Subject: Is it possible ?

Posted by Xemuth on Thu, 16 May 2019 11:55:42 GMT

View Forum Message <> Reply to Message

Hello Community !

Today I asked myself a question:

Here is part on code on my "OldVersion.exe"

```
    const int VERSION = 1;
    std::string s = std::to_string(VERSION);
    bool needAnUpdate=false;
    s << " " << &needAnUpdate;
    char* cmd =(char*) s.c_str();
    BOOL result = CreateProcess(
"C:\\Upp\\upp\\out\\MyApps\\MINGW.Debug.Debug_Full.Noblitz\\newVersion.exe", //New Version const int is 2
                   cmd, NULL, NULL, // here we past version of current .exe
                   NULL, NULL, NULL, NULL,
                   &lpStartupInfo,
                   &lpProcessInfo
                   );
 if (result)
 {
  if(needAnUpdate){
   //Then my soft need to update itself.
  }
 }
```

As you can see, this old version have const int VERSION = 1.

here is NewVersion.exe :

```
 const int VERSION =2;
 if(argc > 1){
  Upp::String arg(argv[1]); //transform args to string
  Upp::String version = arg.Left(arg.Find(" ") -1); //here I get version
  pp::String addrBool = arg.Right(arg.GetCount() - arg.Find(" ")); //Here I got adresse of my Boolean at string format

  int versionToTest = std::stoi(version.ToStd());
  if (VERSION > versionToTest){
   // Here I cast my addrBool (string) to bool*  (I dont know how to do it :/ )
  }
  //Here I end the programme
 }
```

with this "methode" I should be able to know if oldVersion is outdated.

But it's possible ? It's a good idea to do those "stranges" things ? Maybe I should just use a textFile holding current version of newVersion.exe ?

Thanks for taking time to read and respond my Strange question.

Best Regard,

Xemuth

Subject: Re: Is it possible ?
Posted by mr_ped on Thu, 16 May 2019 19:19:56 GMT
View Forum Message <> Reply to Message

Under modern OS the processes have virtual address space, separated by from each other, i.e. when the new exe is running, the memory address from old exe is invalid and not accessible.

You can do some IPC (inter process communication) to do it like this, but I'm not sure why.

If you are only interested in version value, I would probably rather include that in some resource file, and use resource reading to read values from your current exe, and from the other exe and compare those, then you don't need to run the other exe at all to figure out if you need update.

Is there some other requirement forcing you to actually put the value into code?

If yes, you can still do this in simple way, supporting in `main` function extra command line option like `--getversion`, which will return version value as exit code (if values 1..255 are enough for you), or you can output it to STDOUT and execute the other exe in such way, that you will collect all STDOUT in first exe, but this is getting somewhat over-complicated.

As you are writing something about text file, it sounds to me like you can add the version value straight into resources (actually if we are talking about Windows .exe files, those already have some kind of meta data - check properties on exe files, there's often various info who's author, version numbers, etc.. = that's all meta data specified in some resource/manifest file, and you can read those somehow also in code, just by reading the .exe file as data, no need to execute it.

Subject: Re: Is it possible ?
Posted by Xemuth on Fri, 17 May 2019 07:08:24 GMT
View Forum Message <> Reply to Message

Hello mr_ped,

thanks for your reply.

Quote:Under modern OS the processes have virtual address space, separated by from each other
I did not know that.

I will write of simple file. Gonna be easier :blush: