
Subject: GetExeFilePath() in Linux
Posted by [zsolt](#) on Mon, 06 Nov 2006 23:47:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

I have implemented GetExeFilePath(). It is a little bit tricky, I dont know if it is able to work in BSD systems, but OK for Linux. It is based on the proc filesystem, as there is no standard posix function to this problem (AFAIK).

```
String GetExeFilePath()
{
    char h[_MAX_PATH + 1];
    String link;
    link << "/proc/" << getpid() << "/exe";
    int ret = readlink(link, h, _MAX_PATH);
    if(ret == -1 || ret > _MAX_PATH)
        return "";
    h[ret] = 0;
    return FromSystemCharset(h);
}
```

Subject: Re: GetExeFilePath() in Linux
Posted by [mirek](#) on Mon, 06 Nov 2006 23:53:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

Wow, this is really interesting solution to the problem! (Personally, I was thinking about searching in the path....).

Mirek

Subject: Re: GetExeFilePath() in Linux
Posted by [guido](#) on Tue, 07 Nov 2006 00:53:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

Yes, that's thoroughly non-portable voodoo - Linux only.

Solaris has /proc too.
There it seems to be /proc/self/object/something

But on further investigation, Sun's Unix Systems had getexecname() since like forever:

<http://docs.sun.com/app/docs/doc/816-5168/6mbb3hrbp?a=view>

BSDs? No idea! Only thing I know is, they don't have /proc at all, MacOS X included.

Guido

Subject: Re: GetExeFilePath() in Linux
Posted by [mirek](#) on Tue, 07 Nov 2006 08:35:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

OK, so what we will do?

Separate branches for various posixes are OK, but before adding Linux version, I would like to know about BSDs.... (it has not much sense to have this in Linux only I think, win32 being specific affair).

Mirek

Subject: Re: GetExeFilePath() in Linux
Posted by [zsolt](#) on Tue, 07 Nov 2006 09:09:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

After using google:
On FreeBSD u can use /proc/\$pid/file. The only problem is, that /proc is not mounted by default

Subject: Re: GetExeFilePath() in Linux
Posted by [mirek](#) on Thu, 09 Nov 2006 19:27:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

What about the original idea - perform search of argv[0] in PATH variable? Alternatively, as FreeBSD version?

Mirek

Subject: Re: GetExeFilePath() in Linux
Posted by [masu](#) on Thu, 09 Nov 2006 22:08:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Thu, 09 November 2006 20:27What about the original idea - perform search of argv[0] in PATH variable? Alternatively, as FreeBSD version?

Mirek

This could be an appropriate solution for FreeBSD.
A silly question, but why do you need the location at all ?

Matthias

Subject: Re: GetExeFilePath() in Linux
Posted by [mirek](#) on Thu, 09 Nov 2006 23:13:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

masu wrote on Thu, 09 November 2006 17:08luzr wrote on Thu, 09 November 2006 20:27What about the original idea - perform search of argv[0] in PATH variable? Alternatively, as FreeBSD version?

Mirek

This could be an appropriate solution for FreeBSD.
A silly question, but why do you need the location at all ?

Matthias

Well, a sort of win32-ism

E.g. to put some data files there. Or .log files in debug version. I know it is not quite appropriate for Posix and in fact, that is why it was not defined for posix.

But Zoltan obviously has a use for it

Mirek

Subject: Re: GetExeFilePath() in Linux
Posted by [zsolt](#) on Fri, 10 Nov 2006 00:01:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

Quote:But Zoltan obviously has a use for it
I just wanted to make my app installable to anywhere in the filesystem. I didn't want to hardcode a path to resource files.

I think, on FreeBSD GetExeFilePath() would try proc first and continue with searching when proc is not mounted.

Btw, Zoltan and Zsolt are different Hungarian names, my name is Zsolt

Subject: Re: GetExeFilePath() in Linux

Posted by [masu](#) on Fri, 10 Nov 2006 08:22:02 GMT

[View Forum Message](#) <> [Reply to Message](#)

zsolt wrote on Fri, 10 November 2006 01:01I just wanted to make my app installable to anywhere in the filesystem. I didn't want to hardcode a path to resource files.

Will you execute your program with giving the complete path on the command line then?

Quote:I think, on FreeBSD GetExeFilePath() would try proc first and continue with searching when proc is not mounted.

