DragonFlyBSD - Bug #1616

hotplug notification

11/26/2009 11:27 PM - polachok

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

I think it could be nice to receive notification for creation and removal of device nodes in userland. So I ported hotplug(4) and hotplugd(8) from OpenBSD. I like the design, it's clean and simple: an event contains type (attach or detach), device class (disk, tty or generic) and name (as in /dev). Events are queued and can be read from /dev/hotplug one-by-one (hotplugd(8) does that and can be configured to execute commands on event arrival).

You can get code to review here:
http://gitweb.dragonflybsd.org/~polachok/dragonfly.git/hotplug2

History

#1 - 11/27/2009 07:24 AM - alexh

in devfs_alias_create(), you are notifying hotplug about the attach of the underlying device and not the alias. As such, if there was an alias such as serno/00001 for da0, you'd get two attach notices for da0 and none for serno/00001.
devfs links are no cdev_t, they are synthetic devfs nodes. The name that should be passed to hotplug at this point is the function parameter name_orig.

I don't know how you want to handle that, but I don't think synthesizing a cdev_t just to pass the name and/or type to hotplug is a good idea. You could pass in this case both the underlying cdev_t (which is what you are doing now, 'target') and the name of the link/alias.

Another option could be to pass devfs nodes to hotplug, which contain a lot more information than the cdev_t, but that causes a number of problems:
- you'd need to acquire a lock (and also release it) for every node you use
- devfs would not be well encapsulated anymore
- for links and aliases you would need to use a function to get the full path, as each node only stores its name; for this you would just iterate up to the root node across ->parent elements, and stashsing the nodes. Then, just walk through that stash in reverse order, putting together the path.

In any case, I'm more inclined to the solution of creating some
struct hotplug_device {
cdev_t dev;
char *name;
};
or similar. This will make it much easier to add new information/options in the future.

Cheers,
Alex Hornung

#2 - 11/27/2009 12:19 PM - TGEN

Alexander Polakov wrote:
> I think it could be nice to receive notification for creation and
> removal of device nodes in userland. So I ported hotplug(4) and
> hotplugd(8) from OpenBSD. I like the design, it's clean and simple:
> an event contains type (attach or detach), device class (disk, tty or
> generic) and name (as in /dev). Events are queued and can be read
> from /dev/hotplug one-by-one (hotplugd(8) does that and can be
> configured to execute commands on event arrival).
>

> You can get code to review here:
> http://gitweb.dragonflybsd.org/~polachok/dragonfly.git/hotplug2

I've always thought kqueue would be the ideal place to put events like
this, also allowing multiple programs to listen for device events. Just
a pointer though, I have no code to show :).

Cheers,
--
Thomas E. Spanjaard
tgen@netphreax.net
tgen@deepbone.net

#3 - 11/28/2009 04:05 AM - dillon

Looks nice and clean. No reason why the alias issue should delay
a commit too long, it might be a good idea to get the infrastructure
in now (or soon), and then deal with aliases as a second stage.

-Matt
Matthew Dillon
<dillon@backplane.com>

#4 - 12/02/2009 11:29 PM - polachok

Yo. Alias issue should be fixed in
http://gitweb.dragonflybsd.org/~polachok/dragonfly.git/commitdiff/e3bf5370354c3268d7e4cdd77dd31c384f8a7e10
I'm not sure if some assumptions are safe, alexh@ take a look, please.

#5 - 12/03/2009 07:08 AM - alexh

The alias issue looks good now.

I've noticed something else though: in the path from devfs_unlinkp or any
other for that matter, you call hotplug_devfs_* and from there
hotplug_put_event, which in turn acquires a lockmgr lock, meaning it can go to
sleep. I don't think this is a good idea, as all devfs functions you are
hooking hold the devfs_lock. You could put a process to sleep that is holding
a different lock.

I'm not sure how much of an issue this can be or how to best approach it. For
notifications as these, lwkt messages come to my mind, but there might be
another solution, or it might not even be that bad.

If anyone else can provide some insight on how to approach this, it would be
good.

Cheers,
Alex Hornung

#6 - 04/03/2010 04:09 PM - alexh

this is committed.