Subject: TheIDE zombie problem
Posted by mirek on Sat, 30 Dec 2006 08:36:45 GMT
View Forum Message <> Reply to Message

TheIDE leaves launched programs as zombies.

Yes, I am well aware that you have to call wait for cleanup. However, in TheIDE, that is not entirely possible, because we need GUI to continue running.

Therefore we need a "start&forget" way of launching programs.

Somewhere I have found a recipe to solve the problem using singnals; the code is at ide/Host.cpp starting at line 156. But it does not seem to work well.

Any ideas?

Mirek

Subject: Re: TheIDE zombie problem
Posted by lundman on Sat, 30 Dec 2006 10:04:01 GMT
View Forum Message <> Reply to Message

Well, you can start children you don't need to wait on, if you really do not need to know when they die. Just start a new process group (setsid()) before calling.

```
if (!fork()) {
  // as child
  close(fileno(STDIN));
  close(fileno(STDOUT));
  close(fileno(STDERR));
  setsid();
  execve(argv[0], argv, env);
  /* not reached */
}
```

Iwould assume that's fine for "execute", but if you want to run it in debugged, you may need/want the SIGCHLD to know when it dies? Or if you are using stdin/out, deal/ignore SIGPIPE.

I can certainly fix it, so it leaves no zombies, if that is all that is needed.

Subject: Re: TheIDE zombie problem
Posted by mirek on Sat, 30 Dec 2006 10:32:50 GMT

lundman wrote on Sat, 30 December 2006 05:04
Iwould assume that's fine for "execute", but if you want to run it in debugged, you may need/want the SIGCHLD to know when it dies? Or if you are using stdin/out, deal/ignore SIGPIPE.

For debugging that really is not problem, the wait is (ok, should be called there (it is another mode..)

Mirek

Subject: Re: TheIDE zombie problem
Posted by mirek on Sat, 30 Dec 2006 10:56:44 GMT

lundman wrote on Sat, 30 December 2006 05:04Well, you can start children you don't need to wait on, if you really do not need to know when they die. Just start a new process group (setsid()) before calling.

```
if (!fork()) {
  // as child
  close(fileno(STDIN));
  close(fileno(STDOUT));
  close(fileno(STDERR));
  setsid();
  execve(argv[0], argv, env);
  /* not reached */
}
```

Iwould assume that's fine for "execute", but if you want to run it in debugged, you may need/want the SIGCHLD to know when it dies? Or if you are using stdin/out, deal/ignore SIGPIPE.

I can certainly fix it, so it leaves no zombies, if that is all that is needed.

Well I am afraid that above solution is not correct - you have to wait for pid returned from fork in the parent process.

Mirek

Subject: Re: TheIDE zombie problem
Posted by lundman on Sat, 30 Dec 2006 11:10:21 GMT
View Forum Message <> Reply to Message

Code meant as pseudo example, not "as is". But I can take a look at it Monday, no need to do that on "my time"

Subject: Re: TheIDE zombie problem
Posted by mirek on Sat, 30 Dec 2006 12:50:06 GMT
View Forum Message <> Reply to Message

lundman wrote on Sat, 30 December 2006 06:10
Code meant as pseudo example, not "as is". But I can take a look at it Monday, no need to do that on "my time"

Fixed...