/\*

\* lookup3 : This file does no looking up locally, but instead asks

\* a server for the answer. Communication is by named pipes.

\* The server sits listening on a "well-known" named pipe

\* (this is what is passed to lookup3 as resource)

\* The Client sets up a FIFO for reply, and bundles the FIFO

\* name with the word to be looked up. (See Client in dict.h)

\* Care must be taken to avoid deadly embrace (client & server

\* both waiting for something which can never happen)

\*/

#include <fcntl.h>

#include <string.h>

#include <stdlib.h>

#include <stdio.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <unistd.h>

#include "dict.h"

char \*tmpnam(char \*s);

static Client me;

void cleanup(void) {

/\* Delete named pipe from system.

\*

\* Fill in code. \*/

unlink(me.id);

}

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

static int write\_fd;

int read\_fd;

static int first\_time = 1;

if (first\_time) {

first\_time = 0;

/\* Open existing named pipe for client->server communication.

\*

\* Fill in code. \*/

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

/\* Create temporary named pipe for server->client communication. \*/

umask(0);

tmpnam(me.id);

/\* Fill in code. \*/

if (access("clientfifo", F\_OK) == -1) {

if ( (mkfifo("clientfifo", O\_CREAT|O\_EXCL|0755)) < 0){DIE("mkfifo");} //creat read fifo，0755為執行權限

}

/\* Register cleanup handler.

\*

\* Fill in code. \*/

cleanup();

}

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

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

/\* Fill in code. \*/

strcpy(me.id,"clientfifo");

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

/\* the temporary named pipe to get the answer from it.

\*

\* Fill in code. \*/

if ((read\_fd = open("clientfifo",O\_RDONLY)) < 0){DIE("open");}

/\* Get the answer.

\*

\* Fill in code. \*/

if( (read(read\_fd,sought,sizeof(Dictrec))) < 0 ){DIE("read");}

/\* Close the temporary named pipe as done.

\*

\* Fill in code. \*/

close(read\_fd);

/\* If word returned is not XXXX it was found. \*/

if (strcmp(sought->text,"XXXX") != 0){

return FOUND;

}

return NOTFOUND;

}