question?** Update the question so it can be answered with facts and citations by [editing this post](#).

Closed 9 years ago.

[Improve this question](#)

I've noticed that plenty of opensource project doesn't use BITWISE flags anymore, even if it's fully supported by programming enviroment common for web (php/Mysql). It's that a "lost practise" for some effective problem, or is just that a lot of php programmers don't know how to handle this type of implementation?

Nothing too important, just very curious :) thanks to you all

PHP

php

mysql

open-source

Share

Improve this question

Follow

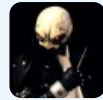
edited Feb 28, 2010 at 20:37



Tim Lytle

17.6k ● 10 ● 63 ● 92

asked Oct 22, 2008 at 8:04



Joshi Spawnbrood

515 ● 6 ● 13

9 Answers

Sorted by:

Highest score (default)



13



I'll stick my neck out and say that every technical position requires a sound understanding of bitwise operations.

And I have an anecdote that indirectly addresses the topic.

January 2007 I was in Cochin, India, recruiting for permanent development staff. Since I wasn't involved in the preliminary screening of candidates I had no idea what standard to expect so I had prepared a range of questions and topics ranging from simple understanding of binary and hexadecimal through to architecture, design, and project management.

When I discussed my approach with the Indian HR guy I was (gently) chided for pitching too low. He made it clear that my questions about hex would possibly be construed as an insult to the candidates experience or education.

But my experience of interviewing hundreds of candidates in the UK had fixed in me a conviction that it wasn't possible to pitch too low. My opinion was and still

is that if it becomes obvious a candidate is well qualified then it's simple and easy to adjust the level of discussion. I've never had anyone express feelings of being insulted, on the contrary I think a well qualified candidate might feel relieved at a flying start to the interview. It also helps to break the ice and build a rapport needed for a meaningful interview. On the other hand, unqualified candidates usually fall at these lower hurdles.

But not wanting to completely ignore local advice I cautiously decided to include my basic interview topics, and was quite prepared to abandon them if they didn't work.

As the interviews progressed I was glad that I started at that level. It didn't offend anyone, and unsuitable candidates were easily identified.

This is not to say that I expect candidates to deal with bit-twiddling day to day, but whatever the language a sound understanding of the fundamentals of programming is essential. Even developers at the higher levels of abstraction are exposed to hex on a regular basis (RGB values, for example). Parroting [stuff you find on the net](#) will only help to the extent that things work perfectly first time.

But for developers starting out in the past five years I believe it's all too easy to gloss over the fundamentals, cosseted by well intentioned IDEs and the meme of "codeless" programming. The Visual Studio installation

spash screens boast about developing without writing code. Indeed, [does Visual Studio rot the mind?](#)

Share Improve this answer

edited Oct 22, 2008 at 13:41

Follow

answered Oct 22, 2008 at 8:47



[Ed Guinness](#)

35.1k ● 16 ● 113 ● 149

4 An interesting anecdote, but I fail to see how this answers the question. – [Cory House](#) Jun 24, 2011 at 20:45



12



A lot of programmers these days seem to just have their heads filled with just enough knowledge to brute-force code out and then sent into the workforce without being taught what words like "bitwise" even mean.

It's a dying art I tell you...



Share Improve this answer

answered Oct 22, 2008 at 8:09

Follow



[Ignacio Vazquez-Abrams](#)

797k ● 160 ● 1.4k ● 1.4k

2 I see no coherent justification. Yes, this is a decreasingly relevant/prevalent skill, but this doesn't answer the question. – [Cory House](#) Jun 24, 2011 at 21:04



9

Bitwise operations aren't just about conserving memory. They're really useful for writing clear code. Which would you rather see?



```
OpenFile("...", true, false)
```



or



```
OpenFile("...", writeonly | append)
```

That's kind of a nonsensical/trivial example, but you get the idea.

Share Improve this answer

answered Oct 22, 2008 at 11:14

Follow



[Ferruccio](#)

101k ● 38 ● 229 ● 303

3 I would rather see the first, with code documentation popups. The second seems more ambiguous. – [Petah](#) Jan 5, 2011 at 21:54

2 With the second, there is no need to remember the order of the options, there is no increase in the number of parameters the function must support, and the invocation is self-documenting. – [Stoutie](#) Feb 10, 2013 at 4:22



7

If you have an already complex application, why would you make it more complex by using bitwise flags. I personally wouldn't use this kind flags if there weren't any more upsides then just being cool. And out of experience,



writing it is quite easy, adjusting it later is more harder, let alone if you haven't written it yourself.



