Lab Instruction
Socket Programming
Due: Feb 14, 2024 11:59 PM

Introduction In this homework students are required to implement two simple programs that use socket to communicate with a server. You can finish this project in one of following programming languages: Java, C/C++ or Python.

Server The server implementation is given as lab material. You are required to run the server on your machine to test your codes.

Grading The grades for this homework is separated into four parts:
2 points: Part 1
2 points: Part 2
0.5 points: The program compiles and runs correctly
0.5 points: Submit correct files and an informative ReadMe.

Part 1 TCP Socket
Implement a simple program that use a TCP socket to communicate with a server. Follow the instructions listed below to get full-credit for this part:

1. Start the server locally: python lab1_s.py
2. Send your **lastname.#** to server with TCP socket. Port number: **12345**. (1 point)
   example code: **socket.send("buckeye.1")**
3. Receive a string from the server (1 point)
   example code: **received = socket.recv(4096)**

Put the codes you use for step 1 and 2 in a script named **tcp_socket.java or .c or .py**. Make sure your script compiles and runs in terminal with one of following comments:

Java javac tcp_socket.java; java tcp_socket
C/C++ gcc tcp_socket.c -o tcp_socket; ./tcp_socket
Python python tcp_socket.py

Part 2 UDP Socket
Implement a simple program that use a UDP socket to communicate with a server. Follow the instructions listed below to get full-credit for this part

1. Start the server locally: python lab1_s.py
2. Send the string you received from part 1 to the server with UDP socket. Port number: **54321**. (1 point)
   example code:
   s.sendto("202fc9e9-48ea-409f-b72f-246eea4e537d", (host address,port#))

3. Receive a confirmation from the remote server, which indicates that you have finished both part 1 and part 2 of this project. (1 point)
   example code: **s.recvfrom(4096)**

Put the codes you use for step 1 and step 2 in a script named **udp_socket.java or .c or .py**. Make sure your script compiles and runs in terminal with one of following comments:

Java javac udp_socket.java; java udp_socket
C/C++ gcc udp_socket.c -o udp_socket; ./udp_socket
Python python udp_socket.py

Testing Make sure you test the code before you submit the project. You will get .5 point for making sure your codes compiles and runs correctly **in terminal**. (0.5 points)

Environment You can test your program on student stdlinux, if your program compiles and runs correctly on stdlinux, you can assume that TAs can compile and run your program.

Libraries You should not use any external libraries for this homework, just use the class and functions that are provided by the programming language you use.

Submission Submit following files in an archive file (no folder is needed) to get full credit for this lab:

1 tcp_socket.java or .c or .py

2 udp_socket.java or .c or .py

3 Readme (.5 points)

In the Readme, include instructions (compiler versions and other information) to run your program, or any external materials you used to finish this project.

Clarification 1 uuid You could write your uuid to a file in part 1, and read the file in part 2. Or you could hardcode the uuid to your program in part 2.