You already know what your target architecture is (or at least you should)

devblogs.microsoft.com/oldnewthing/20060906-07

September 6, 2006

Raymond Chen

Sometimes the questions I see make me shake my head in wonderment.

> How do I determine programmatically what processor architecture my program was compiled for? I want the x86 version of my program to behave differently from the ia64 version. Is there some API I can call?

Note that this person isn't asking whether the program is running on 64-bit Windows. This person wants the program to detect whether it was compiled with an x86 compiler, an ia64 compiler, an amd64 compiler, or whatever.

But why do you need an API for this? You already know what your target architecture is because you compiled it yourself!

It so happens that the Microsoft Visual C++ compiler defines several symbols for you automatically (assuming you're not running the compiler in "strict ANSI compliance mode"). If you're willing to tie yourself to the Microsoft Visual C++ compiler, you can use those symbols.

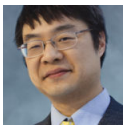
```
#ifdef _M_IX86
    // the target architecture is x86
#else
    // the target architecture is something else
#endif
```

If you don't want to tie yourself to a particular compiler, you'll have to pass that information yourself. For example, you could have your x86 `makefile` pass `-DBUILDING_FOR_x86` in the compiler flags, while having the ia64 `makefile` pass `-DBUILDING_FOR_ia64`, and so on. This is the approach used by the `build` utility that comes with the Windows DDK: The DDK's makefile system defines a variety of symbols that programs (and other makefiles) can use to alter their behavior depending on the compilation environment:

```
// assumes you use makefile.def
#if defined(_X86_)
    // the target architecture is x86
#elif defined(_IA64_)
    // the target architecture is Intel ia64
#elif defined(_AMD64_)
    // the target architecture is AMD x86-64
#else
    // some other architecture
#endif
```

As we saw in the earlier article, you can also use the `_WIN64` symbol to detect that the target platform is 64-bit Windows.

But the point is that this is all something you control yourself. You're the one who is compiling the program. You know what your target architecture is. No need to ask somebody to tell you something that is already entirely under your own control.



Raymond Chen

Follow