

## Re: main function address

**Source:** [http://coding.derkeiler.com/Archive/C\\_CPP/comp.lang.c/2004-04/0856.html](http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2004-04/0856.html)

---

**From:** Martin Dickopp (*expires-2004-05-31\_at\_zero-based.org*)

**Date:** 04/07/04

Date: Wed, 07 Apr 2004 20:26:00 +0200

pete <pfiland@mindspring.com> writes:

> *Lew Pitcher wrote:*

>>

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

>> Hash: SHA1

>>

>> *pete wrote:*

>> | *Martin Dickopp wrote:*

>> |

>> |> *Lew Pitcher <lpitcher@sympatico.ca> writes:*

>> |>

>> |>

>> |>>#include <stdio.h>

>> |>>#include <stdlib.h>

>> |>>

>> |>>int main(void)

>> |>>{

>> |>> printf("main() at %p\n",(void \*)&main);

>> |>>

>> |>> return EXIT\_SUCCESS;

>> |>>}

>> |>

>> |>Is the cast to `void \*' valid?

>> |

>> |

>> | *No.*

>> |

>> | *In N869, it's one of the common extensions.*

>> |

>> | *J.5.7 Function pointer casts*

>> | *[#2] A pointer to a function may be cast to a pointer to an*

>> | *object or to void, allowing a function to be inspected or*

>> | *modified (for example, by a debugger) (6.5.4).*

>> |

>> | *... which makes it more obviously not part of standard C.*

>>

>> *In 9989-1999 (admittedly, just the draft C99 standard, and not the*

comp.lang.c: Re: main function address

>> /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.  
>  
> I interpret it as meaning that printing the address of a function  
> isn't something that you are guaranteed to be able to do.

So do I. You certainly cannot do it with the '%p' specifier. 6.3.2.3#1 makes it quite clear that only pointers to incomplete or object type can be converted to 'void \*'.

Martin

```
--
,--.      Martin Dickopp, Dresden, Germany
/  ,-- )  http://www.zero-based.org/
\  `-'
 `-.      Debian, a variant of the GNU operating system.
                                     ,= ,---. =.
                                     ((_/)o o(\_/)
                                     `-' (. .)`-'
                                     \_/
```