
Subject: Re: capturing stdout/err/in of subprocess
Posted by [lundman](#) on Sat, 18 Mar 2006 00:01:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

I must agree I am not quite following why you need to know if your pipe-child is listening or not. You'll know if the child has gone away (EPIPE). The whole line buffering and echo is a tty/terminal feature which you can disable with `ioctl()`, that is quite trivial.

But if you create pipes to your child you need not worry about linemode/echo as it is direct without the tty/terminal layer.

You can check the number of bytes buffered in the pipe, set the low and high water marks, and maybe even the process flags. But why?

In LiON, to make sure to be portable, and allow for the pipe to be in `select()` non-blocking loops (HANDLES can't be in `select()`, and nonblocking polling is hard if you want to work on Win95/98/Me), I had to create read/write threads to translate between HANDLE and SOCKET.

But it works well, the same code will spawn processes on Unix, as Windows, and talk to it. I had to implement `fork()` and `execv()` for Windows though.
