Subject: Write an app to start and kill another app periodically
Posted by jpderyck on Thu, 29 Apr 2010 15:33:24 GMT
View Forum Message <> Reply to Message

Hello,
Under Linux, I would like to write a small application which can check for a hung application periodically. If it is hung, it should kill the process and restart the application.

The application to be checked periodically was written in u++: it reads a data module acquisition connected by usb every 10 seconds. It runs for +/- 2 months and then it crash. I suppose the problem is with the usb driver or the usb module itself. I made the same program reading a different module connected by rs232 port with no problem.

I know that I have to use command like 'killall app'
How to start this kind of command from an u++ gui program ?

The best should be if the small checking app was a linux service. I saw a service app sample for windows, but does anybody have sample for linux ?

best regards
Jean-Paul

Subject: Re: Write an app to start and kill another app periodically
Posted by koldo on Thu, 29 Apr 2010 15:51:28 GMT
View Forum Message <> Reply to Message

jpderyck wrote on Thu, 29 April 2010 17:33Hello,
Under Linux, I would like to write a small application which can check for a hung application periodically. If it is hung, it should kill the process and restart the application.

The application to be checked periodically was written in u++: it reads a data module acquisition connected by usb every 10 seconds. It runs for +/- 2 months and then it crash. I suppose the problem is with the usb driver or the usb module itself. I made the same program reading a different module connected by rs232 port with no problem.

I know that I have to use command like 'killall app'
How to start this kind of command from an u++ gui program ?

The best should be if the small checking app was a linux service. I saw a service app sample for windows, but does anybody have sample for linux ?

best regards
Jean-Paul

Hello jpderyck

I am not an expert but this works.

You can use function int kill(pid_t pid, int sig);

pid is the id of the process to be killed.

There are these options from less to more aggresive:

```
#include <sys/types.h>
#include <signal.h>
#include <unistd.h>
...
 pid_t pid;
...
 kill(pid, SIGTSTP);
 kill(pid, SIGTERM);
 kill(pid, SIGHUP);
 kill(pid, SIGKILL);
```

You can put sleep() between kill() functions just to give the program a little opportunity to stop itself.

To get the pid you can use getpid() and put this value somewhere where the killer process can take it.

Subject: Re: Write an app to start and kill another app periodically
Posted by dolik.rce on Thu, 29 Apr 2010 17:18:13 GMT
View Forum Message <> Reply to Message

Hi jpderyck,

I'm not sure if you know what you really want You were talking about GUI app on one side and periodical checks and service on the other. That doesn't get quite together

If you want to have GUI, than what Koldo says is the good way for you to go. I just add other possibility, which doesn't require you to know process id: You can simply use the Sys function from U++ core: Sys("killall -SIGXYZ app"); You can substitute the "SIGXYZ" with any of the signal names Koldo mentioned above, or omit it totally to use default SIGTERM.

The other option is a "service". Actually on linux it is called daemon, but basically it a same thing That is just regular console application, and your system is set up to start or terminate it as necessary. This is usually done via initscripts. Some more advanced solutions (like upstart) also provide nice features like checking if the daemon runs, so it is restarted even when it terminates for some reason. Details depend on what distribution you use.

And there is also a third option. Don't write an U++ app at all. From what you said, I believe that all you need is a simple shell script #!/bin/sh
here should be some check to see if the app is running or if it hangs,
that would depend on how it hangs and how you can test it ...
killall "yourapp"
/path/to/yourapp
Then you can set up cron to launch it periodically, let's say every 5 minutes: echo $(crontab -l; echo '*\5 * * * * /path/to/the/script' For more details about cron and how to use it see man cron and man crontab.

There is probably many more options, but I think the last one is just the one you need. Also it uses just standard tools, so it can be used on almost any system.

Best regards,
Honza

Subject: Re: Write an app to start and kill another app periodically
Posted by jpderyck on Fri, 30 Apr 2010 08:32:36 GMT
View Forum Message <> Reply to Message

Hi Honza and Koldo
thanks for reply

Of course I know that a gui app can not be as a service
I was just asking me what sould be the best (and the easist thing to do)

the way I can detect that the application which realize data acquisition is hung is by checking if new data are added to the mysql database. A new data is written every 10 secs, if not, the program is not started or is hung.

I am not familiar with linux scripts (I am at first a win32 programmer) so I will continue to look in direction of a u++ application development (console or gui)

many thanks for help and advice

best regards from Belgium
Jean-Paul

Subject: Re: Write an app to start and kill another app periodically
Posted by dolik.rce on Fri, 30 Apr 2010 23:21:07 GMT
View Forum Message <> Reply to Message

Quote:Of course I know that a gui app can not be as a service
Well, to be fair, it can There is nothing that would prevent you from doing that, you would just

have to make sure either that the app doesn't start gui when run in daemon-mode or tell it on which display to run. Anyway, this is just off-topic note, don't take it seriously

Good luck,
Honza

Subject: Re: Write an app to start and kill another app periodically
Posted by jpderyck on Tue, 07 Sep 2010 13:16:29 GMT
View Forum Message <> Reply to Message

Hi,
Back to this project, I would like to ask help again,

from my gui app, I can now kill another app with the killall command : Sys("killall -SIGXYZ app");

but how to restart it from the same gui app without wait for end?

ex: Sys("./app");
start the app but wait for end

best regards
Jean-Paul

Subject: Re: Write an app to start and kill another app periodically
Posted by dolik.rce on Tue, 07 Sep 2010 13:55:00 GMT
View Forum Message <> Reply to Message

