Tecnológico de Monterrey

Network Attached Storage in C with socket (server/client)

Prof. Gilberto Echeverria Furio

Enrique Lira Martinez A01023351
Emiliano Abascal Gurría A01023234
Cesar Armando Valladares Martinez A01023503

Description:

The program created is a network-attached storage making a file-level computer data storage server connected to a computer network providing data access to a certain group of clients. The NAS is specialized for serving files by its software so we are going to make a program in which you as a client can interact with the server and send file or data to be store in the server.

They typically provide access to files using network file sharing protocols. NAS devices began as a convenient method of sharing files among multiple computers.The benefits of dedicated network-attached storage, compared to general-purpose servers also serving files, include faster data access, easier administration, and simple configuration.

The program would require a log in for security reasons, when a client logs in, then it would have access to it's assigned directory, in it, the client will be able to create, delete or modify directories, and upload or download files into it's home directory or subdirectories created by the user.

Also the program have sharing functions so the user can decide if they want to share a certain directory with other users, with the permissions that the owner decides to give to the invited client (read, write, download or upload).

This would function over command line.

Functions:

```
char * Encrypt_password(char s[])
```

This function recives an original string (passwoord) and is responsible for encrypting ti at the time of registration, it is also used at the login, all stored passwords are encrypted.

```
int checkValidAccount(int account,char password[])
```

This is the function that is used at the login, it is responsible for checking that the user ID exists, and if it existsm compares of the encrypted passwords, if both conditions are true, the session is started without problems.

```
int SaveUser(thread_data_t* data ,char password[])
```

When registering a new user, a unique ID is created that is delivered to the user, as well as calling the function that encrypts the passwords and save the result.

```
int checkPermisess(thread_data_t* data ,int account_num,char
*other_account);
```

Check wich users have permissions

```
int AddPersmisses(thread_data_t * data ,int account_num, char
other_account [])
```

Since we have the option to share files, this function allows us to create a folder where there is a file with the ID's of the users that have the permissions to see the files of the original user.

```
int DeletePermisses(thread_data_t* data ,int account_num,char
other_account[])
```

Just as there is the option of giving permissions, they can also be revoked, providing the ID to which the permissions will be revoked, it will be deleted from the file.

```
void SaveFile(thread_data_t* data, int account_num,unsigned char *
message,int numbytes, char * fileName, char * path, int flag)
```

This function is responsible for taking the files uploaded by the user and saving them on the server.

```
void Delete(thread_data_t* data, int account_num)
```

This will erase the files that the user has on the server.

```
char * read_line_by_line(FILE *fin)
```

Returns a char pointer with the text of a file.

```
void printLocalIPs()
```

Prints the local IP.

```
int initServer(char * port, int max_queue)
```

Initialize the server in the given port.

```
int connectSocket(char * address, char * port)
```

Creates the connection to the server.

```
int recvString(int connection_fd, char * buffer, int size);
int recvFile(int connection_fd,  char * buffer, int size);
```

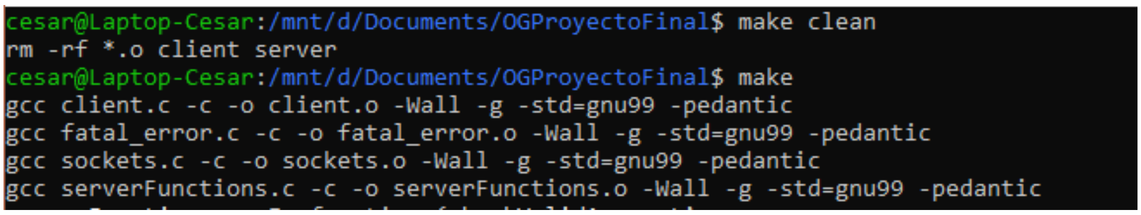
Functions that manages the buffer for communication between the server and the users.

```
bool saveFile(unsigned char * file, char * fileName, FILE * infile,
int numbytes)
```

