

## Re: non-blocking io, EINTR

---

*Source:* <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/hackers/2008-02/msg00457.html>

---

- *From:* "Mark Linn" <[mark.linn@xxxxxxxx](mailto:mark.linn@xxxxxxxx)>
  - *Date:* Wed, 27 Feb 2008 10:43:13 -0800
- 

On Wed, Feb 27, 2008 at 5:25 AM, James Bailie <[jimmy@xxxxxxxxxxxxxxxxxxxx](mailto:jimmy@xxxxxxxxxxxxxxxxxxxx)> wrote:

Mark Linn wrote:

- > I am setting the O\_NONBLOCK flag on a socket file descriptor using fcntl,
- >
- > will a read() on the socket return EINTR when the process get a signal?

By default, read() will restart itself automatically, regardless of whether the socket is blocking or not, as long as there is data to be read in the socket receive buffer. You can change this behavior by calling sigaction(). For example, the code below will make SIGTERM interrupt system calls. They will return an error code, usually -1, with the global errno set to EINTR. If the socket is non-blocking and the socket receive buffer is empty, then read() will also return an error, but with errno set to EWOULDBLOCK.

```
#include <signal.h>

struct sigaction sigact;

sigact.sa_handler = sigterm_handler;
sigemptyset( &sigact.sa_mask );
sigact.sa_flags = 0;

if ( sigaction( SIGTERM, &sigact, NULL ) < 0 )
{
    perror( "sigaction()" );
    exit( 1 );
}
```

--

James Bailie <[jimmy@xxxxxxxxxxxxxxxxxxxx](mailto:jimmy@xxxxxxxxxxxxxxxxxxxx)>  
<http://www.mammothcheese.ca>

Thanks, Ed and James,

Re: non-blocking io, EINTR

Then why in the world the sample code in this acm paper would test  
EINTR in read and write?

link is here.

<http://delivery.acm.org/10.1145/1350000/1344155/9815.html?key1=1344155&key2=2950393021&coll=GUIDE&dl=>

---

freebsd-hackers@xxxxxxxxxxx mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-hackers>

To unsubscribe, send any mail to "freebsd-hackers-unsubscribe@xxxxxxxxxxx"