

Re: How to identify the platform & compiler while compiling?

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2007-01/msg03864.html

- *From:* Chris Torek <nospam@xxxxxxxx>
 - *Date:* 28 Jan 2007 08:29:46 GMT
-

In article <1169891565.245849.11090@xx>
aki27 <akilan27@xxxxxxxx> wrote:

Is there a way to find the platform and compiler name during compile time so that conditional compilation can be performed in C? thanks.

It is often (but not always) possible to do so. However, it is often — perhaps even usually — unwise to do so.

In particular, people seem to like to do things like this:

```
#ifdef WIN32
some bizarre special-case code
#endif
#ifdef LINUX
some code
#endif
#ifdef HPUX
some code essentially identical to linux version
#endif
#ifdef BSD
third variant of code again essentially identical
#endif
#ifdef SOLARIS
fourth variant, with some extra goop for Solaris threads
#endif
```

What one *should* do, instead, is:

```
% cat win.c
... special windows-specific code ...
(no ifdefs at all here)
```

```
% cat everythingelse.c
... generic Unixy version of code ...
#ifdef USE_POSIX_THREADS
extra goop for POSIX threads
```

Re: How to identify the platform & compiler while compiling?

```
#endif
#ifdef USE_SOLARIS_THREADS
extra goop for Solaris threads
#endif
... more generic code ...
```

In other words, take the special (in this case, Windows-specific) code and put it in a special file used only on the special system. Take the common code and put it in a common file, which is used on the common systems. If "#ifdef"s are needed -- which should be made as rare as possible by separating out the "special" code -- make them conditional on the *feature* being exploited, not the *system*. Here, although "Solaris threads" is the conditional, it is the *feature*, not the system, that is being tested -- and note that "Solaris threads" are not even present in sufficiently ancient versions of Solaris. The code still works on those, even though the platform and compiler are "Solaris"; you simply compile without turning on "USE_SOLARIS_THREADS".

(You can of course set the default configuration based on the platform and compiler. Just give the software engineer a clear and obvious place to *change* those settings. If the default for FooBlick systems turns out to be incorrect for the FooBlick 42, the place to adjust that will then be clear and obvious.)

In-Real-Life: Chris Torek, Wind River Systems
Salt Lake City, UT, USA (40°39.22'N, 111°50.29'W) +1 801 277 2603
email: forget about it <http://web.torek.net/torek/index.html>
Reading email is like searching for food in the garbage, thanks to spammers.