Save file in the user's PC

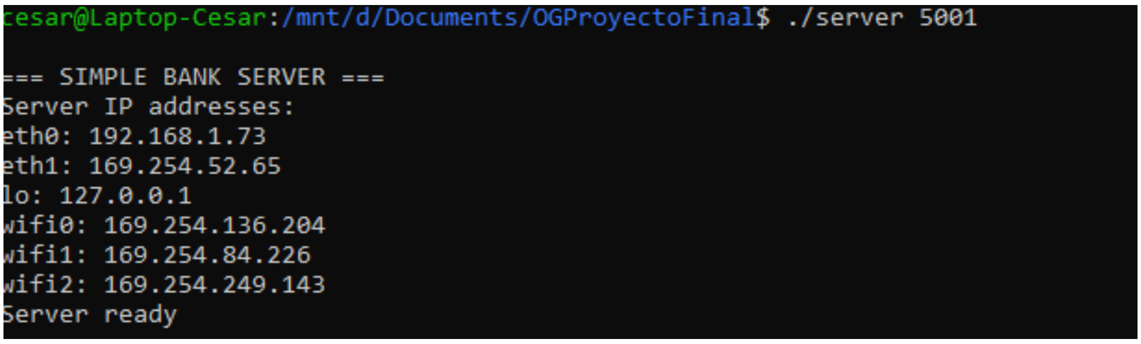
Manual:

once we have the folder with all the files, and in the terminal we are in this folder, just to make sure we type "make clean" to delete the .o files, then we will only type "make".



```
cesar@Laptop-Cesar:/mnt/d/Documents/OGProyectoFinal$ make clean
rm -rf *.o client server
cesar@Laptop-Cesar:/mnt/d/Documents/OGProyectoFinal$ make
gcc client.c -c -o client.o -Wall -g -std=gnu99 -pedantic
gcc fatal_error.c -c -o fatal_error.o -Wall -g -std=gnu99 -pedantic
gcc sockets.c -c -o sockets.o -Wall -g -std=gnu99 -pedantic
gcc serverFunctions.c -c -o serverFunctions.o -Wall -g -std=gnu99 -pedantic
```

Then we use “./server <port>” where we specity the port that will be use for the server, and it will show a message that the server is waiting for connections:

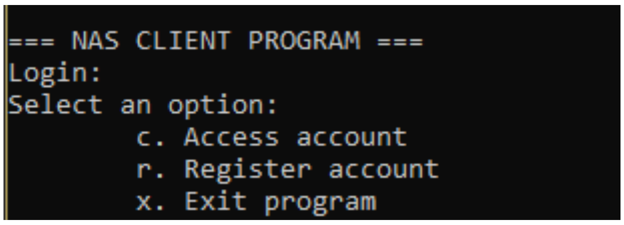


```
cesar@Laptop-Cesar:/mnt/d/Documents/OGProyectoFinal$ ./server 5001

=== SIMPLE BANK SERVER ===
Server IP addresses:
eth0: 192.168.1.73
eth1: 169.254.52.65
lo: 127.0.0.1
wifi0: 169.254.136.204
wifi1: 169.254.84.226
wifi2: 169.254.249.143
Server ready
```

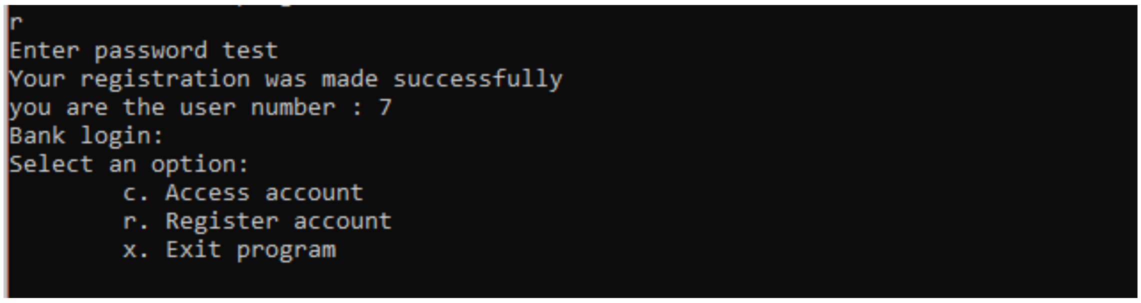
It is then when the user can already connect, in this case we will use another terminal and the address of localhost, but if it is done from another computer it is necessary to have the IP of the server.

We will see the menu of the client where 3 options will be shown: log in, register and exit.



