Subject: Re: capturing stdout/err/in of subprocess
Posted by mr_ped on Fri, 17 Mar 2006 09:52:57 GMT
View Forum Message <> Reply to Message

I don't see problem with knowing if the subprocess is looking for input.

If it does, it must call "getchar()" and similar things, i.e. listen to the standart input. If you (as an console) own the standart input line, you will know when the subprocess listens to it, so you know when to echo the input.

I.e.
OS input -> your console listen to it, and fetch the input into internal buffer, which will be standart input for subprocess -> subprocess connected to input from your console.

I don't know how this should be done in C++ and Windows, but that's imho the proper model how it should be done.

The cmd.exe allows you to choose the input for the subprocess, so you can run the subprocess and feed him with input from file or from output from other process trough pipe, and the reason why it works is because it knows when the subprocess calls getchar(), so the whole file is not feeded to the subprocess right at the start, but it's sended char by char whenever subprocess asks for next one.
(actually having ability to redirect input/output is very important in unix world, as most of the standart tools are highly specialized, and to fullfill your task you need usually to chain several tools to produce the final result, like "ps | grep gcc" to see only gcc processes. So the "ps" tool does not need to contain some "filter" code, as the filter code is included in grep, etc...