

Re: Abolute path of argv[0]

Source: <http://linux.derkeiler.com/Newsgroups/comp.os.linux.development.apps/2005-11/0275.html>

phil-news-nospam_at_ipal.net

Date: 11/14/05

Date: 14 Nov 2005 19:31:27 GMT

On 12 Nov 2005 05:59:47 -0800 stork@storkyak.com wrote:

|>Please rethink what you would like to do. There is no good reason to do
|>this and many good reasons not to.

|
| This is a bit of a Windows convention... you could have an application
| cd to itself so that it can find data files associated with it. For
| example, if my application were installed in Program Files \ Mightyware
| \ Commodity Server \, I can chdir("config") to get the configuration
| file that goes with it. This is particularly useful to do in an
| environment where a "root path" does not exist. [Drive letters again!]

|
| It -seems- like all configuration files live in etc for a system so
| that... in other words, do correct me if I'm wrong as I'm a Linux Noob
| myself, but under Linux when an apps installs its libraries go to \lib,
| its "executable" goes to \usr, and its configuration stuff goes in
| \etc, and user data goes into \home\username, and all of this is by
| convention and there is no need to cd in your application because you
| can just use the straight up path to your configuration and other
| files.

I can understand the value for a large application to include most of its resources outside the actual executable file itself. Resources can be many things, depending on what the application actually is. For example a bunch of font files might be a resource for something that renders text into graphics.

However, limiting such resources to live only in the same directory subtree where the application is, is a bad design. Instead of finding where such resources are located by means of backtracking to where the application is located (regardless of how symlinks managed to get it executed), it should instead find them via a configurable element within "/etc". This could be a configuration file such as "/etc/myapp.cf" or a subdirectory of several files and/or symlinks such as "/etc/myapp/". Wherever the resources are installed into should be identified there. The application would have its simple name hard coded, or derive it from the simple form of argv[0] (just the base part of a full path if that is present), and use that to find the configuration. Then from there, find the resources. This then gives the

system administrator the ability to change where the resources are located for purposes that vary from simply moving the resource files to another filesystem, to having more than one set of resources available and choosing which to use. A more sophisticated application would also look in the user home directory for an alternative set of resources, as well as be able to accept alternatives via command line options or environment variables (with consideration for security in certain applications that might need that).

Additionally, the Unix, BSD, and Linux worlds also tend to share many kinds of resources between programs, using standards as much as possible, rather than the proprietary efforts that tend to predominate among Windows. For example, a common set of font resources could be used both by an office word processing program, as well as a TV video titling editor. In that sense, resources like that really shouldn't "belong" to any one program, anyway.

Given your incorrect (but excusable considering the experience) usage of the backwards slash, I'm guessing you don't have enough experience with Unix or Linux to understand its scheme of locating resources. Work with it a while and just see what all is going on.

--

| Phil Howard KA9WGN | <http://linuxhomepage.com/> | <http://ham.org/> |
| (first name) at ipal.net | <http://phil.ipal.org/> | <http://ka9wgn.ham.org/> |
