Posted by mirek on Mon, 17 Mar 2008 13:55:18 GMT
View Forum Message <> Reply to Message

Today I had a nice idea of "fixing" console app "execute" behaviour by launching xterm with -e option....

And I have tried and failed miserably:

```
void LocalHost::Launch(const char *_cmdline, bool console)
{
 String cmdline = FindCommand(exedirs, _cmdline);
 PutVerbose(cmdline);
#ifdef PLATFORM_WIN32
 if(console)
  cmdline = GetExeFilePath() + " ! " + cmdline;
 int n = cmdline.GetLength() + 1;
 Buffer<char> cmd(n);
 memcpy(cmd, cmdline, n);
 SECURITY_ATTRIBUTES sa;
 sa.nLength = sizeof(SECURITY_ATTRIBUTES);
 sa.lpSecurityDescriptor = NULL;
 sa.bInheritHandle = TRUE;
 PROCESS_INFORMATION pi;
 STARTUPINFO si;
 ZeroMemory(&si, sizeof(STARTUPINFO));
 si.cb = sizeof(STARTUPINFO);
 Buffer<char> env(environment.GetCount() + 1);
 memcpy(env, environment, environment.GetCount() + 1);
 if(CreateProcess(NULL, cmd, &sa, &sa, TRUE,
          NORMAL_PRIORITY_CLASS|CREATE_NEW_CONSOLE,
           ~env, NULL, &si, &pi)) {
  CloseHandle(pi.hProcess);
  CloseHandle(pi.hThread);
 }
 else
  PutConsole("Unable to launch " + String(_cmdline));
#endif
#ifdef PLATFORM_POSIX
 if(console) // why this does not work?!
  cmdline = "/usr/bin/xterm -hold -e " + cmdline;
 Buffer<char> cmd_buf(strlen(cmdline) + 1);
 char *cmd_out = cmd_buf;
 Vector<char *> args;
 const char *p = cmdline;
```

For some reason, if I try to launch xterm, ide (launching program) crashes miserably. I can launch any other X11 program without a problem...

Does anybody know what can be the cause?

Mirek

Subject: Re: The very strange xterm issue
Posted by mdelfede on Mon, 17 Mar 2008 14:35:45 GMT
View Forum Message <> Reply to Message

uhmmmm... what's the rest of your code ? I mean, where you launch xterm....

Max

Subject: Re: The very strange xterm issue
Posted by mirek on Mon, 17 Mar 2008 15:22:01 GMT
View Forum Message <> Reply to Message

In ide/Host.cpp:

```
void LocalHost::Launch(const char *_cmdline, bool console)
{
 String cmdline = FindCommand(exedirs, _cmdline);
 PutVerbose(cmdline);
#ifdef PLATFORM_WIN32
 if(console)
  cmdline = GetExeFilePath() + " ! " + cmdline;
 int n = cmdline.GetLength() + 1;
 Buffer<char> cmd(n);
 memcpy(cmd, cmdline, n);
 SECURITY_ATTRIBUTES sa;
 sa.nLength = sizeof(SECURITY_ATTRIBUTES);
 sa.lpSecurityDescriptor = NULL;
 sa.bInheritHandle = TRUE;
 PROCESS_INFORMATION pi;
 STARTUPINFO si;
 ZeroMemory(&si, sizeof(STARTUPINFO));
 si.cb = sizeof(STARTUPINFO);
 Buffer<char> env(environment.GetCount() + 1);
 memcpy(env, environment, environment.GetCount() + 1);
```

```
   if(CreateProcess(NULL, cmd, &sa, &sa, TRUE,
            NORMAL_PRIORITY_CLASS|CREATE_NEW_CONSOLE,
             ~env, NULL, &si, &pi)) {
    CloseHandle(pi.hProcess);
    CloseHandle(pi.hThread);
   }
   else
    PutConsole("Unable to launch " + String(_cmdline));
#endif
#ifdef PLATFORM_POSIX
/* if(console) // why this does not work?!
    cmdline = "/usr/bin/xterm -hold -e " + cmdline;*/
   Buffer<char> cmd_buf(strlen(cmdline) + 1);
   char *cmd_out = cmd_buf;
   Vector<char *> args;
   const char *p = cmdline;
   const char *b = p;
   while(*p && (byte)*p > ' ')
    if(*p++ == '\"')
     while(*p && *p++ != '\"')
      ;
   const char *app = cmd_out;
   args.Add(cmd_out);
   memcpy(cmd_out, b, p - b);
   cmd_out += p - b;
   *cmd_out++ = '\0';

   while(*p)
    if((byte)*p <= ' ')
     p++;
    else {
     args.Add(cmd_out);
     b = p;
     while(*p && (byte)*p > ' ')
      if(*p++ == '\"')
      {
       memcpy(cmd_out, b, p - b - 1);
       cmd_out += p - b - 1;
       b = p;
       while(*p && *p != '\"')
        p++;
       memcpy(cmd_out, b, p - b);
       cmd_out += p - b;
       if(*p == '\"')
        p++;
       b = p;
      }
     memcpy(cmd_out, b, p - b);
```

```
    cmd_out += p - b;
    *cmd_out++ = '\0';
  }

 args.Add(NULL);

 ONCELOCK {
  struct sigaction sigchld_action;
    memset(&sigchld_action, 0, sizeof(sigchld_action));
    sigchld_action.sa_handler = sCleanZombies;
    sigaction(SIGCHLD, &sigchld_action, NULL);
 }

 pid_t pid = fork();
 if(pid == 0)
 {
  const char *from = environment;
  Vector<const char *> env;
  while(*from)
  {
   env.Add(from);
   from += strlen(from) + 1;
  }
  env.Add(NULL);
  const char **envp = env.Begin();
  execve(args[0], args, (char *const *)envp);
 }
 LLOG("Launch pid: " << pid);
 sPid().Add(pid);
#endif
}
```

Mirek

Subject: Re: The very strange xterm issue
Posted by mdelfede on Mon, 17 Mar 2008 20:31:41 GMT

luzr wrote on Mon, 17 March 2008 16:22In ide/Host.cpp:


```
.............................
#ifdef PLATFORM_POSIX
/* if(console) // why this does not work?!
  cmdline = "/usr/bin/xterm -hold -e " + cmdline;*/
................
```

Mirek

I don't understand your question.... If I add :

....
#ifdef PLATFORM_POSIX
    console = true;
 if(console) // why this does not work?!
  cmdline = "/usr/bin/xterm -hold -e " + cmdline;
....