Hi Jean-Paul,

Have a look at LocalProcess. It allows for fine grained control over an app, but for your needs it should be enough to use simply
LocalProcess("myapp -parameters ...").Detach();

Best regards,
Honza

Subject: Re: Write an app to start and kill another app periodically
Posted by jpderyck on Tue, 07 Sep 2010 16:23:50 GMT
View Forum Message <> Reply to Message

thanks for answer,

my problem is now:
when I use the LocalProcess method and then I kill the created app by the killall method, how can I know if the app is still really running since the command ps -N continue to show the app with a status <defunct>, is there another way to list process except those with the <defunct> ?

another way I checked is to use the Kill() method of the LocalProcess but it runs only once, I mean if I do a Start() method again after a Kill(), the next Kill() method will do nothing ! I suppose the LocalProcess should be destroyed and renewed but how to do that?

best regards
Jean-Paul

Subject: Re: Write an app to start and kill another app periodically
Posted by dolik.rce on Tue, 07 Sep 2010 20:07:03 GMT
View Forum Message <> Reply to Message

jpderyck wrote on Tue, 07 September 2010 18:23another way I checked is to use the Kill() method of the LocalProcess but it runs only once, I mean if I do a Start() method again after a Kill(), the next Kill() method will do nothing ! I suppose the LocalProcess should be destroyed and renewed but how to do that?

Congratulations, you just found a bug I posted the fix in separate thread, so it gets some attention and gets into SVN faster.

As for the <defunct>: In other words, it is a zombie process. It is not really running, it just hangs there until it's parent (which is the controlling app in this case) is terminated. The cause of this is probably an imperfect design of the Detach() function. There is nothing bad about zombies (sounds funny), but if you want to the controlling app running continuously, there would be a lot of them and it's just not clean solution.

So for now, I would recommend you to stay with Start() and Kill() (after fixing it) if it works for you. There is probably many other solutions to your task, but I would need to know much more about your project to tell you something useful.

Honza

Subject: Re: Write an app to start and kill another app periodically
Posted by andrei_natanael on Tue, 07 Sep 2010 21:56:50 GMT
View Forum Message <> Reply to Message

Until LocalProcess get fixed you may use (Linux Only) system("/path/to/app &");
It will run your application and not wait for it to return.
For windows useLaunchWebBrowser("/path/to/app");

Subject: Re: Write an app to start and kill another app periodically
Posted by jpderyck on Wed, 08 Sep 2010 08:36:49 GMT
View Forum Message <> Reply to Message

thanks for bug fix

can you tell me if the Kill() method of the LocalProcess has the same effectiveness than the 'killall -SIGKILL' because the goal is to kill a hung application (I know the best thing should be to fix the application wich enter in hung state but for the moment I do not find the bug...)

best regards
Jean-Paul

Subject: Re: Write an app to start and kill another app periodically
Posted by jpderyck on Wed, 08 Sep 2010 08:56:07 GMT
View Forum Message <> Reply to Message

hello andrei_natanael

thanks for your suggestion:

system("/path/to/app &"); works fine but
Sys("/path/to/app &"); does not ?

finally the combination of

Sys("pidof app"); to know if app is running
Sys("killall -SIGKILL app"); to stop app and
system("/path/to/app &"); to restart the app

works fine without zombie process (<defunct>)
for dolik.rce:

I will check the other method with the LocalProcess:
IsRunning();
Kill();
Start();

with new release of upp

best regards
Jean-Paul

Subject: Re: Write an app to start and kill another app periodically

Posted by dolik.rce on Wed, 08 Sep 2010 09:25:27 GMT
View Forum Message <> Reply to Message

jpderyck wrote on Wed, 08 September 2010 10:36thanks for bug fix

can you tell me if the Kill() method of the LocalProcess has the same effectiveness than the 'killall -SIGKILL' because the goal is to kill a hung application (I know the best thing should be to fix the application wich enter in hung state but for the moment I do not find the bug...)

best regards
Jean-Paul

It uses SIGTERM to end the process. If I am not mistaken, if the process won't terminate the result will be the same as calling Detach(), so the LocalProcess is ready for next use. That would of course left your hanged up app running, until it exits whatever loop it stuck in and processes the SIGTERM.

Is there any way we could help you with the hanging app?

Honza

Subject: Re: Write an app to start and kill another app periodically
Posted by jpderyck on Wed, 08 Sep 2010 14:26:38 GMT
View Forum Message <> Reply to Message

Dear Honza

thanks for help,

the application that I need to monitor with a kind of watchdog communicates with a data acquisition module (LabJack U3) via usb port. It takes measures every 10 seconds and put results in a MySQL table. After running for a time between 1 month and 3 months, the application enter in hung state (the main window becomes blank) and the only thing to do is restart the program.

I do not know if the problem is in the usb reading part of the program or in the MySQL part.

In the 'main' I use two timers:
SetTimeCallback(-10000, callback(this, &MyApp::Timer1));
SetTimeCallback(-300000, callback(this, &MyApp::Timer2));

Timer1: do measure and store it to databases
Timer2: reduce size of one table to keep 7 days of data

maybe I could change this to :
mytimer.KillSet(10000, THISBACK(Timer1));
in the 'main'

and in the Timer1() function :
mytimer.Kill();
do the measures and others and at the end :
mytimer.KillSet(10000, THISBACK(Timer1));

but normally the code between stop and restart timer should not last more than 1 second and so I should never have re-entry problem but...

if you have any suggestion

best regards
Jean-Paul
