



PROJECT

Network Programming



Project Network PROGRAMMING WITH C

Project Server

This work is to be done in team in 4 students (same TD group). It must be returned no later than March 14, 2019 at 18h

- Any discount after this slot will be punished by a penalty of -5 points per day of delay.**
- You will deposit on Devinci Online a .zip file, in team, containing:**
 - the source code in C (.c and .h files)**
 - the mini-report in PDF of 5 pages minimum**

Note:

A render without the report will be noted 00/20.

A program with warnings at the compilation will be noted with a penalty

up to -5 points depending on the nature of the warnings.

A program that does not compile or run will be rated 00/20.

A cheat results in a score of 00/20 for the entire module.

Project Server

Mini-report

A mini-report of 5 pages minimum is to render with the code.

Must be included:

- the name and surname of the students who did the work;**
- An explanation in French or in English of how is coded each feature proposed.**
- A small diagrams complete the explanation.**

Project Network programming

Only code write in C

- Project Clients/server (Windows/Linux):

- 1) Possibility to connect Client/Server in UDP or TCP
- 2) Possibility to command line example (#exit) for exit client
- 3) Possibility Send and receive Messages (len max 280 characters by message)
- 4) Possibility to transfers files (client to client , client to server)
- 5) Possibility to Multi-Threads more clients(50) for a server
- 6) Possibility dual accept Client (Linux/Windows) and Possibility client running a different platform (Windows/Linux)
- 7) Server require a username and password for connect is ok (eventually use sqlite (SGBD SQL) for password management
- 8) Log (login , @IP, date, ...)
- 9) Your original feature

Bonus :

- 1) Possibility to connect security in port-knocking (5 ports minimum) ⁵

Project Network programming

Only code write in C

- Project Clients/server (Command Line):

- 1) #Exit
- 2) #Help (list Command)
- 3) #ListU (list user in a server)
- 4) #ListF (list files in a server)
- 5) #TrfU (transfert Upload file in a server)
- 6) #TrfD (transfert Download file in a server)
- 7) #Private <user> (commute to private)
- 8) #Public (commute to public)
- 9) #Ring <user> (notification if user is connect)
- 10) Your original Commands

TIPS 1

- Sometimes, an ungraceful exit from a program (e.g., ctrl-c) does not properly free up a port
- Eventually (after a few minutes), the port will be freed
- You can kill the process, or
- To reduce the likelihood of this problem, include the following code:
 - In header include:

```
#include <signal.h>
void cleanExit(){exit(0);}
```
 - In socket code:

```
signal(SIGTERM, cleanExit);
signal(SIGINT, cleanExit);
```

TIPS 2

- Check Beej's Guide to Network Programming Using Internet Sockets
- Search the specification for the function you need to use for more info, or check the man pages.

TIPS 3

- How to find the IP address of the machine my server program is running on?
 - Use 127.0.0.1 or localhost for test and accessing a server running on your local machine.
 - For a remote server running linux use the bash shell command: “\$ /sbin/ifconfig”
 - For windows, use ipconfig in cmd

REFERENCES

• These are good references for further study of Socket programming with C:

- Beej's Guide to Network Programming Using Internet Sockets
 - 2019S1_C5_DOC_BGNET_Socket Programming_EN.pdf
- 2019S1_C5_DOC_TCP_IP_Sockets_in_C:_Practical_Guide_for_Programmers_EN.pdf
- The GNU C Reference Manual
 - <https://www.gnu.org/software/gnu-c-manual/gnu-c-manual.html>
 - <https://www.gnu.org/software/libc/manual/pdf/libc.pdf>