

OsmoGSMTester - Feature #2512

osmo-gsm-tester: Convert event_loop.wait to support passive polling

09/14/2017 10:43 AM - pespin

Status:	Resolved	Start date:	09/14/2017
Priority:	Normal	Due date:	
Assignee:	osmo-gsm-tester	% Done:	100%
Category:			
Target version:			
Spec Reference:			
Description			
Right now, all polling in osmo-gsm-tester is done by sleeping small amounts of times (like 0.1 seconds), then calling poll function everywhere. Some of those poll functions are actually event-based (like glib main loop, or any fd/socket being used) and can be monitored with select() instead of using active polling. This way we avoid wasting resources and we avoid delaying signal-driven events.			
For objects not supporting passive polling, we should provide an API to create a timer object (man timerd_create ?) and just call select() on those.			
Related issues:			
Related to OsmoGSMTester - Feature #3234: osmo-gsm-tester: poll the python-sm...		New	05/04/2018

History

#1 - 09/19/2017 11:13 PM - neels

+1

#2 - 03/26/2018 12:21 PM - pespin

As far as I could see now, it seems the best idea will be to re-use some of the code in modem.py and move it to event_loop.py to have an EventLoop class and public global functions which use a global instance of that object.

This EventLoop will contain a glib main loop inside. It will call glib_main_loop.iterate() every time we want to poll. New methods will be added to be able to register fds for passive polling instead of only having active polling functions. Also we will add methods to add/remove timer callbacks. We can then fire active polling every 0.1 seconds by using the timer registered function, and we can use the same to implement the wait() API we already have.

The idea is to slowly move required active polling users (python-smpp fd, wait(), sleep(), etc.) to use passive polling/timers.

Also interesting to look at GSource to implement management of all the fds/poll functions:

<https://developer.gnome.org/glib/stable/glib-The-Main-Event-Loop.html>

<https://blogs.gnome.org/desrt/2012/05/09/glib-mainloop-sources-in-python-e-g-for-irclib/>

#3 - 03/28/2018 05:23 PM - pespin

- Status changed from New to Feedback

- % Done changed from 0 to 60

Patches re-implementing event_loop wait() and sleep() using glib loop have been submitted to Gerrit:

```
remote: https://gerrit.osmocom.org/7559 event_loop: Use glib as mainloop impl and move modem to use event_lo  
op
```

```
remote: https://gerrit.osmocom.org/7560 Drop event_loop global functions and import MainLoop
```

With this code, we remove part of busy polling and from here we can easily add a register_fd() and unregister_fd() APIs to event_loop to be able to use passive polling with any source using an FD. We can even implement GSource subclasses.

#4 - 05/04/2018 12:23 PM - pespin

- Related to Feature #3234: osmo-gsm-tester: poll the python-smplib socket using non-busy approach added

#5 - 05/04/2018 12:24 PM - pespin

- *Status changed from Feedback to Resolved*

- *% Done changed from 60 to 100*

A new more specific task [#3234](#) has been created to move users of event_loop which can potentially use FDs as event triggers.