/\*

\* udp\_server : listen on a UDP socket ;reply immediately

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

\* PORT is defined in dict.h

\*/

#include <sys/types.h>

#include <sys/socket.h>

#include <string.h>

#include <netdb.h>

#include <netinet/in.h>

#include <errno.h>

#include "dict.h"

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

static struct sockaddr\_in server,client;

int sockfd,siz;

Dictrec dr, \*tryit = &dr;

int len =sizeof(struct sockaddr\_in);

if (argc != 2) {

fprintf(stderr,"Usage : %s <datafile>\n",argv[0]);

exit(errno);

}

/\* Create a UDP socket.

\* Fill in code. \*/

if((sockfd = socket (AF\_INET,SOCK\_DGRAM,0))==-1){

perror("socket");

exit(1);

}

/\* Initialize address.

\* Fill in code. \*/

server.sin\_family = AF\_INET;

server.sin\_addr.s\_addr = htonl(INADDR\_ANY);

server.sin\_port = htons(PORT);

/\* Name and activate the socket.

\* Fill in code. \*/

if(bind (sockfd, (struct sockaddr \*)&server,sizeof(server))<0){

perror("bind");

exit(1);

}

for (;;) { /\* await client packet; respond immediately \*/

siz = sizeof(client); /\* siz must be non-zero \*/

if(siz==0){

perror("error");

exit(1);

}

/\* Wait for a request.

\* Fill in code. \*/

while (recvfrom(sockfd,&dr,sizeof(Dictrec),0,(struct sockaddr \*)&client,&siz)) {

/\* Lookup request and respond to user. \*/

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

case FOUND:

/\* Send response.

\* Fill in code. \*/

sendto(sockfd,tryit,sizeof(Dictrec),0,(struct sockaddr \*)&client,len);

break;

case NOTFOUND :

/\* Send response.

\* Fill in code. \*/

strcpy(tryit->text,"XXXX");

sendto(sockfd,tryit,sizeof(Dictrec),0,(struct sockaddr \*)&client,len);

break;

case UNAVAIL:

DIE(argv[1]);

} /\* end lookup switch \*/

} /\* end while \*/

} /\* end forever loop \*/

} /\* end main \*/