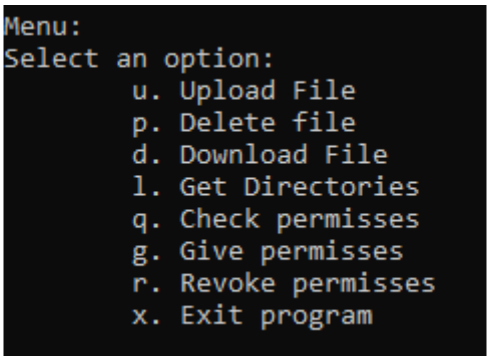
```
=== NAS CLIENT PROGRAM ===
Login:
Select an option:
        c. Access account
        r. Register account
        x. Exit program
```

To register an account first we type "r", and then provide a password, when the account has been created we'll receive the ID to login.



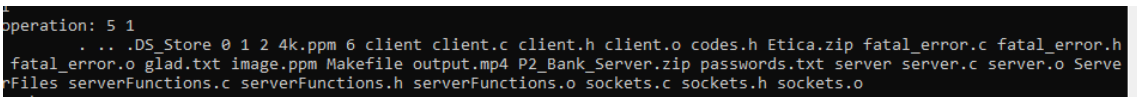
```
r
Enter password test
Your registration was made successfully
you are the user number : 7
Bank login:
Select an option:
        c. Access account
        r. Register account
        x. Exit program
```

We can now login, we type "c" and provide the ID we received, in this case 7, and the password "test".



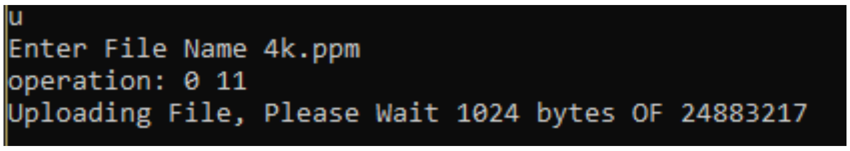
```
Menu:
Select an option:
        u. Upload File
        p. Delete file
        d. Download File
        l. Get Directories
        q. Check permisses
        g. Give permisses
        r. Revoke permisses
        x. Exit program
```

Fist with “l” we can see the files that are in the actual folder.



```
operation: 5 1
        . .. .DS_Store 0 1 2 4k.ppm 6 client client.c client.h client.o codes.h Etica.zip fatal_error.c fatal_error.h
 fatal_error.o glad.txt image.ppm Makefile output.mp4 P2_Bank_Server.zip passwords.txt server server.c server.o Serve
rFiles serverFunctions.c serverFunctions.h serverFunctions.o sockets.c sockets.h sockets.o
```

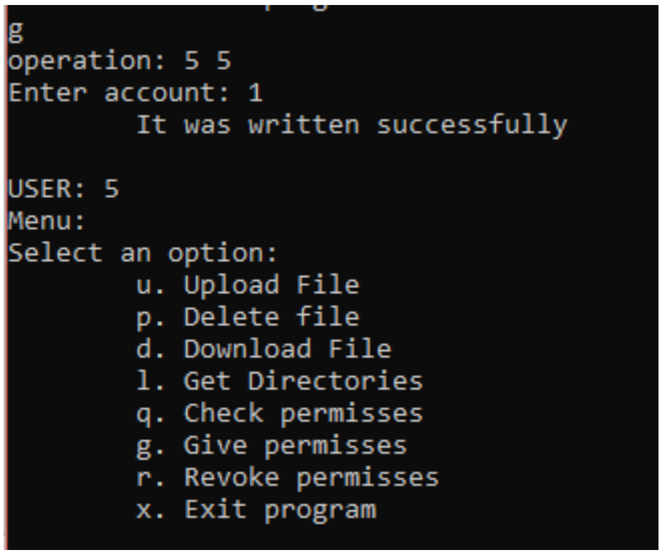
Then we have the option "u", to send the file to the server, you just have to type the name of the file, in this case, an image .ppm



```
u
Enter File Name 4k.ppm
operation: 0 11
Uploading File, Please Wait 1024 bytes OF 24883217
```

Type “d” to download an existing file and “p” to delete the file.

