

```

/*
 *
 * Fichero fs_client.c modificado
 *
 */

#include "fs.h"
#include <stdio.h>
#include <stdlib.h> /* getenv, exit */

void fs_1(char *host, char * nombre_fichero)
{
    CLIENT *clnt;
    LEER_result *result_1;
    char *leer_1_nom;
    int leer_1_offs;
    int leer_1_nbytes;
    /*
    int *result_2;
    char *escribir_1_nom;
    int escribir_1_offs;
    int escribir_1_nbytes;
    BUF escribir_1_buf;
    int *result_3;
    char *crear_1_nom;
    int *result_4;
    char *borrar_1_nom;
    */
    #ifndef DEBUG
        clnt = clnt_create(host, FS, UNO, "netpath");
        if (clnt == (CLIENT *) NULL) {
            clnt_pcreateerror(host);
            exit(1);
        }
    #endif /* DEBUG */
    leer_1_nom = nombre_fichero;
    leer_1_offs=0;
    leer_1_nbytes=BUF_SIZE;

    result_1 = leer_1(leer_1_nom, leer_1_offs,
                     leer_1_nbytes, clnt);
    if (result_1 == (LEER_result *) NULL) {
        clnt_perror(clnt, "call failed");
    }

    if (-1 != result_1->cod_error)
    {
        int i;
        printf("\nSe han leído %d octetos:\n",
              result_1->datos.BUF_len);
        for(i=0;i<result_1->datos.BUF_len;i++)
            printf("%c",result_1->datos.BUF_val[i]);
    }

    /*
    result_2 = escribir_1(escribir_1_nom,
                          escribir_1_offs, escribir_1_nbytes,
                          escribir_1_buf, clnt);
    if (result_2 == (int *) NULL) {
        clnt_perror(clnt, "call failed");
    }
    result_3 = crear_1(crear_1_nom, clnt);
    if (result_3 == (int *) NULL) {
        clnt_perror(clnt, "call failed");
    }
    result_4 = borrar_1(borrar_1_nom, clnt);
    if (result_4 == (int *) NULL) {
        clnt_perror(clnt, "call failed");
    }
    */
    #ifndef DEBUG
        clnt_destroy(clnt);
    #endif /* DEBUG */
    }

int main(int argc, char *argv[])
{
    char *host;

    if (argc < 3) {
        printf("usage: %s server_host
nombre_fichero\n", argv[0]);
        exit(1);
    }
    host = argv[1];
    fs_1(host, argv[2]);

    return 0;
}

```

```

/*
 * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.
 */

#include "fs.h"
#include <stdio.h>
#include <stdlib.h> /* getenv, exit */

void fs_1(char *host)
{
    CLIENT *clnt;
    LEER_result *result_1;
    char *leer_1_nom;
    int leer_1_offs;
    int leer_1_nbytes;

    int *result_2;
    char *escribir_1_nom;
    int escribir_1_offs;
    int escribir_1_nbytes;
    BUF escribir_1_buf;
    int *result_3;
    char *crear_1_nom;
    int *result_4;
    char *borrar_1_nom;

    #ifndef DEBUG
        clnt = clnt_create(host, FS, UNO, "netpath");
        if (clnt == (CLIENT *) NULL) {
            clnt_pcreateerror(host);
            exit(1);
        }
    #endif /* DEBUG */

    result_1 = leer_1(leer_1_nom, leer_1_offs,
                     leer_1_nbytes, clnt);
    if (result_1 == (LEER_result *) NULL) {
        clnt_perror(clnt, "call failed");
    }

    result_2 = escribir_1(escribir_1_nom,
                          escribir_1_offs, escribir_1_nbytes,
                          escribir_1_buf, clnt);
    if (result_2 == (int *) NULL) {
        clnt_perror(clnt, "call failed");
    }
    result_3 = crear_1(crear_1_nom, clnt);
    if (result_3 == (int *) NULL) {
        clnt_perror(clnt, "call failed");
    }
    result_4 = borrar_1(borrar_1_nom, clnt);
    if (result_4 == (int *) NULL) {
        clnt_perror(clnt, "call failed");
    }

    #ifndef DEBUG
        clnt_destroy(clnt);
    #endif /* DEBUG */
    }

main(int argc, char *argv[])
{
    char *host;

    if (argc < 2) {
        printf("usage: %s server_host\n", argv[0]);

        exit(1);
    }
    host = argv[1];
    fs_1(host);
}

```