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 'lseek64.3' command

$ man lseek64.3

LSEEK64(3) Linux Programmer's Manual LSEEK64(3)

NAME

lseek64 - reposition 64-bit read/write file offset

SYNOPSIS

```
#define _LARGEFILE64_SOURCE     /* See feature_test_macros(7) */
#include <sys/types.h>
#include <unistd.h>
off64_t lseek64(int fd, off64_t offset, int whence);
```

DESCRIPTION

The lseek() family of functions reposition the offset of the open file associated with the file descriptor fd to offset bytes relative to the start, current position, or end of the file, when whence has the value SEEK_SET, SEEK_CUR, or SEEK_END, respectively.

For more details, return value, and errors, see lseek(2).

Four interfaces are available: lseek(), lseek64(), llseek(), and _llseek().

lseek()

Prototype:

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

The C library's lseek() wrapper function uses the type off_t. This is a 32-bit signed type on 32-bit architectures, unless one compiles with

```
#define _FILE_OFFSET_BITS 64
```

in which case it is a 64-bit signed type.

lseek64()

Prototype:

off64_t lseek64(int fd, off64_t offset, int whence);

The lseek64() library function uses a 64-bit type even when off_t is a 32-bit type. Its prototype (and the type off64_t) is available only when one compiles with

#define _LARGEFILE64_SOURCE

The function lseek64() is available since glibc 2.1.

llseek()

Prototype:

loff_t llseek(int fd, loff_t offset, int whence);

The type loff_t is a 64-bit signed type. The llseek() library function is available in glibc and works without special defines. However, the glibc headers do not provide a prototype. Users should add the above prototype, or something equivalent, to their own source. When users complained about data loss caused by a miscompilation of e2fsck(8), glibc 2.1.3 added the link-time warning

"the `llseek? function may be dangerous; use `lseek64? instead."

This makes this function unusable if one desires a warning-free compi? lation.

Since glibc 2.28, this function symbol is no longer available to newly linked applications.

_llseek()

On 32-bit architectures, this is the system call that is used (by the C library wrapper functions) to implement all of the above functions.

The prototype is:

int _llseek(int fd, off_t offset_hi, off_t offset_lo,
loff_t *result, int whence);

For more details, see llseek(2).

64-bit systems don't need an _llseek() system call. Instead, they have an lseek(2) system call that supports 64-bit file offsets.

ATTRIBUTES

For an explanation of the terms used in this section, see at?

tributes(7).

??????????????????????????????????????

?Interface ? Attribute ? Value ?

??????????????????????????????????????

?lseek64() ? Thread safety ? MT-Safe ?

??????????????????????????????????????

NOTES

lseek64() is one of the functions that was specified in the Large File Summit (LFS) specification that was completed in 1996. The purpose of the specification was to provide transitional support that allowed ap? plications on 32-bit systems to access files whose size exceeds that which can be represented with a 32-bit off_t type. As noted above, this symbol is exposed by header files if the _LARGEFILE64_SOURCE fea? ture test macro is defined. ALternatively, on a 32-bit system, the symbol lseek is aliased to lseek64 if the macro _FILE_OFFSET_BITS is defined with the value 64.

SEE ALSO

llseek(2), lseek(2)

COLOPHON

This page is part of release 5.10 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/.