

**NAME**

`pread`, `pwrite` - read from or write to a file descriptor at a given offset

**SYNOPSIS**

```
#include <unistd.h>
```

```
ssize_t pread(int fd, void *buf, size_t count, off_t offset);
```

```
ssize_t pwrite(int fd, const void *buf, size_t count, off_t offset);
```

Feature Test Macro Requirements for glibc (see [feature\\_test\\_macros\(7\)](#)):

```
pread(), pwrite():
```

```
  _XOPEN_SOURCE >= 500
```

```
  || /* Since glibc 2.12: */ _POSIX_C_SOURCE >= 200809L
```

**DESCRIPTION**

**pread()** reads up to *count* bytes from file descriptor *fd* at offset *offset* (from the start of the file) into the buffer starting at *buf*. The file offset is not changed.

**pwrite()** writes up to *count* bytes from the buffer starting at *buf* to the file descriptor *fd* at offset *offset*. The file offset is not changed.

The file referenced by *fd* must be capable of seeking.

**RETURN VALUE**

On success, **pread()** returns the number of bytes read (a return of zero indicates end of file) and **pwrite()** returns the number of bytes written.

Note that it is not an error for a successful call to transfer fewer bytes than requested (see [read\(2\)](#) and [write\(2\)](#)).

On error, -1 is returned and *errno* is set to indicate the cause of the error.

**ERRORS**

**pread()** can fail and set *errno* to any error specified for [read\(2\)](#) or [lseek\(2\)](#). **pwrite()** can fail and set *errno* to any error specified for [write\(2\)](#) or [lseek\(2\)](#).

**VERSIONS**

The **pread()** and **pwrite()** system calls were added to Linux in version 2.1.60; the entries in the i386 system call table were added in 2.1.69. C library support (including emulation using [lseek\(2\)](#) on older kernels without the system calls) was added in glibc 2.1.

**CONFORMING TO**

POSIX.1-2001, POSIX.1-2008.

**NOTES**

The **pread()** and **pwrite()** system calls are especially useful in multithreaded applications. They allow multiple threads to perform I/O on the same file descriptor without being affected by changes to the file offset by other threads.

**C library/kernel differences**

On Linux, the underlying system calls were renamed in kernel 2.6: **pread()** became **pread64()**, and **pwrite()** became **pwrite64()**. The system call numbers remained the same. The glibc **pread()** and **pwrite()** wrapper functions transparently deal with the change.

On some 32-bit architectures, the calling signature for these system calls differ, for the reasons described in [syscall\(2\)](#).

**BUGS**

POSIX requires that opening a file with the **O\_APPEND** flag should have no effect on the location at which **pwrite()** writes data. However, on Linux, if a file is opened with **O\_APPEND**, **pwrite()** appends data to the end of the file, regardless of the value of *offset*.

**SEE ALSO**

[lseek\(2\)](#), [read\(2\)](#), [readv\(2\)](#), [write\(2\)](#)

**COLOPHON**

This page is part of release 4.10 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <https://www.kernel.org/doc/man-pages/>.