Subject: Re: [RFC PATCH 0/5] net: socket bind to file descriptor introduced
Posted by ebiederm on Wed, 15 Aug 2012 21:25:42 GMT
View Forum Message <> Reply to Message

"H. Peter Anvin" <hpa@zytor.com> writes:

> On 08/15/2012 12:49 PM, Eric W. Biederman wrote:
>>
>> There is also the trick of getting a shorter directory name using
>> /proc/self/fd if you are threaded and can't change the directory.
>>
>> The obvious choices at this point are
>> - Teach bind and connect and af_unix sockets to take longer AF_UNIX
>>   socket path names.
>>
>> - introduce sockaddr_fd that can be applied to AF_UNIX sockets,
>>   and teach unix_bind and unix_connect how to deal with a second type of sockaddr.
>>   struct sockaddr_fd { short fd_family; short pad; int fd; };
>>
>> - introduce sockaddr_unix_at that takes a directory file descriptor
>>   as well as a unix path, and teach unix_bind and unix_connect to deal with a
>>   second sockaddr type.
>>   struct sockaddr_unix_at { short family; short pad; int dfd; char path[102]; }
>>   AF_UNIX_AT
>>
>> I don't know what the implications of for breaking connect up into 3
>> system calls and changing the semantics are and I would really rather
>> not have to think about it.
>>
>> But it certainly does not look to me like you introduce new systems
>> calls to do what you want.
>>
>
> How would you distinguish the new sockaddr types from the traditional
> one? New AF_?

Yeah. AF_FD or AF_UNIX_AT is what I was thinking. The way the code
falls out that should be compartively simple to implement.

recvmsg etc would give you sockaddr_un sockets when they come from the
kernel.

Eric