

Project Part 2: Basic TCP/IP Socket Programming

Overview

This is Part 2 of a project consisting of four parts. Each part of the project is worth 5% of your final course grade. Refer to your Suggested Course Schedule to confirm your Part 2 due date.

Read all the project directions and questions before you begin preparing your answers, and consult your Open Learning Faculty Member if you have any questions about the project.

The overall purpose of this Project is to design and develop a simple data link layer protocol using C programming language.

Objectives

In Project Part 2, you will gain a better understanding of Transmission Control Protocol (TCP) / Internet Protocol (IP), experience using sockets as a communication mechanism, and explore how to use TCP and User Datagram Protocol (UDP) with sockets.

Let's think of an application that sends data from an end host computer system to another computer. A reasonably good way to send data over the Internet is to use a transport layer protocol, such as Transmission Control Protocol (TCP) or User Datagram Protocol (UDP), or a network layer protocol Internet Protocol (IP), not data link layer protocols. (Why?) Here are some questions:

- Do we really have to develop those protocols in the application? No, we just need to use one of them because they are implemented in the operating system kernels.
- What protocol, then, do we need to use? TCP, UDP or IP? It depends on our applications.
- How do we use those protocols in our applications?

Instructions

1. Read the section called *Project Part 2: Background Information*. You will need this information to complete Part 2 of the project.
2. After you have read and understand the background information, implement the following two programs:
 - a. Complete UDP echo server and client programs.

- b. Complete TCP echo server and client programs.

Requirements for Project Part 2

The messages must be given to the client programs as arguments, not only one argument, from the user. The client programs need to send the messages separately, not at once as one message.

All your programs should print your name and student number on the terminal window.

You need to test your programs on two terminals, one terminal for server and another terminal for client. The necessary port number and IP address should be given from the user as arguments.

Submit screen shots that show how your programs are compiled and run. Please refer to the general guidelines for preparing and submitting your project found under the Assignments Overview tab of your course.

Project Marking Criteria

Each Project Part is marked according to the following criteria and weightings:

Marking Criteria	Weighting
No syntax error: All requirements are fully implemented without syntax errors. Submitted screen shots will be reviewed with source code.	/5
Correct implementation: All requirements are correctly implemented and produce correct results Submitted screen shots will be reviewed with source code.	/5
Total	/10