The first word on the command line is the program name only by convention

devblogs.microsoft.com/oldnewthing/20060515-07

May 15, 2006

Raymond Chen

The format of the command line returned by `GetCommandLine` is "`program args`", but this is only a convention. If you pass `NULL` for the `lpApplicationName` to the `CreateProcess` function, then the `CreateProcess` function will treat the first word of the `lpCommandLine` as the program name. However, if you pass a value for `lpApplicationName`, then that string determines the program that is run, and the string passed as the `lpCommandLine` is not used for that purpose. This means that if somebody runs your program with the following parameters to the `CreateProcess` function

`lpApplicationName`	`=`	`"C:\Path\To\Program.exe"`
`lpCommandLine`	`=`	`"slithy toves"`

then when your program calls the `GetCommandLine` function, it will get the string `"slithy toves"`, which doesn't give your program much help at all in determining its own name or location. If your program needs to determine its own name and location, use the `GetModuleFileName` function, as I noted some time ago. What is the point of letting a program specify something different as the first word on the command line from the actual program being run? There isn't much point to it in Windows, although it is used to greater effect in unix, where you can run a program under various "alias" names, executing one program but lying to it and putting a different name at the start of the command line. Some programs are specially designed to be run this way and alter their behavior depending on the "alias" name they were given. For example, the visual editor runs in screen mode if its name is given as `"vi"` but in line mode if its name is given as `"ex"`. Although extremely few Windows programs use this quirk (I am not aware of any myself), the behavior nevertheless is supported, and you need to be aware of it when writing your own program, even if you don't intend to use it. For example, if you forget to repeat the program name on the command line and create the process like this

`lpApplicationName`	`=`	`"C:\Path\To\Program.exe"`

`lpCommandLine`	`=`	`"arg1 arg2"`

then when that program runs, you will most likely see it ignore the `arg1` because it thinks that `arg1` is just the program name. If that program is a console program that uses the C runtime startup code, then it will receive its parameters as

`argv[0]`	`=`	`"arg1"`
`argv[1]`	`=`	`"arg2"`

As I noted earlier, most console programs merely ignore their `argv[0]` since that slot is just the program name. (In this case, it's the alias program name, but the program being run doesn't know that.) Similarly, if the program is a Windows program that uses the C runtime startup code, then the C runtime startup code will merely skip over the first word on the command line, passing `"arg2"` to the `WinMain` function as its `lpCmdLine`.

What was the point of all this discussion? Two things. First, that if you are launching other programs and passing an explicit `lpApplicationName`, then it behooves you to format the command line in a compatible manner. Otherwise, the results may not be what you expect. Second, that you as a program should not use the first token on the command line to control any security decisions since the value is controlled by the launcher and need not have any connection to reality.

Raymond Chen

Follow