CS 3210 – Computer Networks Lab
Jan – Apr 2014
Shankar Balachandran

Programming Assignment 2 – UDP Based File Server
Due date : March 10, 2014, 9 AM, on Moodle

Assigned on February 26, 2014

1. Description

The purpose of this assignment is to help you understand Client-Server connections and how network noise is handled by the various participating components. The assignment requires you to do the following :

1. A client interface which can be used to view the files in a remote directory. Use port 8500 to receive the responses.
2. A remote server on which the files reside. Use UDP port 8000 to take in requests.

We will assume that there is exactly one client connecting to a server at any point of time. Thus, the state of the system is implicitly maintained.

The client makes the following kinds of requests:

0. **open** – Indicating that the client needs information from the server
1. **ls** - List the contents of the current directory
2. **cd** - Change to a directory
3. **pwd** - Display current working directory
4. **get** – get a file from the remote directory
5. **done** – indicating that the client has no further requests

By default, the current directory is set up to be “/” the root directory.

ls: Listing of the contents should support the basic options of the unix command **ls** namely -a, -f, -l, -h, -r, -s and -t options.

cd: “Change to a directory” command can take a parameter which is one of the following:

.. - to go to the parent directory
/ - to go to the root directory
<dir name> - do go to the directory named <dir name>

pwd: If the command is pwd, the remote server should get back to you with the current directory

get: It takes one file name as a parameter and fetch the contents of the file and store it locally.

Use the Unix commands and system calls to do some of the basic file system actions.

In connectionless model we do not have connect call at the client because there is no connection that will be established. A connectionless server does not listen continuously but provides service when a request arrives from the client.

1. Server:

```
      1. Initialise in-memory directory from a file
      2. Open a UDP socket
      3. Do Forever: receive and process requests
            3.1 Receive a UDP packet
            3.2 if it is a valid request
                  3.2.1 perform the request
                  3.2.2 put the result in one or more packet s
                  3.2.3 send it to the client
            3.3 else ignore the packet
```

2. Client:

```
      1. Open a UDP socket
      2. Get a request from the user
            2.1 Construct a request appropriate for the server
            2.3 Send it to the server
            2.4 Receive a response from the server
                  2.4.1 Upon timeout, repeat step 2.3
            2.5 Display the response
      3. Close the socket
```

Note that packets may be lost in either side or can result in incomplete/incorrect information in the client's side. For this homework, you don't have to do any error-recovery.

2. What to Submit

The source code along with the make file in the desired format as explained in the previous assignment.

Note : This is an INDIVIDUAL assignment. If you are having any difficulty with the assignment, please talk to the TAs/instructor. No downloaded code from the web.