NAME

       SDBM_File - Tied access to sdbm files

SYNOPSIS

```
    use Fcntl; # For O_RDWR, O_CREAT, etc.
    use SDBM_File;

    tie(%h, 'SDBM_File', 'filename', O_RDWR|O_CREAT, 0666)
    or die "Couldn't tie SDBM file 'filename': $!; aborting";

    # Now read and change the hash
    $h{newkey} = newvalue;
    print $h{oldkey};
    ...

    untie %h;
```

DESCRIPTION

       `SDBM_File` establishes a connection between a Perl hash variable and a file in SDBM_File format. You
       can manipulate the data in the file just as if it were in a Perl hash, but when your program exits, the data
       will remain in the file, to be used the next time your program runs.

   Tie

       Use `SDBM_File` with the Perl built-in `tie` function to establish the connection between the variable and
       the file.

```
    tie %hash, 'SDBM_File', $basename, $modeflags, $perms;

    tie %hash, 'SDBM_File', $dirfile, $modeflags, $perms, $pagfilename;
```

       `$basename` is the base filename for the database. The database is two files with ".dir" and ".pag"
       extensions appended to `$basename`,

```
    $basename.dir (or .sdbm_dir on VMS, per DIRFEXT constant)
    $basename.pag
```

       The two filenames can also be given separately in full as `$dirfile` and `$pagfilename`. This suits for
       two files without ".dir" and ".pag" extensions, perhaps for example two files from File::Temp.

       `$modeflags` can be the following constants from the `Fcntl` module (in the style of the *open(2)* system
       call),

```
    O_RDONLY read-only access
    O_WRONLY write-only access
    O_RDWR read and write access
```

       If you want to create the file if it does not already exist then bitwise-OR (`|`) `O_CREAT` too. If you omit
       `O_CREAT` and the database does not already exist then the `tie` call will fail.

```
    O_CREAT create database if doesn't already exist
```

       `$perms` is the file permissions bits to use if new database files are created. This parameter is mandatory
       even when not creating a new database. The permissions will be reduced by the user's umask so the usual
       value here would be 0666, or if some very private data then 0600. (See "umask" in perlfunc.)

EXPORTS

       SDBM_File optionally exports the following constants:

*   `PAGFEXT` - the extension used for the page file, usually `.pag`.

*   `DIRFEXT` - the extension used for the directory file, `.dir` everywhere but VMS, where it is
        `.sdbm_dir`.

- • PAIRMAX - the maximum size of a stored hash entry, including the length of both the key and value.

These constants can also be used with fully qualified names, eg. SDBM_File::PAGFEXT

DIAGNOSTICS

On failure, the tie call returns an undefined value and probably sets $! to contain the reason the file could not be tied.

sdbm store returned -1, errno 22, key ``...'' at ...

This warning is emitted when you try to store a key or a value that is too long. It means that the change was not recorded in the database. See BUGS AND WARNINGS below.

BUGS AND WARNINGS

There are a number of limits on the size of the data that you can store in the SDBM file. The most important is that the length of a key, plus the length of its associated value, may not exceed 1008 bytes.

See ''tie'' in perlfunc(1), perldbmfilter(1), Fcntl