Unix Programming FAQ (v1.37)

Source: http://unix.derkeiler.com/Newsgroups/comp.unix.programmer/2003−08/0040.html

---

From: Andrew Gierth (*andrew_at_erlenstar.demon.co.uk*)
Date: 07/31/03

```
Date: 31 Jul 2003 19:40:02 +0100
```

Archive−Name: unix−faq/programmer/faq
Comp−unix−programmer−Archive−Name: faq
URL: http://www.erlenstar.demon.co.uk/unix/faq_toc.html
URL: http://www.whitefang.com/unix/faq_toc.html
Posting−Frequency: every 2 weeks

Last−Modified: 2000/09/01 06:34:57
Version: 1.37

==========

About this FAQ
****

$Id: rawfaq.texi,v 1.37 2000/09/01 06:34:57 andrew Exp $

This FAQ was originally begun by Patrick Horgan in May 1996; I took it over after it had been lying idle for several months. I've reorganised it a bit and added some stuff; I still regard it as `under development'.

Comments, suggestions, additions, corrections etc. should be sent to the maintainer at: <andrew@erlenstar.demon.co.uk>.

A hypertext version of this document is available on the WWW. The home site is located at `http://www.erlenstar.demon.co.uk/unix/faq_toc.html'. A US mirror site is available at `http://www.whitefang.com/unix/faq_toc.html'.

This document is available by FTP from the news.answers archives at rtfm.mit.edu and its many mirror sites worldwide. The official archive name is `unix−faq/programmer/faq'. Sites which also archive *.answers posts by group should also carry the file under the `comp.unix.programmer' directory.

Other sources of information are not listed here. You can find pointers to other FAQs, books, source code etc. in the regular [READ ME FIRST] posting that should appear weekly in comp.unix.programmer. Administrivia regarding newsgroup conduct, etc., are also found there; I want to reserve this document specifically for technical Q's and A's.

All contributions have been edited by the maintainer, therefore any errors or omissions are my responsibility rather than that of the contributor.

This FAQ is now maintained as Texinfo source; I'm generating a raw text version for Usenet using the `makeinfo' program, and an HTML version using `texi2html'.


```
    short revents; /* Events found are returned here. */
};
```

A lot like `select()', the return value if positive reflects how many descriptors were found to satisfy the events requested. A zero return value is returned if the timeout period is reached before any of the events specified have occured. A negative value should immediately be followed by a check of `errno', since it signifies an error.

If no events are found, `revents' is cleared, so there's no need for you to do this yourself.

The returned events are tested to contain the event.

Here's an example:

```
/* Poll on two descriptors for Normal data, or High priority data.
   If any found call function handle() with appropriate descriptor
   and priority. Don't timeout, only give up if error, or one of the
   descriptors hangs up. */

#include <stdlib.h>
#include <stdio.h>

#include <sys/types.h>
#include <stropts.h>
#include <poll.h>

#include <unistd.h>
#include <errno.h>
#include <string.h>

#define NORMAL_DATA 1
#define HIPRI_DATA 2

int poll_two_normal(int fd1,int fd2)
{
   struct pollfd poll_list[2];
   int retval;
```

```
poll_list[0].fd = fd1;
poll_list[1].fd = fd2;
poll_list[0].events = POLLIN|POLLPRI;
poll_list[1].events = POLLIN|POLLPRI;

while(1)
{
   retval = poll(poll_list,(unsigned long)2,−1);
   /* Retval will always be greater than 0 or −1 in this case.
      Since we're doing it while blocking */

   if(retval < 0)
   {
      fprintf(stderr,"Error while polling: %s\n",strerror(errno));
      return −1;
   }

   if(((poll_list[0].revents&POLLHUP) == POLLHUP) ||
     ((poll_list[0].revents&POLLERR) == POLLERR) ||
```