/\*

\* lookup4 : This file does no looking up locally, but instead asks

\* a server for the answer. Communication is by message queues.

\* The protocol is : messages of type 1 are meant for the server.

\* The client only reads messages whose type matches their own pid.

\* The message queue key is what is passed as resource.

\*/

#include <sys/stat.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#include <string.h>

#include <stdlib.h>

#include <sys/types.h>

#include <unistd.h>

#include "dict.h"

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

static ClientMessage snd;

ServerMessage rcv;

static int qid;

static int pid;

static int first\_time = 1;

if (first\_time) { /\* open message queue \*/

first\_time = 0;

/\* Prepare our ClientMessage structure. \*/

pid = getpid();

sprintf(snd.content.id,"%d",pid);

snd.type = 1L;

/\* Open the message queue. Use resource pointer value as key.

\*

\* Fill in code. \*/

key\_t key;

if((key = strtol (resource,NULL,16))==0){DIE("strtol");}

if((qid = msgget( key, IPC\_CREAT | 0600)) == -1){DIE("msgget");}

}

/\* Send server the word to be found ; await reply

\*

\* Fill in code. \*/

/\*sender\*/

strcpy(snd.content.word,sought->word);

if(msgsnd(qid,&snd,sizeof(ClientMessage)-sizeof(long),0) == -1){DIE("msgsnd");}

/\*receive\*/

if(msgrcv( qid, &rcv, sizeof(ServerMessage)-sizeof(long),atol(snd.content.id), 0) == -1) {DIE("msgrcv");}

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

/\* Server returns XXXX when it cannot find request. \*/

if (strcmp(rcv.text,"XXXX") != 0) return FOUND;

return NOTFOUND;

}