NAME
        recno - record number database access method

SYNOPSIS
        #include <sys/types.h>
        #include <db.h>

DESCRIPTION
        Note well: This page documents interfaces provided in glibc up until version 2.1. Since version 2.2, glibc
        no longer provides these interfaces. Probably, you are looking for the APIs provided by the *libdb* library
        instead.

        The routine dbopen(3) is the library interface to database files. One of the supported file formats is record
        number files. The general description of the database access methods is in dbopen(3), this manual page
        describes only the recno-specific information.

        The record number data structure is either variable or fixed-length records stored in a flat-file format,
        accessed by the logical record number. The existence of record number five implies the existence of
        records one through four, and the deletion of record number one causes record number five to be renum-
        bered to record number four, as well as the cursor, if positioned after record number one, to shift down one
        record.

        The recno access-method-specific data structure provided to dbopen(3) is defined in the *<db.h>* include file
        as follows:

                typedef struct {
                unsigned long flags;
                unsigned int cachesize;
                unsigned int psize;
                int lorder;
                size_t reclen;
                unsigned char bval;
                char *bfname;
                } RECNOINFO;

        The elements of this structure are defined as follows:

        flags   The flag value is specified by ORing any of the following values:

                R_FIXEDLEN
                        The records are fixed-length, not byte delimited. The structure element *reclen* specifies
                        the length of the record, and the structure element *bval* is used as the pad character. Any
                        records, inserted into the database, that are less than *reclen* bytes long are automatically
                        padded.

                R_NOKEY
                        In the interface specified by dbopen(3), the sequential record retrieval fills in both the
                        caller's key and data structures. If the **R_NOKEY** flag is specified, the *cursor* routines
                        are not required to fill in the key structure. This permits applications to retrieve records at
                        the end of files without reading all of the intervening records.

                R_SNAPSHOT
                        This flag requires that a snapshot of the file be taken when dbopen(3) is called, instead of
                        permitting any unmodified records to be read from the original file.

        cachesize
                        A suggested maximum size, in bytes, of the memory cache. This value is **only** advisory, and the
                        access method will allocate more memory rather than fail. If *cachesize* is 0 (no size is specified), a
                        default cache is used.

        psize   The recno access method stores the in-memory copies of its records in a btree. This value is the
                        size (in bytes) of the pages used for nodes in that tree. If *psize* is 0 (no page size is specified), a

page size is chosen based on the underlying filesystem I/O block size. See btree(3) for more information.

lorder   The byte order for integers in the stored database metadata. The number should represent the order as an integer; for example, big endian order would be the number 4,321. If *lorder* is 0 (no order is specified), the current host order is used.

reclen   The length of a fixed-length record.

bval   The delimiting byte to be used to mark the end of a record for variable-length records, and the pad character for fixed-length records. If no value is specified, newlines ("\n") are used to mark the end of variable-length records and fixed-length records are padded with spaces.

bfname   The recno access method stores the in-memory copies of its records in a btree. If *bfname* is non-NULL, it specifies the name of the btree file, as if specified as the filename for a dbopen(3) of a btree file.

The data part of the key/data pair used by the *recno* access method is the same as other access methods. The key is different. The *data* field of the key should be a pointer to a memory location of type *recno_t*, as defined in the *<db.h>* include file. This type is normally the largest unsigned integral type available to the implementation. The *size* field of the key should be the size of that type.

Because there can be no metadata associated with the underlying recno access method files, any changes made to the default values (e.g., fixed record length or byte separator value) must be explicitly specified each time the file is opened.

In the interface specified by dbopen(3), using the *put* interface to create a new record will cause the creation of multiple, empty records if the record number is more than one greater than the largest record currently in the database.

ERRORS

The *recno* access method routines may fail and set *errno* for any of the errors specified for the library routine dbopen(3) or the following:

EINVAL
    An attempt was made to add a record to a fixed-length database that was too large to fit.

BUGS

Only big and little endian byte order is supported.

SEE ALSO

btree(3), dbopen(3), hash(3), mpool(3)

Document Processing in a Relational Database System, Michael Stonebraker, Heidi Stettner, Joseph Kalash, Antonin Guttman, Nadene Lynn, Memorandum No. UCB/ERL M82/32, May 1982.

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/.