

---

Subject: [RFC] Posix timers improvements, required for CRIU project  
Posted by [Stanislav Kinsbursky](#) on Tue, 09 Oct 2012 15:18:23 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi.

We would like to make CRIU able to migrate posix timers.

Currently we require additional info, which have to be provided by kernel.

In particular, it's:

- 1) Timers id's.
- 2) Timer clock (gained by id).
- 3) Timer sigevent structure.
- 4) Timer current overrun status (not the last one).

We are not sure, how to implement such support properly.

Right now we are considering two approaches of posix timers interface update (for dump):

- 1) First one it to use /proc filesystem to provide posix timers data by task (something like /proc/<pid>/posix\_timers)
- 2) Second one is to add new system calls like:

```
int timer_getnext(int start_id); // return next timer id after start-id
int timer_getclock(timer_t id, cclock_t *clock); // return clock
int timer_getsigevent(timer_t id, struct sigevent *evp); // return sigevent
int timer_getoverrun_cur(timer_t id, int *overrun); // current overrun
```

Of course, both approaches can be combined.

Also, we require a syscall to set desired overrun (on restore):

```
int timer_setoverrun(timer_t id, int *overrun);
```

Moreover, we need to make posix timer id to be allocated per task (not globally) to make sure, that we can restore timer with desired id (desired id can be passed to timer\_create() in timerid variable).

Our current idea is to replace idr by hash table, where "per\_process\_struct\_signal + per\_process\_id" will be the key.

We would greatly appreciate any comments or suggestions.

Thanks.

--

Best regards,  
Stanislav Kinsbursky

---