

Re: main function address

Source: <http://coding.derkeiler.com/Archive/C/ CPP/comp.lang.c/2004-04/0798.html>

From: Leor Zolman (leor_at_bdsoft.com)

Date: 04/07/04

Date: Wed, 07 Apr 2004 12:58:56 GMT

On Wed, 07 Apr 2004 07:47:04 -0400, Lew Pitcher <Lew.Pitcher@td.com> wrote:

.

>

>In 9989-1999 (admittedly, just the draft C99 standard, and not the
>/actual standard itself), the printf() function documentation in
>7.19.6.3 refers the reader to the fprintf() documentation for a
>description of it's input. The fprintf() documentation in 7.19.6.1 says
>of the %p format

>

>~ p The argument shall be a pointer to void. The value of the pointer is
>~ converted to a sequence of printing characters, in an
>~ implementation-defined manner.

>

>So, to satisfy the %p format character, the argument to
>fprintf()/printf() /must/ be a "pointer to void". Since main is a
>"pointer to function returning int", and not a "pointer to void", I
>interpreted the documentation as requiring a cast to void pointer.

Curious -- I've never considered how implicit conversion rules ought to play out in the arena of variadic functions... on one hand, pointers to /anything/ implicitly convert to pointer-to-void, but on the other hand there's no declaration for the receiving pointer-to-void.

Or, does the implicit conversion apply when the pointer value is extracted and cast to void * in the receiving function?

-leor

--

Leor Zolman --- BD Software --- www.bdsoft.com
On-Site Training in C/C++, Java, Perl and Unix
C++ users: Download BD Software's free STL Error Message Decryptor at:
www.bdsoft.com/tools/stlfilt.html