That works quite well, opens an xterm and runs app.
The only stuff I don't understand is why 'console' is set to false on entering Launch().
BTW, no crash at all....

Max

Subject: Re: The very strange xterm issue
Posted by mirek on Mon, 17 Mar 2008 23:13:18 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Mon, 17 March 2008 16:31luzr wrote on Mon, 17 March 2008 16:22In
ide/Host.cpp:

....
#ifdef PLATFORM_POSIX
/* if(console) // why this does not work?!
  cmdline = "/usr/bin/xterm -hold -e " + cmdline;*/
....

Mirek

I don't understand your question.... If I add :

....
#ifdef PLATFORM_POSIX
    console = true;
 if(console) // why this does not work?!
  cmdline = "/usr/bin/xterm -hold -e " + cmdline;
....

That works quite well, opens an xterm and runs app.

The only stuff I don't understand is why 'console' is set to false on entering Launch().
BTW, no crash at all....

Max

Too weird. On my rig, it ends with broken X11 (most likely) - it starts returning a lot of X11 errors and eventually ends on segfault.

Looks like some weird issue with fork, hard to say...

Hm, console == true if application is build with CONSOLE mode. It is true I did tested only with such apps, while you might be trying with X11 apps.. maybe a clue?!

Mirek

Subject: Re: The very strange xterm issue
Posted by mdelfede on Tue, 18 Mar 2008 08:27:39 GMT

luzr wrote on Tue, 18 March 2008 00:13
Too weird. On my rig, it ends with broken X11 (most likely) - it starts returning a lot of X11 errors and eventually ends on segfault.

Looks like some weird issue with fork, hard to say...

Hm, console == true if application is build with CONSOLE mode. It is true I did tested only with such apps, while you might be trying with X11 apps.. maybe a clue?!

Mirek

Well, usually I like to have a console even on X11 apps, so I can see error and log messages, so I put console=true on Host.cpp and I got a running xterm that launches the X11 gui app.
All goes ok, no error and no crashes, and the xterm stays there (correctly) when I kill the X11 app.
I didn't try with a pure console app.... but I can do if you need.
BTW, I think it's not a bad idea to have a 'xterm' toggle on run options, sometimes it's better to have a separate console than the ide one...

Max

Subject: Re: The very strange xterm issue
Posted by mirek on Tue, 18 Mar 2008 09:18:50 GMT

mdelfede wrote on Tue, 18 March 2008 04:27
Well, usually I like to have a console even on X11 apps, so I can see error and log messages, so I put console=true on Host.cpp and I got a running xterm that launches the X11 gui app.
All goes ok, no error and no crashes, and the xterm stays there (correctly) when I kill the X11 app.
I didn't try with a pure console app.... but I can do if you need.

Please do. I believe that could be the problem (because I do not see anything else...).

Quote:
BTW, I think it's not a bad idea to have a 'xterm' toggle on run options, sometimes it's better to have a separate console than the ide one...

Definitely. That is why I am trying this

Mirek

Subject: Re: The very strange xterm issue
Posted by mdelfede on Tue, 18 Mar 2008 11:53:52 GMT
View Forum Message <> Reply to Message

Well, strange enough, but you're right.... console app gives X problems.
My app is :

#include <Core/Core.h>

using namespace Upp;

CONSOLE_APP_MAIN
{
 String s;

 Cout() << "Enter a string:";
 s = ReadStdIn();
 Cout() << "\nyour string is: " << s;

Run conditions are following :
1-I run theide-svn (from latest svn build)
2-I load theide devel, patch the console stuff (to enable it)
3-I run theide patched from theide svn
4-From inside theide patched, I load and run my console app

Behaviour is :
1) the xterm is correctly opened
2) the app is correctly run inside xterm
3) on app termination, the xterm stays there (no error, ok)
4) when I close xterm all seems ok
5) BUT, if I try for example to do a text copy from inside the patched ide (edit/copy) it gives some weird X11 errors (once property change error, then on a next try font error, and so on)

The weirdest stuff is the following :
1) I close both theide patched AND theide svn.
2) I relaunch theide-svn, all ok
3) From inside theide-svn, I launch theide patched, all ok
4) WITHOUT running the console app, if I try again some copy operation from text inside the patched ide, it gives the same X11 errors..... it seems it has some "stored behaviour" somewhere.

I looked inside processes, none there. So, I think is something inside /var directory related to X11 stuffs....

Max

EDIT : Sorry but it was a mistake... I rebooted my laptop, started theide (patched one) from command line, tryed an edit/copy from inside it and got the same X11 error, without previous console app run.
Now I'm thinking.... as my devel stuff is (besides the small console patch...) identical to svn one, why does it crash ? The only different stuf is that it's compiled in debug mode... maybe that's the real problem ?

Max

EDIT2 : Confirmed, the problem is NOT in console app, but in theide compiled in debug mode. Debug mode ide has many X11 weird crashes with, for example, copy/paste operations.
Besides of that, console app run fine in both....

Max