

DragonFlyBSD - Bug #1716

can't stop "bmake" (without args) in /usr/pkgsrc

04/06/2010 09:02 PM - ker2x

Status:	In Progress	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Userland	Estimated time:	0.00 hour
Target version:	4.2.x		
Description			
step to reproduce the problem :			
* Fresh install of 2.6.1			
* cd /usr			
* make pkgsrc-create			
* cd /usr/pkgsrc			
* bmake			
* press ^c (ctrl-c) after a few seconds			
* you get the shell prompt, but some process (look like "configure" stuff) keep running in the background and flood your terminal. doing a "killall bmake" (or killall -9 bmake) doesn't help.			

History

#1 - 04/06/2010 10:12 PM - elektretterr

On Wednesday 07 April 2010 07:02:27 laurent laborde (via DragonFly issue tracker) wrote:

> New submission from laurent laborde <kerdezixe@gmail.com>:

>

> step to reproduce the problem :

>

> * Fresh install of 2.6.1

> * cd /usr

> * make pkgsrc-create

> * cd /usr/pkgsrc

> * bmake

> * press ^c (ctrl-c) after a few seconds

> * you get the shell prompt, but some process (look like "configure" stuff)

> keep running in the background and flood your terminal. doing a "killall

> bmake" (or killall -9 bmake) doesn't help.

I can confirm this is an issue.

Petr

#2 - 04/07/2010 07:28 AM - ahuate.devel

Hi,

Well, yes. A lot of bmake process will be spawned and you will have a bad time to stop them all. What I use to do the to run a ton of 'killall -9 bmake' very fast until there are no processes left. Of course this is just a rude workaround, but it gives you the chance to avoid the reboot :P

Cheers,
Antonio Huete

2010/4/7 Petr Janda <elektretterr@exemail.com.au>:

> On Wednesday 07 April 2010 07:02:27 laurent laborde (via DragonFly issue tracker) wrote:

>> New submission from laurent laborde <kerdezixe@gmail.com>:

>>

>> step to reproduce the problem :

>>

>> * Fresh install of 2.6.1

```
>> * cd /usr
>> * make pkgsrc-create
>> * cd /usr/pkgsrc
>> * bmake
>> * press ^c (ctrl-c) after a few seconds
>> * you get the shell prompt, but some process (look like "configure" stuff)
>> * keep running in the background and flood your terminal. doing a "killall
>> * bmake" (or killall -9 bmake) doesn't help.
>
>
> I can confirm this is an issue.
>
> Petr
>
```

#3 - 04/07/2010 07:38 AM - dillon

The main problem with bmake is that it temporarily sets a signal handler to SIG_IGN before setting it to something else. This means that when a ton of bmake's are running hitting ^C will often be ignored by some of them. DragonFly is doing the right thing, it's bmake which has to be fixed.

-Matt

#4 - 04/07/2010 07:51 AM - swildner

Am 07.04.2010 09:37, schrieb Matthew Dillon:

```
> The main problem with bmake is that it temporarily sets a signal
> handler to SIG_IGN before setting it to something else. This
> means that when a ton of bmake's are running hitting ^C will often
> be ignored by some of them. DragonFly is doing the right thing, it's
> bmake which has to be fixed.
```

As far as I remember, our make(1) has the same bug. It can be triggered quite easily (at least here) by going to /usr/src, do 'make clean' and trying to ^C it. I just tried it.

Sascha

#5 - 04/07/2010 10:43 AM - steve

On Wed, 7 Apr 2010 09:26:48 +0200

Antonio Huete Jimenez <ahuete.devel@gmail.com> wrote:

```
> Hi,
>
> Well, yes. A lot of bmake process will be spawned and you will have a
> bad time to stop them all. What I use to do the to run a ton of
> 'killall -9 bmake' very fast until there are no processes left. Of
> course this is just a rude workaround, but it gives you the chance to
> avoid the reboot :P
```

```
while killall -9 bmake; do cat /dev/null; done
```

Should do the job and drop out when they're all gone.

#6 - 04/07/2010 04:30 PM - dillon

```
: while killall -9 bmake; do cat /dev/null; done
:
: Should do the job and drop out when they're all gone.
:
:--
:Steve O'Hara-Smith | Directable Mirror Arrays
```

This is primarily due to a bug in bmake. bmake temporarily SIG_IGN's various signals before installing their signal handler so if a lot of bmake's are running and you hit ^C, sometimes some don't get killed.

-Matt
Matthew Dillon
<dillon@backplane.com>

#7 - 04/07/2010 04:32 PM - dillon

:As far as I remember, our make(1) has the same bug. It can be triggered
:quite easily (at least here) by going to /usr/src, do 'make clean' and
:trying to ^C it. I just tried it.

:
:Sascha

There might be another issue with signal races and fork(). The only
two possible issues are programs ignoring ^C and a signal race in
fork() (where the signal gets caught during the fork() but the
child is then created and never gets the signal).

-Matt
Matthew Dillon
<dillon@backplane.com>

#8 - 04/07/2010 04:52 PM - swildner

Am 07.04.2010 18:29, schrieb Matthew Dillon:

- > There might be another issue with signal races and fork(). The only
- > two possible issues are programs ignoring ^C and a signal race in
- > fork() (where the signal gets caught during the fork() but the
- > child is then created and never gets the signal).

I've looked a bit at this with tuxillo and in our make's case it's -B
(compatMake == true) which exposes the buggy behavior.

compatMake is false when you use -j and true when you don't. As soon as
I run 'make -j 1 clean' it no longer fails.

Sascha

#9 - 01/19/2015 04:16 AM - tuxillo

- *Description updated*
- *Category set to Userland*
- *Status changed from New to In Progress*
- *Assignee deleted (0)*
- *Target version set to 4.2.x*

Hi all,

Many changes have gone in since this was originally reported.
Can somebody please reproduce this one and see if the issue still persists?

Cheers,
Antonio Huete