

---

Subject: Re: Sound in linux

Posted by [dolik.rce](#) on Sun, 21 Dec 2014 20:04:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

rainbowsally wrote on Fri, 19 December 2014 14:45Hi Honza.

Quote:

There is no need for an invisible window. Just remember the pid returned from fork and call kill on it after the timeout is exceeded. In GUI apps, you can do this easily using SetTimeCallback().

Technically, you're right, but getting GUIs and the old terminal based linuxes is not so straight forward.

In U++ there is even ready to use solution for non-GUI apps. See bazaar/Timer. Or you could write your own, based on thread, or POSIX alarms, or any from many other suitable technologies.

rainbowsally wrote on Fri, 19 December 2014 14:45For example how could you get the PID without "waitpid()" or reading a return string from a system() call.

The calling app would hang so you'd never get a chance to shut it down.

The fork() call in your previous example returns the pid of child process to the parent. There is no need to block on the waitpid, it should only be called to collect the finished processes. If you don't want it to block your application, you can use it with the NOHANG option.

rainbowsally wrote on Fri, 19 December 2014 14:45A system call to an intermediate shell utility similar to kshell<N> is a much better alternative for a simple dialog bell of some sort.

Either it runs or it doesn't.

Calling shell just to run a command is in most cases unnecessary overkill. The shell actually does pretty much the same thing as described above.

Honza

---