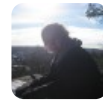
Oh and for the record, I do know how to use them.



Share Improve this answer

answered Oct 22, 2008 at 8:25

Follow



Johnny

437 ● 4 ● 8

1 I agree, they are just harder, when memory was expensive they made sense. Now memory is cheap, and bandwidth is faster there is no point. – user18443 Oct 22, 2008 at 8:27

1 Still every developer should know what bitwise flags are, in case he ever has to deal with them (i.e. legacy code or hardware communication). Seems that many developers are really missing it. – OregonGhost Oct 22, 2008 at 8:55

Yeah I totally agree there, but because someone isn't using a particular technique doesn't necessarily mean he isn't aware or doesn't know it. – Johnny Oct 22, 2008 at 11:23



3



Outside of embedded programming where memory management is critical, their cost outweighs their minimal value. In particular, querying db columns containing bitwise flags produces cryptic SQL that requires referencing the app's code just to determine what value each bitwise operation references.



```
WHERE (ISNULL(flags, 0) & 4096 = 0 -- What does this b
```

If having to query your DB this way doesn't scare you, it should.

Share Improve this answer

answered Jun 24, 2011 at 21:01

Follow



Cory House

15k ● 13 ● 72 ● 89

Indeed, the values are a nuisance to reverse engineer. I have this problem when seeing php error reporting values in apache configs! – [Stoutie](#) Feb 10, 2013 at 4:23

1 This bit of SQL looks familiar... – [Jesse C. Slicer](#) Feb 13, 2013 at 5:27

1 Ha! Hey Jesse - Small world that ya saw this! – [Cory House](#) Feb 14, 2013 at 16:28



2

Bitfields are very useful if you have very tight memory and speed requirements, or when doing low-level hardware programming.



Unless you are doing embedded programming or something similar, you do not need them. There are far more readable, powerful, extensible mechanisms for performing the same task. For instance, Java's EnumSets.



I would expect a good programmer to know them and not use them :-p

Share Improve this answer

answered Oct 22, 2008 at 9:34

Follow



alex



2



"Plenty" is quite generic, we want names! ;-)

Seriously, I come from assembly language and C, and I am quite proficient with bitwise operations. But I wouldn't use them to replace, say 10 unrelated booleans in a class with binary flags in an integer... Unless, perhaps, I have millions of instances of this class!

Should I have to manipulate lot of instance of booleans, I would use some bitset class, to optimize and abstract away these arrays of bits.

Now, bit manipulation fu is almost mandatory in programming, if you want to access to components of RGB[A] values, if you want to call some APIs (even if that's just ORing a number of named flags), etc.

In short, I won't use that in every project I do, but you can't ignore how to do it (a bit like regexes...).

Share Improve this answer

answered Oct 22, 2008 at 10:53

Follow



PhiLho

41.1k ● 6 ● 99 ● 136



0

I still use bitwise operators, though as my main language of choice (and professionally) is C# I often use Flag enums for it :) But yeah, I think alot of people who code in



things like PHP are self taught, and don't know about bitwise operations or the usages of them.



Being self taught myself, I didn't know much about them until I started writing C and C++ code. Now I use them where they make sense :)

Share Improve this answer

answered Oct 22, 2008 at 8:17

Follow



[Sekhat](#)

4,469 ● 6 ● 44 ● 50

This does not provide an answer to the question. To critique or request clarification from an author, leave a comment below their post. – [Ilija Dimov](#) Feb 7, 2015 at 9:09

@IlijaDimov No offense, but this answer was written nearly 7 years ago now, when things were a bit different on stack overflow. The comment you left was not required. – [Sekhat](#) Feb 10, 2015 at 12:20

This answer appeared in the Low Quality Posts review queue a few days ago. I marked it as not an answer based on the current rules for good answer and the comment appeared automatically. – [Ilija Dimov](#) Feb 10, 2015 at 12:34



They always have a place for storing lots of boolean flags in a small amount of space;

0



Just think if you have 32 flags (for example); You could store them all in one "long integer" and just use up 4 bytes, Now imagine you chose to use a long for each flag (just because the code is easier to maintain - you've now got 128 bytes to deal with.





My background is primarily C/C++ so I was brought up on them so I definately use them in every language I use; but I do agree; newbie programmers tend not to even care - all this memory lying around - who cares?

Share Improve this answer

answered Oct 22, 2008 at 9:54

Follow



[Christopher Lightfoot](#)

1,167 ● 1 ● 12 ● 27
