BRNO
UNIVERSITY
OF TECHNOLOGY

FACULTY
OF INFORMATION
TECHNOLOGY

# Data Communications, Computer Networks and Protocols

## Short messaging exchange
## Client – Server based application

author : Dimitar Jordanov – xjorda01

## 1. Implementation manual

The communication between the server and the clients is based on Libnet library.

**The server:**

- Threads
  - The server creates three threads. As a communication channel (IPC ) between the threads is used Message Queue.
  - Command Line Interface thread – A simple command set is provided.
    - print – prints the user database
    - ? - Help
    - Enter – Quit
  - TCP Thread – Wait for TCP connections.
  - UDP Thread – Collects message from the message queue. Check the type of the message. If normal message – Send to all . If Sign in message – Add new user to the user's database. If Sign off message delete a user from the database.

- IPC - Massage queue
- Signal handling – All signals are blocked. If user press Ctrl + C an appropriate message is displayed.
- Command line interface (CLI)
- Maintains user's database – User's database is implemented as Linked list. Saves users IP, UDP port and PID.

**The client:**

- Threads
  - The client creates two threads. The first thread read the user's input and send it to the server via TCP connection. This thread is responsible for sing in and sign off messages as well.
  - The second thread receives messages from the server over UDP channel and print them.
  - Signal handling – All signals are blocked. If user press Ctrl + C an appropriate message is displayed.

### The test program:

The test program starts one server and two clients. As input is used the following

- file set:
- server.conf
- client_1.conf
- client_2.conf

The output for the clients is redirected to the following output files.

- client_1.out
- client_2.out

The output from the server is sent to stdout.

## 2.User's manual:

### The server :

example : ./messserver -tcp 1234 -udp 555

When you start the server you will be asked to enter an IP address that the server is to use. After the initial initialization the server will be ready to accept connections from the clients. All Sing in , Sing off evens and all messages are printed on the screen. Command “print” is implemented and can be used from the administrator. The command gives information about the currently singed in users.

The server commits exit if the administrator press Enter. All signals are blocked ( Ctrl +C, Ctrl + Z does not have any effect.)

### Client :

example : ./messclient <server's IP> -tcp 1234 -udp 555

When you start the server you will be asked to enter the IP that is to represent you in the communication precess with the server.

After that the client will try to set up connection. If the connection fails more the 10 times the client gives up and print an appropriate message to the user. Otherwise the user is welcome to enter a message. The user will receive all his messages plus the messages from all other currently signed in users.

In case of empty message the client commits exit.

## 3. **TODO**

- To implement function that will check the correctness of the input data.
- To obtain all IP address that are currently configured and and ask the

user to choose one of them.

- Set up a log system on the server.

## 4. Known problems :

The project compiles without problem on FreeBDS machine but the client failed to open a channel using Libnet. The same code runs without any problem under Red Hat Linux.

The output from the test program is not completely correct but manage demonstrate the functionality of the program.