

# Thread safety for epoll/libaio

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-02/msg10073.html>

---

- *From:* "Li, Peng" <[ringer9cs@xxxxxxxx](mailto:ringer9cs@xxxxxxxx)>
  - *Date:* Tue, 28 Feb 2006 15:36:11 -0500
- 

I apologize if I should not post this on LKML, but there seems to be some lack of documentation for using epoll/AIO with threads. Are these interfaces thread-safe? Can I use them safely in the following way:

```
Thread A: while(1) { io_getevents(); ... }  
// wait forever until an event occurs, then handles the event and loop
```

```
Thread B: while(1) { epoll_wait(); ... }  
// same as thread A
```

```
Thread C: ... io_submit(); ...
```

```
Thread D: ... epoll_ctl(); ....
```

Suppose thread B calls `epoll_wait` and blocks before thread D calls `epoll_ctl`. Is it safe to do so? Will thread B be notified for the event submitted by thread D? Thread A and C pose the same question for AIO.

I wrote a simple program to test these interfaces and they seem to work without problems, but I am not sure if it is really safe to do so in general. If all of them works, it seems easy to use `epoll` and AIO together as I can simply use another thread to harvest events from thread A and B and make it look like a unified event notification interface.

Peng

–

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to [majordomo@xxxxxxxxxxxxxxxx](mailto:majordomo@xxxxxxxxxxxxxxxx)

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>