Subject: Re: [PATCH v6 02/10] ipc: "use key as id" functionality for resource get system ca

Posted by Stanislav Kinsbursky on Tue, 16 Oct 2012 07:52:01 GMT

View Forum Message <> Reply to Message

- > 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.

It's a part of design. Consider IPC\_PRESET as an advice. It's up to user to check, does returned id suits it's needs.

- > 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.

>

Key assigning ability is implemented in 4-6'th patches of the series. Changing creator uid and creator gid with key can be dropped. The reason why I added this was to give more abilities to syscall caller.

- > 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.

>

This way suits my needs. The reason, why I didn't this that way was trying to affect user-space API as less, as possible.

So, if there will be more votes for adding one more argument to existent systems calls - I'll do it.

> Also msgget, semget, and shmget should fail if unregconized flags are

passed in. That ipcget doesn't do that today is bizarre.	
his looks like another issue, which can be fixed separately.	
Eric	
est regards,	
tanislav Kinsbursky	
	_