NAME

epoll_wait − wait for an I/O event on an **epoll** file descriptor

SYNOPSIS

#include <sys/epoll.h>

int epoll_wait(int epfd**, struct epoll_event *** events**, int** maxevents**, int** timeout)

DESCRIPTION

Wait for events on the **epoll** file descriptor *epfd* for a maximum time of *timeout* milliseconds. The memory area pointed to by *events* will contain the events that will be available for the caller. Up to *maxevents* are returned by **epoll_wait**(2). The *maxevents* parameter must be greater than zero. Specifying a *timeout* of −1 makes **epoll_wait**(2) wait indefinitely, while specifying a *timeout* equal to zero makes **epoll_wait**(2) to return immediately even if no events are available (return code equal to zero). The **struct epoll_event** is defined as :

```
typedef union epoll_data {
        void *ptr;
        int fd;
        __uint32_t u32;
        __uint64_t u64;
} epoll_data_t;

struct epoll_event {
        __uint32_t events;      /* Epoll events */
        epoll_data_t data;/* User data variable */
};
```

The *data* of each returned structure will contain the same data the user set with a **epoll_ctl**(2) *(*EPOLL_CTL_ADD*,*EPOLL_CTL_MOD*)* while the *events* member will contain the returned event bit field.

RETURN VALUE

When successful, **epoll_wait**(2) returns the number of file descriptors ready for the requested I/O, or zero if no file descriptor became ready during the requested *timeout* milliseconds. When an error occurs, **epoll_wait**(2) returns −1 and *errno* is set appropriately.

ERRORS

EBADF
epfd is not a valid file descriptor.

EINVAL
The supplied file descriptor, *epfd*, is not an **epoll** file descriptor, or the *maxevents* parameter is less than or equal to zero.

EFAULT
The memory area pointed to by *events* is not accessible with write permissions.

CONFORMING TO

epoll_wait(2) is a new API introduced in Linux kernel **2.5.44**. The interface should be finalized by Linux kernel **2.5.66**.

SEE ALSO

epoll(4) **epoll_create**(2) **epoll_ctl**(2)