



Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'lseek.3p' command

\$ man lseek.3p

LSEEK(3P) POSIX Programmer's Manual LSEEK(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

lseek ? move the read/write file offset

SYNOPSIS

```
#include <unistd.h>

off_t lseek(int fildes, off_t offset, int whence);
```

DESCRIPTION

The lseek() function shall set the file offset for the open file description associated with the file descriptor fildes, as follows:

- * If whence is SEEK_SET, the file offset shall be set to offset bytes.
- * If whence is SEEK_CUR, the file offset shall be set to its current location plus offset.
- * If whence is SEEK_END, the file offset shall be set to the size of the file plus offset.

The symbolic constants SEEK_SET, SEEK_CUR, and SEEK_END are defined in <unistd.h>.

The behavior of lseek() on devices which are incapable of seeking is

implementation-defined. The value of the file offset associated with such a device is undefined.

The `lseek()` function shall allow the file offset to be set beyond the end of the existing data in the file. If data is later written at this point, subsequent reads of data in the gap shall return bytes with the value 0 until data is actually written into the gap.

The `lseek()` function shall not, by itself, extend the size of a file.

If `fildev` refers to a shared memory object, the result of the `lseek()` function is unspecified.

If `fildev` refers to a typed memory object, the result of the `lseek()` function is unspecified.

RETURN VALUE

Upon successful completion, the resulting offset, as measured in bytes from the beginning of the file, shall be returned. Otherwise, -1 shall be returned, `errno` shall be set to indicate the error, and the file offset shall remain unchanged.

ERRORS

The `lseek()` function shall fail if:

EBADF The `fildev` argument is not an open file descriptor.

EINVAL The `whence` argument is not a proper value, or the resulting file offset would be negative for a regular file, block special file, or directory.

EOVERFLOW

The resulting file offset would be a value which cannot be represented correctly in an object of type `off_t`.

ESPIPE The `fildev` argument is associated with a pipe, FIFO, or socket.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

None.

RATIONALE

The ISO C standard includes the functions `fgetpos()` and `fsetpos()`,

which work on very large files by use of a special positioning type.

Although `lseek()` may position the file offset beyond the end of the file, this function does not itself extend the size of the file. While the only function in POSIX.1?2008 that may directly extend the size of the file is `write()`, `truncate()`, and `ftruncate()`, several functions originally derived from the ISO C standard, such as `fwrite()`, `fprintf()`, and so on, may do so (by causing calls on `write()`).

An invalid file offset that would cause `[EINVAL]` to be returned may be both implementation-defined and device-dependent (for example, memory may have few invalid values). A negative file offset may be valid for some devices in some implementations.

The POSIX.1?1990 standard did not specifically prohibit `lseek()` from returning a negative offset. Therefore, an application was required to clear `errno` prior to the call and check `errno` upon return to determine whether a return value of `(off_t)-1` is a negative offset or an indication of an error condition. The standard developers did not wish to require this action on the part of a conforming application, and chose to require that `errno` be set to `[EINVAL]` when the resulting file offset would be negative for a regular file, block special file, or directory.

FUTURE DIRECTIONS

None.

SEE ALSO

`open()`

The Base Definitions volume of POSIX.1?2017, `<sys_types.h>`, `<unistd.h>`

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online

at <http://www.opengroup.org/unix/online.html> .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.kernel.org/doc/man-pages/reporting_bugs.html .

IEEE/The Open Group

2017

LSEEK(3P)