
Subject: Re: [PATCH 0/3] clone64() and unshare64() system calls
Posted by [Sukadev Bhattiprolu](#) on Thu, 10 Apr 2008 01:07:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

H. Peter Anvin [hpa@zytor.com] wrote:

> sukadev@us.ibm.com wrote:

>> This is a resend of the patch set Cedric had sent earlier. I ported
>> the patch set to 2.6.25-rc8-mm1 and tested on x86 and x86_64.

>> ---

>> We have run out of the 32 bits in clone_flags !

>> This patchset introduces 2 new system calls which support 64bit

>> clone-flags.

>> long sys_clone64(unsigned long flags_high, unsigned long flags_low,

>> unsigned long newsp);

>> long sys_unshare64(unsigned long flags_high, unsigned long
>> flags_low);

>> The current version of clone64() does not support CLONE_PARENT_SETTID and

>> CLONE_CHILD_CLEARPID because we would exceed the 6 registers limit of some

>> arches. It's possible to get around this limitation but we might not

>> need it as we already have clone()

>

> I really dislike this interface.

>

> If you're going to make it a 64-bit pass it in as a 64-bit number, instead

> of breaking it into two numbers.

Maybe I am missing your point. The glibc interface could take a 64bit
parameter, but don't we need to pass 32-bit values into the system call
on 32 bit systems ?

> Better yet, IMO, would be to pass a pointer to a structure like:

>

> struct shared {

> unsigned long nwords;

> unsigned long flags[];

> };

>

> ... which can be expanded indefinitely.

Yes, this was discussed before in the context of Pavel Emelyanov's patch

<http://lkml.org/lkml/2008/1/16/109>

along with sys_indirect(). While there was no consensus, it looked like
adding a new system call was better than open ended interfaces.

Containers mailing list

Containers@lists.linux-foundation.org