I agree

Matthias

Subject: Re: GetExeFilePath() in Linux

Posted by [fallingdutch](#) on Fri, 10 Nov 2006 09:21:34 GMT

[View Forum Message](#) <> [Reply to Message](#)

masu wrote on Thu, 09 November 2006 23:08

A silly question, but why do you need the location at all ?

I don't see any use for it, too

you store your app in eg /usr/bin or /usr/local/bin and your ressources either in ~/.Appname/ or /var/Appname and your configuration in /etc/Appname.conf or ~/.Appname/Appname.conf

Bas

Subject: Re: GetExeFilePath() in Linux

Posted by [zsolt](#) on Fri, 10 Nov 2006 09:22:31 GMT

[View Forum Message](#) <> [Reply to Message](#)

masu wrote on Fri, 10 November 2006 09:22zsolt wrote on Fri, 10 November 2006 01:01I just wanted to make my app installable to anywhere in the filesystem. I didn't want to hardcode a path to resource files.

Will you execute your program with giving the complete path on the command line then?

No, but some admins prefer installing 3rd party apps not with /usr prefix, but with /usr/local or /opt. And some prefer putting it to e.g. /opt/appdir-version and using symlinks to switch easily between versions.

Subject: Re: GetExeFilePath() in Linux
Posted by [mirek](#) on Fri, 10 Nov 2006 09:34:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

fallingdutch wrote on Fri, 10 November 2006 04:21masu wrote on Thu, 09 November 2006 23:08
A silly question, but why do you need the location at all ?
I don't see any use for it, too

you store your app in eg /usr/bin or /usr/local/bin and your ressources either in ~/.Appname/ or /var/Appname and your configuration in /etc/Appname.conf or ~/.Appname/Appname.conf

Bas

Well, that is current status quo - one that makes adding and removing applications in linux somewhat difficult.

What many would like to see is macosx like handling - all application specific files (minus configuration) are in single directory. You want to add application - you just copy the directory with application. You want to delete it - remove dir.

Mirek

Subject: Re: GetExeFilePath() in Linux
Posted by [masu](#) on Fri, 10 Nov 2006 09:48:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Normally you run your application as a user and therefore the application does not have any write access to system directories llike /usr/local/bin. Reading is possible of course. But if you don't have write privileges on your application path, you also cannot store any files there. So it does not matter where the application is located in the system unless you are using it as root, which is not advisable.

And if root installs software under /opt then it should also be read-restricted for normal users and normally these dirs are also added to PATH to comfortably execute them.

Matthias

Subject: Re: GetExeFilePath() in Linux
Posted by [zsolt](#) on Fri, 10 Nov 2006 11:17:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

Quote: But if you don't have write privileges on your application path, you also cannot store any files there.

I have a directory structure like this:

bin/appname - this is the binary

share/appname/ - this is a directory for document and configuration templates and other readonly resources

I have a lot of document and configuration templates and the application wants to READ them. I don't want to compile them into the app because they are extensible by installing additional packages.

I can write into the documentation, that resources have to be in e.g. /usr/local/share/appname/ directory, but it is not very admin friendly. It is much better to get these resources with the next algorithm:

```
GetFileFolder(GetExeFilePath()) + "../share/" + appname + "/something.resource"
```

Subject: Re: GetExeFilePath() in Linux

Posted by [mirek](#) on Fri, 10 Nov 2006 11:23:46 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quote:

```
GetFileFolder(GetExeFilePath()) + "../share/" + appname + "/something.resource"
```

Shorter equivalent:

```
GetExeDirFile("../share/" + appname + "/something.resource")
```

Mirek

Subject: Re: GetExeFilePath() in Linux

Posted by [zsoft](#) on Fri, 10 Nov 2006 11:35:08 GMT

[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Fri, 10 November 2006 12:23Quote:

```
GetFileFolder(GetExeFilePath()) + "../share/" + appname + "/something.resource"
```

Shorter equivalent:

```
GetExeDirFile("../share/" + appname + "/something.resource")
```

Mirek

This is why U++ is my favourite development environment

Subject: Re: GetExeFilePath() in Linux

Posted by [masu](#) on Fri, 10 Nov 2006 15:28:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

zsolt wrote on Fri, 10 November 2006 12:17I have a directory structure like this:

bin/appname - this is the binary

share/appname/ - this is a directory for document and configuration templates and other readonly resources

I have a lot of document and configuration templates and the application wants to READ them. I don't want to compile them into the app because they are extensible by installing additional packages.

