

Re: New library: libpidfile.

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2005-08/0218.html>

From: Pawel Jakub Dawidek (*pjd_at_FreeBSD.org*)

Date: 08/24/05

Date: Wed, 24 Aug 2005 01:18:48 +0200

To: Peter Jeremy <PeterJeremy@optushome.com.au>

On Tue, Aug 23, 2005 at 07:42:26PM +1000, Peter Jeremy wrote:

+> On Tue, 2005-Aug-23 10:17:58 +0200, Pawel Jakub Dawidek wrote:
+> >In my first concept (when it was part of libutil) I allocated memory
+> >to store needed informations, because I didn't wanted to use preallocated
+> >memory (someone linking libutil doesn't have to use pidfile
+> >functionality).
+>
+> Since there's an initialisation function, why not just malloc whatever
+> memory you need? You can either store the pointer in a local static
+> or have pidfile_open() return it as an opaque pointer that the user
+> has to pass into other pidfile_XXX functions.

I added a handle, so now it goes like this:

```
struct pidfh *pfh;
pid_t otherpid, childpid;

pfh = pidfile_open("/var/run/daemon.pid", 0644, &otherpid);
if (pfh == NULL) {
    if (errno == EEXIST)
        errx(EXIT_FAILURE, "Daemon already running, pid: %d.", otherpid);
    /* If we cannot create pidfile from other reasons, only warn. */
    warn("Cannot open or create pidfile");
}

if (daemon(0, 0) == -1) {
    warn("Cannot daemonize");
    pidfile_remove(pfh);
    exit(EXIT_FAILURE);
}

pidfile_write(pfh);

for (;;) {
    /* Do work. */
    childpid = fork();
```

freebsd-arch: Re: New library: libpidfile.

```
switch (childpid) {
case -1:
    syslog(LOG_ERR, "Cannot fork(): %s.", strerror(errno));
    break;
case 0:
    pidfile_close(pfh);
    /* Do child work. */
    break;
default:
    syslog(LOG_INFO, "Child %d started.", childpid);
    break;
}
}

pidfile_remove(pfh);
exit(EXIT_SUCCESS);
```

I can't use atexit(3) anymore to remove pidfile, but I like it this way better, anyway.

+> >It also now has 4 functions, which makes it a good candidate of small,
+> >nice, lightweight library:)
+>
+> IMHO, 4 functions is too small. I would prefer to have a smaller
+> number of larger libraries and think this belongs in an existing
+> library – eg libutil. Whilst this may form a nice lightweight
+> library, if everyone wrote small, lightweight libraries, linking
+> an application would require a mile-long command line.

I talked about this with few more guys and they agree it should be a part of libutil, so I moved it there.

+> Can I suggest two enhancements:
+> 1) Allow NULL to be passed to pidfile_open() to indicate that the
+> pidfile should be in /var/run/PROCESS_NAME.pid (and/or if the
+> name doesn't include any '/' then default to /var/run)

I implemented NULL behaviour, but I'm not sure about the second thing. I'd like to avoid it.

+> 2) Write a wrapper function that calls pidfile_open() and if it's
+> successful, exec's the rest of the command line.

There is similar wrapper: lockf(1), so I'll not duplicate functionality, but I changed daemon(8) to use pidfile(3).

Thanks for all suggestions.

--

Pawel Jakub Dawidek
pjd@FreeBSD.org
FreeBSD committer

<http://www.wheel.pl>
<http://www.FreeBSD.org>
Am I Evil? Yes, I Am!

Re: New library: libpidfile.

- application/pgp-signature attachment: stored