

---

Subject: Re: How to query mount propagation state?  
Posted by [Karel Zak](#) on Mon, 16 Apr 2007 21:07:39 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Mon, Apr 16, 2007 at 10:39:46AM -0700, Ram Pai wrote:

> This patch disambiguates multiple mount-instances of the same  
> filesystem (or part of the same filesystem), by introducing a new  
> interface /proc/mounts\_new. The interface has the following format.

^^^^^^^^^^^^^^

... odd name. What will be the name for a next generation?  
"/proc/mounts\_new\_new"? :-)

> 'cat /proc/mounts' shows the following:

> /dev/root /mnt ext2 rw 0 0

> /dev/root /tmp1 ext2 rw 0 0

>

> NOTE: The above mount entries, do not indicate that /tmp1 contains the same  
> directory tree as /var/tmp.

>

> But 'cat /proc/mounts\_new' shows us the following:

> 0x6200 /mnt /var ext2 rw 0 0

> 0x6200 /tmp1 /var/tmp ext2 rw 0 0

Can't you purely and simply add the fsid= option to /proc/mounts?

/dev/root /mnt ext2 rw,fsid=0x6200 0 0

/dev/root /mnt ext2 rw,fsid=0x6200 0 0

I think you can do it without a negative impact to userspace.

> This patch introduces a new proc interface that exposes all the propagation  
> trees within the namespace.

Good idea.

> It walks through each off the mounts in the namespace, and prints the following information.

>

> mount-id: a unique mount identifier

> dev-id : the unique device used to identify the device containing the filesystem

^^^

Why not major:minor?

> path-from-root: mount point of the mount from /

> path-from-root-of-its-sb: path from its own root dentry.

> propagation-flag: SHARED, SLAVE, UNBINDABLE, PRIVATE

> peer-mount-id: the mount-id of its peer mount (if this mount is shared)

> master-mount-id: the mount-id of its master mount (if this mount is slave)

> Example:  
> Here is a sample output of cat /proc/\$\$/mounts\_propagation  
>  
> 0xa917800 0x1 // PRIVATE  
> 0xa917200 0x6200 // PRIVATE  
> 0xa917180 0x3 /proc / PRIVATE  
> 0xa917f80 0xa /dev/pts / PRIVATE  
> 0xa917100 0x6210 /mnt / SHARED peer:0xa917100  
> 0xa917f00 0x6210 /tmp /1 SLAVE master:0xa917100  
> 0xa917900 0x6220 /mnt/2 / SHARED peer:0xa917900

Same thing (although the mounts\_propagation makes more sense than mount\_new from my point of view).

cat /proc/mounts (or /proc/\$\$/mounts)

/dev/root /mnt ext2 rw,mid=0xa917100,did=0x6210,prop=SHARED,peer=0xa917100

my \$0.02...

Karel

--

Karel Zak <kzak@redhat.com>

---

Containers mailing list  
Containers@lists.linux-foundation.org  
<https://lists.linux-foundation.org/mailman/listinfo/containers>

---