I can write into the documentation, that resources have to be in e.g. /usr/local/share/appname/ directory, but it is not very admin friendly. It is much better to get these resources with the next algorithm:

```
GetFileFolder(GetExeFilePath()) + "../share/" + appname + "/something.resource"
```

Now I understand your problem and I vote for including to search via proc first and if it is not available in argv[0] and PATH.

Matthias

Subject: Re: GetExeFilePath() in Linux

Posted by [fallingdutch](#) on Fri, 10 Nov 2006 19:08:20 GMT

[View Forum Message](#) <> [Reply to Message](#)

you could write a config file in /etc and read the dir where the app-data (eg extensions) is stored (the way eg MySQL works)

Bas

Subject: Re: GetExeFilePath() in Linux

Posted by [mirek](#) on Sat, 11 Nov 2006 11:48:06 GMT

[View Forum Message](#) <> [Reply to Message](#)

OK, this is my final version:

```
const char *procexepath_() {
    static char h[_MAX_PATH + 1];
    ONCELOCK {
        char link[100];
        sprintf(link, "/proc/%d/exe", getpid());
        int ret = readlink(link, h, _MAX_PATH);
        if(ret > 0 && ret < _MAX_PATH)
            h[ret] = '\0';
        else
```

```
*h = '\0';  
}  
return h;  
}
```

```
String GetExeFilePath()
```

```
{  
    static String exepath;  
    ONCELOCK {  
        const char *exe = procexepath_();  
        if(exe)  
            exepath = exe;  
        else {  
            String x = Argv0__;  
            if(IsFullPath(x) && FileExists(x))  
                exepath = x;  
            else {  
                exepath = GetHomeDirFile("upp");  
                Vector<String> p = Split(FromSystemCharset(Environment().Get("PATH")), ':');  
                if(x.Find('/') >= 0)  
                    p.Add(GetCurrentDirectory());  
                for(int i = 0; i < p.GetCount(); i++) {  
                    String ep = NormalizePath(AppendFileName(p[i], x));  
                    if(FileExists(ep))  
                        exepath = ep;  
                }  
            }  
        }  
    }  
    return exepath;  
}
```

Please check.

Mirek

P.S.: I need the name of executable in log files, without using the heap and before entering the main - that is why I have separated "procexepath_"...

Subject: Re: GetExeFilePath() in Linux

Posted by [zsolt](#) on Sat, 11 Nov 2006 12:00:41 GMT

[View Forum Message](#) <> [Reply to Message](#)

First impression:

On FreeBSD you can use /proc/\$pid/file, not /proc/\$pid/exe (as on Linux).

Subject: Re: GetExeFilePath() in Linux
Posted by [mirek](#) on Sat, 11 Nov 2006 12:37:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

OK.

Mirek

Subject: Re: GetExeFilePath() in Linux
Posted by [masu](#) on Sun, 12 Nov 2006 17:11:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

I tried the latest version and got the right path with mounted proc and also without.

I had to make a tiny change (marked red, therefore the ugly formatting).

luzr wrote on Sat, 11 November 2006 12:48OK, this is my final version:

```
const char *procexepath_() {
    static char h[_MAX_PATH + 1];
    ONCELOCK {
        char link[100];
        sprintf(link, "/proc/%d/exe", getpid());
        int ret = readlink(link, h, _MAX_PATH);
        if(ret > 0 && ret < _MAX_PATH)
            h[ret] = '\0';
        else
            *h = '\0';
    }
    return h;
}
```

```
String GetExeFilePath()
{
    static String exepath;
    ONCELOCK {
        const char *exe = procexepath_();
        if(*exe)
            exepath = exe;
        else {
            String x = Argv0__;
            if(IsFullPath(x) && FileExists(x))
                exepath = x;
            else {
                exepath = GetHomeDirFile("upp");
                Vector<String> p = Split(FromSystemCharset(Environment()).Get("PATH"), ':');
                if(x.Find('/') >= 0)

```

```
p.Add(GetCurrentDirectory());
for(int i = 0; i < p.Count(); i++) {
    String ep = NormalizePath(AppendFileName(p[i], x));
    if(FileExists(ep))
        exepath = ep;
}
}
}
}
return exepath;
}
```

Please check.

