BLG433E
COMPUTER COMMUNICATIONS
HOMEWORK1
"Socket Programming based on Performance Server "

Instructor: Prof. Dr. Sema F. OKTUG (oktug@itu.edu.tr)
Assistant: Res. Asst. Müge EREL ÖZÇEVİK (erelmu@itu.edu.tr)
Due Date: 23.10.2017 , 23:50

OBJECTIVES

The purpose of this homework is to give you experience with socket programming in python 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. This homework is defined as building a **Performance Program** using multiple **TCP connections**. To be used in the lectures, please prepare a Performance Program of which features are given following section in detail.
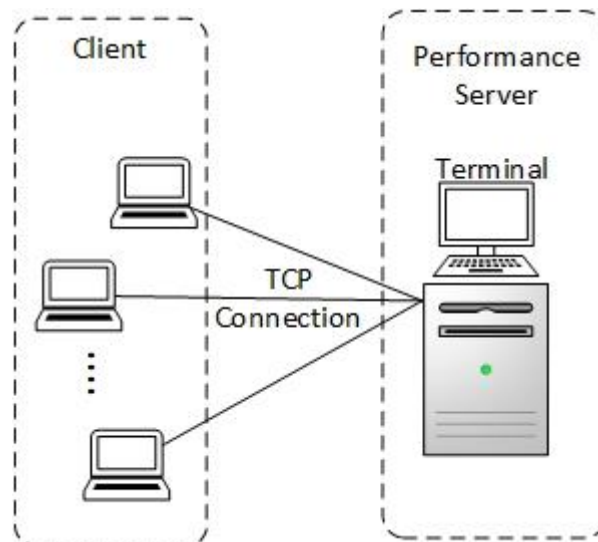
PERFORMANCE PROGRAM

Figure 1: Architecture of Performance Program

The performance program has two main functions: **Taking Attendance**, and **Making Quiz**. It can be used to mark attendance of students (clients) and also quiz them by grading fastest to slowest one. The system architecture of the performance program is shown in Fig.1. It has two main components named as the Client, and the Performance Server. The main requirements of the system are defined as follows:

- **Client** communicates with the performance server by executing given Client.py. You are not allowed to do any change on Client code (Note: Only python3 based changes will be accepted).
- **Performance Server** is a concurrent server that has following steps:
  - **Authentication:** Each client has username and their individual passwords. To start the quiz or sign own attendance, they should authenticate to the Server.
  - **Taking Attendance:** After authentication is done, each client signs attendance and sent to the server (Design of signing is up to you). An output file is enough to aggregate them in the server side.
  - **Making Quiz:** After authentication is done, Questions are sent to the client one by one and answer is received from the client. Each client has 30 minutes to finish quiz.
    - **Grading:** Each correct answer is graded with 10 points. At the end of quiz, all clients would be graded over 100. The grades of students have bonus points that are reduced from fastest to slowest one.

HOMEWORK GRADING

- **Attending the quiz (30 points)**
  - The performance server for quiz of socket programming will be opened on October, 10th at 20:00.
  - Execute given client code as: **python Client.py** (do not change server IP in the client code).
  - Please authenticate to server with an input as: "StudentID Password". Passwords are announced individually over Ninova.
  - **Each student has 30 minutes for 10 questions. Providing quiz environment is under your responsibility.**
  - Please do not exit from quiz with keyboard interrupts such as **Ctrl^C,Ctrl^X,** and **Ctrl^Z.** In quiz, **you will be graded only once.** Afterwards, you will not be allowed to quiz.
  - The questions are selected from pool, randomly.
  - The class list can be seen by requesting results from WebServer. To do request from your browser, please type following in it:
    - **160.75.26.161:12001/index.html**
  - **The grades of students will have bonus points that are reduced from fastest to slowest one** according to shown list in WebServer. Therefore, making quiz by studying Socket Programing will have important effect on grades.

- **Building the Server code (70 points)**
  - Please build the architecture, which is coded in python programming language (python2 or python3 are allowed).
  - In each action, status of Performance Server with **timestamps** should be printed.
    - The Server should be concurrent. Therefore, you can use threads, processes, or file descriptor operations (such as select()).

ORGANIZING YOUR SUBMISSIONS

The homework is built individually. All relevant code files should be under one directory (If you are coding on python3, please add also client code that can be executed with python3). This directory should be named with the number of the student such an example:"150100000.zip". It should be uploaded to Ninova. The total grades of homework will be announced later.