

## Re: C portability is a myth

**Source:** <http://coding.derkeiler.com/Archive/C/ CPP/comp.lang.c/2005-02/3566.html>

---

**From:** infobahn ([infobahn\\_at\\_btinternet.com](mailto:infobahn_at_btinternet.com))

**Date:** 02/26/05

Date: Sat, 26 Feb 2005 06:37:59 +0000 (UTC)

Allin Cottrell wrote:

>

> *Keith Thompson wrote:*

> > *Mark McIntyre <markmcintyre@spamcop.net> writes:*

>

> > > *Dammit, I showed that it was plain `_wrong_`. Is that not*

> > > *enough?*

> > >

> > > *In what way is the information that `malloc` is ANSI compatible wrong? Please*

> > > *explain.*

> >

> > *I think it's about the mention of the `<malloc.h>` header.*

>

> *It is indeed. The implication of the MSDN page is that `<malloc.h>`*

> *is a standard C header. I have not found such misinformation in*

> *\*nix man pages for functions such as `malloc()`.*

man system

### DESCRIPTION

`system()` executes a command specified in string by calling `/bin/sh -c string`, and returns after the command has been completed. During execution of the command, `SIGCHLD` will be blocked, and `SIGINT` and `SIGQUIT` will be ignored.

### RETURN VALUE

The value returned is `-1` on error (e.g. fork failed), and the return status of the command otherwise. This latter return status is in the format specified in `wait(2)`. Thus, the exit code of the command will be `WEXITSTATUS(status)`. In case `/bin/sh` could not be executed, the exit status will be that of a command that does `exit(127)`.

### CONFORMING TO

ANSI C, POSIX.2, BSD 4.3

(etc)

comp.lang.c: Re: C portability is a myth

The implication of the man page is that `/bin/sh -c`, `SIGCHLD`, `SIGQUIT`, `wait`, `WEXITSTATUS`, and the semantics of the return status, are all defined by ANSI.

Neither MSDN nor man pages define C. They can, however, be used as a quick reference by those who have a firm grasp of the language already. I can read the above descriptions of `malloc` and `system`, and get what I need from them (in the unlikely event that I might have forgotten), without being confused by what you call misinformation. So can you, I'm sure.

Those who can't should be learning C from a good C book, not from man pages or MSDN.