## NAME

epoll_ctl − control interface for an **epoll** descriptor

## SYNOPSIS

**#include <sys/epoll.h>**

**int epoll_ctl(int** epfd**, int** op**, int** fd**, struct epoll_event** ***event)**

## DESCRIPTION

Control an **epoll** descriptor, *epfd*, by requesting the operation *op* be performed on the target file descriptor, *fd*. The *event* describes the object linked to the file descriptor *fd*. 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 *events* member is a bit set composed using the following available event types :

**EPOLLIN**
The associated file is available for **read**(2) operations.

**EPOLLOUT**
The associated file is available for **write**(2) operations.

**EPOLLPRI**
There is urgent data available for **read**(2) operations.

**EPOLLERR**
Error condition happened on the associated file descriptor.

**EPOLLHUP**
Hang up happened on the associated file descriptor.

**EPOLLET**
Sets the Edge Triggered behaviour for the associated file descriptor. The default behaviour for **epoll** is Level Triggered. See **epoll**(4) for more detailed informations about Edge and Level Triggered event distribution architectures.

**EPOLLONESHOT**
Sets the One-Shot behaviour for the associated file descriptor. It means that after an event is pulled out with **epoll_wait**(2) the associated file descriptor is internally disabled and no other events will be reported by the **epoll** interface. The user must call **epoll_ctl**(2) with **EPOLL_CTL_MOD** to re-enable the file descriptor with a new event mask.

The **epoll** interface supports all file descriptors that support **poll**(2). Valid values for the *op* parameter are :

**EPOLL_CTL_ADD**
Add the target file descriptor *fd* to the **epoll** descriptor *epfd* and associate the event *event* with the internal file linked to *fd*.

**EPOLL_CTL_MOD**
Change the event *event* associated to the target file descriptor *fd*.

**EPOLL_CTL_DEL**
Remove the target file descriptor *fd* from the **epoll** file descriptor, *epfd*.

## RETURN VALUE

When successful, **epoll_ctl**(2) returns zero. When an error occurs, **epoll_ctl**(2) returns −1 and *errno* is set appropriately.

## ERRORS

**EBADF**
The *epfd* file descriptor is not a valid file descriptor.

**EPERM**
The target file *fd* is not supported by **epoll**.

**EINVAL**
The supplied file descriptor, *epfd*, is not an **epoll** file descriptor, or the requested operation *op* is not supported by this interface.

**ENOMEM**
There was insufficient memory to handle the requested *op* control operation.

## CONFORMING TO

**epoll_ctl**(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_wait**(2)