libvpoll: create synthetic events for poll, select and friends

Renzo Davoli   Luca Bassi

FOSDEM 2024

Introduction

Many programs use poll/select system calls to wait for events that are triggered by file descriptor I/O events.

To write a library able to behave like a network stack or a device, it's possible to implement functions like `my_socket`, `my_accept`, `my_open` and `my_ioctl`, as drop-in replacement of the system call counterparts.

It's possible to use dynamic library magic to rename/divert the system call requests to use their virtual implementation...

Introduction

Many programs use poll/select system calls to wait for events that are triggered by file descriptor I/O events.

To write a library able to behave like a network stack or a device, it's possible to implement functions like `my_socket`, `my_accept`, `my_open` and `my_ioctl`, as drop-in replacement of the system call counterparts.

It's possible to use dynamic library magic to rename/divert the system call requests to use their virtual implementation...

...but this approach does not allow using `select`, `poll` and similar system calls to wait for events on a mix of real file descriptors and library ones.

libvpoll

`libvpoll` permits to define file descriptors whose I/O events can be generated at user level.

This permits to generate synthetic events for `poll`, `select`, `ppoll`, `pselect`, `epoll`, etc.

This approach allows mixing real file descriptors with others provided by libraries as parameters of `poll`/`select` system calls.

libvpoll API

The interface of `libvpoll` consists of three functions:

`int vpoll_create(uint32_t init_events, int flags);`
: Creates a *vpollfd*.

`int vpoll_ctl(int fd, int op, uint32_t events);`
: Changes the set of pending events reported by a *vpollfd*.

`int vpoll_close(int fd);`
: Closes the *vpollfd* file descriptor.

Implementation

`libvpoll` needs kernel support for a complete implementation of its features.

The `libvpoll` library can use two different supports:

- Kernel patch extending the `eventfd` system call
- Kernel module implementing a virtual device (`/dev/vpoll`)

A feature-limited emulation is provided as a fallback.

Extending the eventfd system call

`eventfd` is used in some research papers to notify network events.

It was chosen because of the affinity of the feature.

A Linux kernel patch to add a new tag for `eventfd(2)`: `EFD_VPOLL`.

Otherwise it would have been possible to add two specific new system calls: `vpollfd_create` and `vpollfd_ctl`.

Extending the eventfd system call

To create a file descriptor for I/O event generation:

```
int fd = eventfd(EPOLLOUT, EFD_VPOLL | EFD_CLOEXEC);
```

`read(2)` returns the current state of the pending events.
`write(2)` is an or-composition of a control command:

- `EFD_VPOLL_ADDEVENTS`
- `EFD_VPOLL_MODEVENTS`
- `EFD_VPOLL_DELEVENTS`

For example:

```
uint64_t req = EFD_VPOLL_ADDEVENTS | EPOLLIN | EPOLLPRI;
write(fd, &req, sizeof(req));
```

Implementing a virtual device

Implemented as a virtual device creating a kernel module that when loaded creates the device `/dev/vpoll`.

This adds support in unpatched kernels.

To create a file descriptor:

```
int fd = open("/dev/vpoll", O_RDWR | O_CLOEXEC);
```

To generate events with `ioctl`:

```
ioctl(fd, VPOLL_IO_ADDEVENTS, EPOLLIN | EPOLLPRI)
```

Usage in picoxnet

Picoxnet[1] is a user-level network stack implemented as a library for the Internet of Threads.

When a picoxnet socket is created, the returned file descriptor is a *vpollfd*. So the user of the library can directly the kernel system call for event I/O (e.g. `select`, `poll`, etc.) as if it were a normal file descriptor.

[1] `https://github.com/virtualsquare/picoxnet`

Conclusions

`libvpoll` provides an easy-to-use API to create file descriptor whose I/O event can be generated at user level.

`libvpoll` and the virtual device module is available in the Debian stable repo.

The kernel patch was proposed upstream in 2019[2], we want to improve and propose a newer version in the near future.

Source code: `https://github.com/rd235/libvpoll-eventfd`

[2] `https://lore.kernel.org/all/20190526142521.GA21842@cs.unibo.it/`

Thank you for your attention

Questions?

VirtualSquare: `https://wiki.virtualsquare.org`