/\*

\* lookup2 : local file ; setup an in-memory index of words

\* and pointers into the file. resource is file name

\* use qsort & bsearch

\*/

#include <stdlib.h>

#include <string.h>

#include "dict.h"

typedef struct {

char word[WORD]; /\* The word to be looked up \*/

long off; /\* Offset into file for word definition \*/

} Index;

/\*

\* This ugly little function can be used by qsort & bsearch

\* It compares the word part of two structures of type Index

\*/

int dict\_cmp(const void \*a,const void \*b) {

return strcmp(((Index \*)a)->word,((Index \*)b)->word);

}

int lookup(Dictrec \* sought, const char \* resource) {

static Index \* table;

Dictrec dr;

static int numrec;

int i=0;

Index \* found,tmp;

static FILE \* in;

static int first\_time = 1;

if (first\_time) { /\* set up index \*/

first\_time = 0;

/\* Open file.

\*

\* Fill in code. \*/

if ((in = fopen(resource,"r")) == NULL){DIE(resource);}

/\* Get number records in file by dividing ending file offset by recsize.

\*

\* Fill in code. \*/

fseek(in, 0, SEEK\_END);

int j = ftell(in);

numrec = j/sizeof(Dictrec);

/\* Go to the beginning of the file.

\*

\* Fill in code. \*/

fseek(in, 0, SEEK\_SET); //or rewind(in);

/\* Allocate zeroed-out memory: numrec elements of struct Index. \*/

table = calloc(sizeof(Index),numrec);

/\* Read the file into the just-allocated array in memory.

\*

\* Fill in code. \*/

//while(fread(&dr,sizeof(Dictrec),1,in){

while(!feof(in)){

fread(&dr,sizeof(Dictrec),1,in);

if(feof(in)) break;

strcpy(table[i].word,dr.word);

table[i].off = i;

i++;

}

/\* Sort the table of entry/offset Index structures. \*/

qsort(table,numrec,sizeof(Index),dict\_cmp);

}

/\* end of first-time initialization \*/

/\* use bsearch to find word in the table; seek & read from file if found. \*/

strcpy(tmp.word,sought->word);

found = bsearch(&tmp,table,numrec,sizeof(Index),dict\_cmp);

/\* If found, go to that place in the file, and read the record into the

\* caller-supplied space. \*/

if (found) {

/\* Fill in code. \*/

fseek(in, sizeof(Dictrec)\*(found->off), SEEK\_SET);

fread(&dr,sizeof(Dictrec),1,in);

strcpy(sought->text,dr.text);

return FOUND;

}

return NOTFOUND;

fclose(in);

}