

NAME

epoll_create, epoll_create1 – open an epoll file descriptor

LIBRARY

Standard C library (*libc*, *-lc*)

SYNOPSIS

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

```
int epoll_create(int size);
```

```
int epoll_create1(int flags);
```

DESCRIPTION

epoll_create() creates a new **epoll(7)** instance. Since Linux 2.6.8, the *size* argument is ignored, but must be greater than zero; see HISTORY.

epoll_create() returns a file descriptor referring to the new epoll instance. This file descriptor is used for all the subsequent calls to the **epoll** interface. When no longer required, the file descriptor returned by **epoll_create()** should be closed by using **close(2)**. When all file descriptors referring to an epoll instance have been closed, the kernel destroys the instance and releases the associated resources for reuse.

epoll_create1()

If *flags* is 0, then, other than the fact that the obsolete *size* argument is dropped, **epoll_create1()** is the same as **epoll_create()**. The following value can be included in *flags* to obtain different behavior:

EPOLL_CLOEXEC

Set the close-on-exec (**FD_CLOEXEC**) flag on the new file descriptor. See the description of the **O_CLOEXEC** flag in **open(2)** for reasons why this may be useful.

RETURN VALUE

On success, these system calls return a file descriptor (a nonnegative integer). On error, *-1* is returned, and *errno* is set to indicate the error.

ERRORS**EINVAL**

size is not positive.

EINVAL

(**epoll_create1()**) Invalid value specified in *flags*.

EMFILE

The per-process limit on the number of open file descriptors has been reached.

ENFILE

The system-wide limit on the total number of open files has been reached.

ENOMEM

There was insufficient memory to create the kernel object.

STANDARDS

Linux.

HISTORY**epoll_create()**

Linux 2.6, glibc 2.3.2.

epoll_create1()

Linux 2.6.27, glibc 2.9.

In the initial **epoll_create()** implementation, the *size* argument informed the kernel of the number of file descriptors that the caller expected to add to the **epoll** instance. The kernel used this information as a hint for the amount of space to initially allocate in internal data structures describing events. (If necessary, the kernel would allocate more space if the caller's usage exceeded the hint given in *size*.) Nowadays, this hint is no longer required (the kernel dynamically sizes the required data structures without needing the hint), but *size* must still be greater than zero, in order to ensure backward compatibility when new **epoll** applications

are run on older kernels.

Prior to Linux 2.6.29, a `/proc/sys/fs/epoll/max_user_instances` kernel parameter limited live epolls for each real user ID, and caused **epoll_create()** to fail with **EMFILE** on overrun.

SEE ALSO

close(2), **epoll_ctl(2)**, **epoll_wait(2)**, **ioctl_eventpoll(2)**, **epoll(7)**