/\*

\* fifo\_server : listen on a named pipe; do lookup ; reply

\* down another named pipe, the name of which

\* will be sent by the client (in cli.id)

\* argv[1] is the name of the local fil

\* argv[2] is the name of the "well-known" FIFO

\*/

#include <sys/stat.h>

#include <fcntl.h>

#include <errno.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include "dict.h"

int main(int argc, char \*\*argv) {

//struct stat stbuff; this line use to file type, but this project use access replace

int read\_fd,write\_fd;

Client cli;

Dictrec tryit;

if (argc != 3) {

fprintf(stderr,"Usage : %s <dictionary source> ""<resource / FIFO>\n",argv[0]);

exit(errno);

}

/\* Check for existence of dictionary and FIFO (both must exist)

\*

\* Fill in code. \*/

if (access(argv[2], F\_OK) < 0){ //check FIFO existence

printf("does not exist %s",argv[2]);

exit(1);

}

if (access(argv[1], F\_OK) < 0){ // check dictionary existence

printf("does not exist %s",argv[1]);

exit(1);

}

/\* Open FIFO for reading (blocks until a client connects)

\*

\* Fill in code. \*/

if((read\_fd = open(argv[2],O\_RDONLY)) < 0){DIE("open");}

/\* Sit in a loop. lookup word sent and send reply down the given FIFO \*/

for (;;) {

/\* Read request.

\*

\* Fill in code. \*/

if( (read(read\_fd,&cli,sizeof(Client))) < 0) {DIE("read");}

/\* Get name of reply fifo and attempt to open it.

\*

\* Fill in code. \*/

if ((write\_fd = open(cli.id,O\_WRONLY)) < 0){DIE("open");}

/\* lookup the word , handling the different cases appropriately

\*

\* Fill in code. \*/

strcpy(tryit.word,cli.word);

switch(lookup(&tryit,argv[1]) ) {

case FOUND:

/\* Fill in code. \*/

if( write(write\_fd,&tryit,sizeof(Dictrec)) < 0){DIE("write");}

break;

case NOTFOUND:

memset(tryit.text,'\0',480);

strcpy(tryit.text,"XXXX");

if( write(write\_fd,&tryit,sizeof(Dictrec)) < 0){DIE("write");}

break;

case UNAVAIL:

/\* Fill in code. \*/

DIE(argv[1]);

}

/\* close connection to this client (server is stateless)

\*

\* Fill in code. \*/

close(write\_fd);

}

}