Subject: Re: [PATCH v6 02/10] ipc: "use key as id" functionality for resource get system ca
Posted by ebiederm on Mon, 15 Oct 2012 19:47:40 GMT
View Forum Message <> Reply to Message

ebiederm@xmission.com (Eric W. Biederman) writes:

> Stanislav Kinsbursky <skinsbursky@parallels.com> writes:
>
>> This patch introduces new IPC resource get request flag IPC_PRESET, which
>> should be interpreted as a request to try to allocate IPC slot with number,
>> starting from value resented by key. IOW, kernel will try
>> allocate new segment in specified slot.
>>
>> Note: if desired slot is not emply, then next free slot will be used.
>
> This way of handling things is pretty nasty.
>
> - You don't fail if the requested id is not available.
> - You don't allow assigning the key (which leads to the need to change
>   the key in later patches). Changing the creator uid and creator
>   gid and key is semantically ugly.
>
> It would be much cleaner if you could instead add IPC_PRESET and then
> extend the definition of the creation functions all by one argument.
>
> aka
> int msgget(key_t key, int msgflg, int id);
> int semget(key_t key, int nsems, int semflg, int id);
> int shmget(key_t key, size_t size, int shmflg, int id);
>
> Where the extra id argument is ignored unless IPC_PRESET is specified.
>
> Also msgget, semget, and shmget should fail if unregconized flags are
> passed in. That ipcget doesn't do that today is bizarre.

Hmm. Come to think of it I don't see why you need to set the id at all.
We are using an idr allocator which effectively offers the semantics
that the lowest available id will be allocated. The same semantics we
have for file descriptors.

So it should be possible at least for the first pass at
checkpoint/restart to implement the restoration of sysv ipc without
IPC_PRESET at all.

So IPC_PRESET should just be an optimization, not a necessary feature.

That makes all of your code go away except the message queue

peeking, which seems much less intrusive for the first pass.

Eric