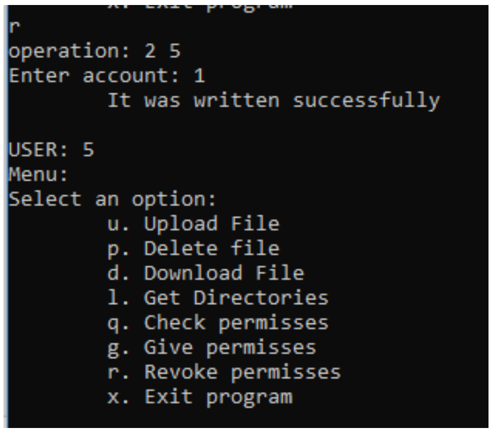
With the option “g” you give permissions to another account, you have to type the ID of the other account.



```
g
operation: 5 5
Enter account: 1
        It was written successfully

USER: 5
Menu:
Select an option:
        u. Upload File
        p. Delete file
        d. Download File
        l. Get Directories
        q. Check permisses
        g. Give permisses
        r. Revoke permisses
        x. Exit program
```

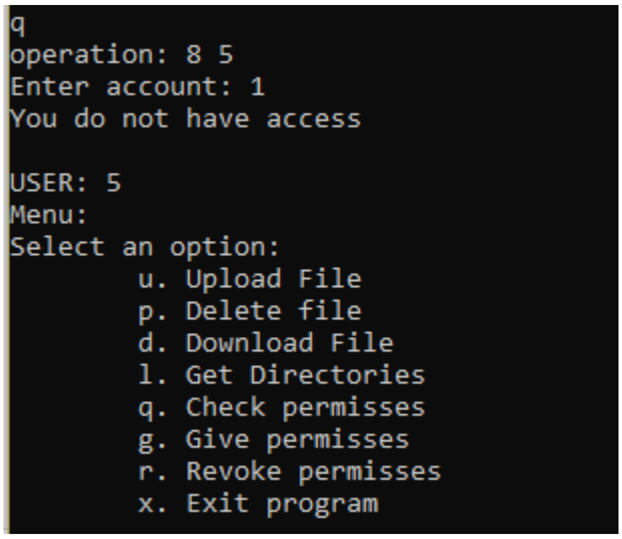
Press “r” to revoke permisses, then type the ID of the other account.



```
r
operation: 2 5
Enter account: 1
        It was written successfully

USER: 5
Menu:
Select an option:
        u. Upload File
        p. Delete file
        d. Download File
        l. Get Directories
        q. Check permisses
        g. Give permisses
        r. Revoke permisses
        x. Exit program
```

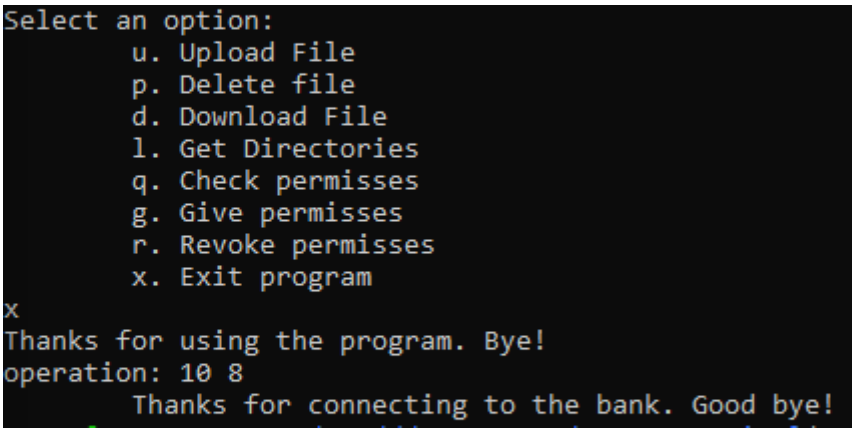
Type “q” t osee if you have permissions to see another files. it should say that it does not have permissions due it is a new account and no other user has added it.



```
q
operation: 8 5
Enter account: 1
You do not have access

USER: 5
Menu:
Select an option:
        u. Upload File
        p. Delete file
        d. Download File
        l. Get Directories
        q. Check permisses
        g. Give permisses
        r. Revoke permisses
        x. Exit program
```

Finally, by pressing "x" the client's account will be closed as well as the program.



```
Select an option:
        u. Upload File
        p. Delete file
        d. Download File
        l. Get Directories
        q. Check permisses
        g. Give permisses
        r. Revoke permisses
        x. Exit program
x
Thanks for using the program. Bye!
operation: 10 8
        Thanks for connecting to the bank. Good bye!
```