NAME
      readahead - initiate file readahead into page cache

SYNOPSIS
      #define _GNU_SOURCE /* See feature_test_macros(7)
      */"
      #include <fcntl.h>

      ssize_t readahead(int *fd***, off64_t** *offset***, size_t** *count***);**

DESCRIPTION
      readahead() initiates readahead on a file so that subsequent reads from that file will be satisfied from the
      cache, and not block on disk I/O (assuming the readahead was initiated early enough and that other activity
      on the system did not in the meantime flush pages from the cache).

      The *fd* argument is a file descriptor identifying the file which is to be read. The *offset* argument specifies
      the starting point from which data is to be read and *count* specifies the number of bytes to be read. I/O is
      performed in whole pages, so that *offset* is effectively rounded down to a page boundary and bytes are read
      up to the next page boundary greater than or equal to *(offset+count)*. **readahead**() does not read beyond
      the end of the file. The file offset of the open file description referred to by *fd* is left unchanged.

RETURN VALUE
      On success, **readahead**() returns 0; on failure, -1 is returned, with *errno* set to indicate the cause of the
      error.

ERRORS
      EBADF
            fd is not a valid file descriptor or is not open for reading.

      EINVAL
            fd does not refer to a file type to which **readahead**() can be applied.

VERSIONS
      The **readahead**() system call appeared in Linux 2.4.13; glibc support has been provided since version 2.3.

CONFORMING TO
      The **readahead**() system call is Linux-specific, and its use should be avoided in portable applications.

NOTES
      On some 32-bit architectures, the calling signature for this system call differs, for the reasons described in
      syscall(2).

BUGS

      readahead() attempts to schedule the reads in the background and return immediately. However, it may
      block while it reads the filesystem metadata needed to locate the requested blocks. This occurs frequently
      with ext[234] on large files using indirect blocks instead of extents, giving the appearance that the call
      blocks until the requested data has been read.

SEE ALSO
      lseek(2), madvise(2), mmap(2), posix_fadvise(2), read(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/.