

Re: how to specify power of number

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2008-04/msg03029.html

- *From:* "Dann Corbit" <dcorbit@xxxxxxxxxx>
 - *Date:* Fri, 18 Apr 2008 15:29:59 -0700
-

"Antoninus Twink" <nospam@xxxxxxxxxxxxxxxxxx> wrote in message news:slrng0i6b9.7i7.nospam@xxxxxxxxxxxxxxxxxx

On 18 Apr 2008 at 21:46, user923005 wrote:

If we look at his original post (and knowing he is just turning these TCP/IP addresses into 4 byte integers):

"

```
numericip=atoi(textip[0])*256*256*256+atoi(textip[1])*256*256+atoi(textip[2-])*256+atoi(textip[3])
```

It could clearly be done as {assuming textip[] is an array of unsigned char} as:

```
numericip = (unsigned long)textip[0] << 24 +  
(unsigned long)textip[1] << 16 +  
(unsigned long)textip[2] << 8 +  
(unsigned long)textip[3] ;
```

True, and someone had already suggested this. But the discussion broadened, and someone claimed that using `pow()` was the best way to raise integers to (positive) integer powers: it was this assertion that I was responding to.

Because math coprocessors are awfully fast now, I guess that `pow(2.0,k)` is nearly as fast as bit shifts and additions, but there is a bigger problem with using `pow(2,x)` to find exact powers of 2. It should not be unexpected for `pow(2.0,24)` to return 16777215.999999999 and if you assign that to an integer, it probably won't be what is wanted unless you are clever enough to round it -- `ceil()` won't help either because we might also have seen 16777215.0000001 or something like that. The Cephes collection of math functions {for instance} uses a ratio of a cubic polynomial divided by a quartic polynomial to form the approximation.

Likely, many implementations will actually recognize integral inputs (my C++ compiler definitely does this -- even sometimes complaining about ambiguity over argument type) but there is no reason to expect that to happen.

So I guess I am really agreeing with you. If you want small and exact

Re: how to specify power of number

powers of 2, then bit shifting of unsigned integers is the most sensible method. It will be fast (probably faster than `pow()`) and also exact. The meaning of either construct is clear to any C programmer and so the argument of writing the code that is most clear does not come into play here.

Or {better yet} simply use `inet_aton()` {which is also fine for Winsock, though `WSAStringToAddress()` is an alternative}. That is what would be normally done to retrieve a TCP/IP address from a dotted address string.

HOWEVER!

The OP should keep in mind that `inet_aton()` does NOT support IPv6 and that `getnameinfo()` should be used instead for IPv4/v6 dual stack support.

Good advice.

To the O.P.:
You can't go wrong with "Unix Network Programming" By W. Richard Stevens

Agreed.

Well look here, it appears that we agree on everything. At least in this post.
;-)

** Posted from <http://www.teranews.com> **

.