

1 Multithreaded Server Application

The solution to this programming assignment is small relative to later programming assignments. It is intended as a gentle reminder to the UNIX platform, socket programming and threads. When this project has been completed, the following outcomes should be realized.

- Programming in ANSI C or Java on the UNIX platform
- Use of on-line manual pages
- Use of TCP and UDP sockets
- Use of threads

2 Preparation

All on-line work for this course will be done in a directory named CEN6520 in your UNIX account. You should create that directory before continuing with the remainder of this assignment.

```
mkdir ~/CEN6520
```

Each programming project will be done in a separate subdirectory of the CEN6520 directory. Create the project1 directory and change to that directory for the remainder of this project.

```
mkdir ~/CEN6520/project1
cd ~/CEN6520/project1
```

3 The Problem

There are two parts to this project, each worth 5 points. The language choices are ANSI C or Java.

4 Part 1

You are to create a multithreaded TCP server. For each connection request, a thread is created to handle the client request. The main server thread should handle all server connection requests. The service provided by this server is trivial. Each client will randomly generate an integer value and send it to the server. The server's responsibility is to keep track of each client's random number. You may assign an id (1,2,...) to each client.

Once a client is connected to the server, it should send its random number. A client may also request the random number of another client. The details of how this is programmed are your design decisions, but the decisions must be based on legitimate reasons.

5 Part 2

Repeat part 1 but instead of TCP use solely UDP.

6 Due Date: Thursday, Jan 13th 11:59pm

Late assignments will not be accepted.