BLG430E
COMPUTER NETWORKS
TERM PROJECT
"Socket Programming based on Game Server"

Instructor: Assoc. Prof. Dr. Berk CANBERK (canberk@itu.edu.tr)
Assistant: Res. Asst. Müge EREL ÖZÇEVİK (erelmu@itu.edu.tr)

1st Due/Demo Date: 26.04.2016 **2nd Due/Demo Date:** 10.05.2016

OBJECTIVES

The purpose of this project is to give you experience with socket programming in C language. You will obtain information about creating socket, binding socket and port number, connecting establishment with clients, listening connections, sending and receiving messages over created connections etc.

The project has two main parts defined as building **Game Program** using TCP connection and **Performance Analysis** measuring **Round Trip Time (RTT)** and **Throughput**. In the first part of project, you have to write a Game program that enables playing a game between multi players in multi sessions. This part will be done over **TCP connection.** In the second part of project, you have to analyze performance of Game Server. To do this, the Game program should be rewritten over **UDP connection**. For each program built over TCP and UDP connection, **average Round Trip Time (RTT)** and **Throughput** are calculated and visualized in the performance graphs. The details of each part are examined in the following sections.

GAME PROGRAM / TCP Connection

Game program enables to create different Game sessions where multi player can play a game with each other. Firstly, all players should authenticate the Game program with predefined password and user name. Then, players start a game with other ends in their chosen game sessions. The details of each sub-parts are organized as follows:

Initializing Game Program

The client source code should take **IP ADDRESS** and **PORT NUMBER**, whereas the server source code should take only **PORT NUMBER** from the command line. To follow each step of execution, please organize your codes as the following example output format:

SERVER	CLIENTS
>Socket is created...	>Socket is created...
>Binding is done...	
>Server is listening...	>Client sends connection request to server...
>Server multiplex sockets…[1]	
>Server accepts connection request of Client...	
>New connection from **[Client address]** on socket **[Socket number]**....	>Connection established...

Authentication

[1] If there is only one connection, the ouput should be like ">There is only one connection.."

After a connection is established between a client and server, client should authenticate with defined **USERNAME** and **PASSWORD**. If any username and password have not been defined yet, please sign the client up to the application. Please indicate each step of communication while defining new password and username or signing in the application. For each step please add the time stamps as shown in example output below:

SERVER	CLIENT
>**[username]** is authenticated at **[time]**[2]	>Enter username and password: >John 321 >You authenticate at time **[time]**

Game Sessions

You should design your Game program that enables to **create, list** Game sessions and **select** one to join. Each client can list open Game sessions and usernames being communicated in them. After selection one of the session is done, the client can join the Game.

SERVER	CLIENT1	CLIENT2
>**[client1]** create the Game session **[session#]** at **[time]** >**[client2]** join the Game session **[session#]** at **[time]**	>**[client2]** join at **[time]** >Game is starting…	>**[client2]** join at **[time]** >Game is starting…

Playing Game

You are allowed to build your Game program whatever you choose. There are some example for Game programs:

- Tic Tac Toe
- Rock Paper Scissors
- Snake etc.

  Choose one of them or use your imagination!...

PERFORMANCE ANALYSIS / Round Trip Time (RTT) & Throughput

In order to analyze performance of Game Server, you should rewrite program over **UDP connection**. Then, for each Game programs that are built over TCP or UDP connections, you should measure **Round Trip Time (RTT)** for each game step and calculate **average** of them. As a result, first graph should visualize the average RTT of Game Servers over TCP and UDP, as the session number increases (do your measurement under that scenario: each player starts playing game at the same time and continue with same steps). The second graph should include **Throughput** analysis of Game programs.

[2] The **[time]** format should be like (hours,minutes,seconds).

SUBMISSION & DEMO GUIDELINES

Organizing your submission

All relevant code files should be under one directory. This directory should be named with numbers of the three students. For example:"040100000_040100001_040100002.zip".

You may be generating more than one executable for server and client program. Please use "**MAKE**" file.

For each Demo, each group should upload a technical report (at most four pages). The report should contain important details about each part of whole project. The report should be named with the numbers of the three students in "pdf" format.

Demos

We will have a project evaluation session, for about 20 minutes per group. The schedule for this will be announced later. During the evaluation sessions, you will have to show a demo of your project. We will also ask questions to test your understanding of the code and its working. We expect both members of the group to be present during these evaluation sessions. In demos, you can evaluate your systems by using one computer for each group.

In the first demo, you're responsible for Game Program/TCP connection. Your group will be graded over %10 part of project. In the second demo, the Performance Analysis of Game program over UDP and TCP connection, will be graded over the remain %10 part of your project.

IMPORTANT NOTES

You can create threads or processes while generating your codes, but the File Descriptors is preferred. There won't be any difference while grading your codes because of this.