Mirek

P.S.: I need the name of executable in log files, without using the heap and before entering the main - that is why I have separated "procexepath_"...

Also, I think we should delete the green line since the result defaults to \$HOME/app_name even if it does not exist. The application has to be in the PATH or in the current dir if it was not specified with the complete path on the command line. So this assignment is useless in my opinion since we check all possibilities.

Function SetExeTitle does not exist anymore, commented out in idewin.cpp:

```
#ifdef _DEBUG
    SetExeTitle("debugide");
#else
    SetExeTitle("theide");
#endif
```

Matthias

Subject: Re: GetExeFilePath() in Linux
Posted by [guido](#) on Mon, 20 Nov 2006 06:55:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I think there is a simpler implementation for POSIX.
Acrobat Reader, OpenOffice, Firefox, ROX-Filer.. achieve relocatability by means of a simple shell launcher script:

```
#!/bin/sh
```

```
PROG=SampleApp
```

```
APP_DIR=`dirname "$0"`  
APP_DIR=`cd "$APP_DIR";pwd`; export APP_DIR  
  
exec "$APP_DIR/SampleApp" "$@"
```

Then, from inside the app simply
`app_dir = strdup(getenv(APP_DIR));`

Now I thought, why not port this shell script to C++?:

```
String GetExeFilePath()  
{  
    static String exepath;  
    ONCELOCK {  
        String x = Argv0___;  
        if(IsFullPath(x) && FileExists(x))  
            exepath = x;  
        else {  
            String cwd = GetCurrentDirectory();  
            String filename = GetFileName(x);  
            x = x.Left(x.ReverseFind('/'));  
            chdir(ToSystemCharset(x));  
            exepath = GetCurrentDirectory() + "/" + filename;  
            chdir(ToSystemCharset(cwd));  
        }  
    }  
    return exepath;  
}
```

I don't know if there are tricky circumstances of the environment that would break this, but so far it seems to be working.

Guido

Subject: Re: GetExeFilePath() in Linux
Posted by [mirek](#) on Wed, 22 Nov 2006 15:30:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Well, I was thinking about this last 4 days (of course, not all the time) and I must admit I am completely clueless how is this supposed to work...

Subject: Re: GetExeFilePath() in Linux
Posted by [guido](#) on Wed, 22 Nov 2006 18:04:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Wed, 22 November 2006 16:30 Well, I was thinking about this last 4 days (of course, not all the time) and I must admit I am completely clueless how is this supposed to work...

Well, as a shell script this method works. You can download e.g. OpenOffice or Firefox from their original site, extract into your home dir and run from there. And that's how they do it. Just open the launch wrapper and see.

If, however, this works without going through a shell in all cases, I'm not sure.

I created a sample console app, which prints exepath to stdout, and a gui app, which pops up a dialog with the exepath, and that worked fine, launching from both a terminal or the file-manager.

Guido

Subject: Re: GetExeFilePath() in Linux
Posted by [mirek](#) on Wed, 22 Nov 2006 18:10:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

OK, what seem wrong to me:

You have your "foo" application somewhere on the PATH. You start it by typing "foo" -> no '\ in the name -> `x = x.Left(x.ReverseFind('/'))` crashes.

Mirek

Subject: Re: GetExeFilePath() in Linux
Posted by [guido](#) on Wed, 22 Nov 2006 21:53:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Wed, 22 November 2006 19:10 OK, what seem wrong to me:

You have your "foo" application somewhere on the PATH. You start it by typing "foo" -> no '\ in the name -> `x = x.Left(x.ReverseFind('/'))` crashes.

Mirek

Oops, sorry

Well, then, when `x.ReverseFind('/')` fails, a path search, like your current implementation, must be done.

Or this is how OO.org does it:

```
guido@Sid:~$ cat /usr/bin/openoffice.org-2.0
```

```
#!/bin/sh
```

```
exec /etc/openoffice.org-2.0/program/soffice "$@"
```

I only suggest it, because I'm familiar with this method, and because it's portable across POSIX systems. I know, it is how ISPs deal with it, to make their software relocatable and distribution independent.

Guido

Subject: Re: GetExeFilePath() in Linux
Posted by [guido](#) on Wed, 22 Nov 2006 22:07:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

...ISVs, not ISPs, of course

Guido
