CS69003 Computer Systems Lab – I

Assignment 8

Deadline: November 7, 2007

Write a concurrent server using TCP sockets that has the functionality of a simple ftp server. The server waits for a connection from the client, and when a client connects, forks a process to respond to that client. The main server process (the parent) then comes back to wait for the next client connection. The forked process that handles the client's requests first sends the string "login:" to the client. The client displays the string on the screen and waits for the user to type in his login name. The user types in a login name which is then transmitted to the server. The server then sends the string "password:" to the client. The client displays this on the screen and the user types in his password, which is transmitted to the server. If the login and password do not match, an error message is sent to the client and the server closes the connection. The client prints this message in the screen and exits. If the login-password matches, the server sends the string "ok>" which is displayed on the screen. The client then waits for a command from the user. The user can type in any of the following commands, which are transmitted to the server, and the server executes the command and responds with the result, if any, followed with "ok>" in the next line through the socket. The client then prints this response (which also gives the prompt ok>) and waits for the next command. This process continues until the user types the command "exit" at which point the client and the server should close the connection. The following commands are supported at the client end:

a. **pwd –** shows the current directory **on the server side**.
b. **cd <dirname>** – the directory **on the server side** is changed to the directory specified.
c. **ls** – shows the contents of the current directory on the server side
d. **get <filename>** - copies the file from the current directory **on the server side** to the current directory **on the client side** with the same filename. If a file with the same name exists on the client side, it is overwritten. If the remote file does not exist, the server sends back a failure message to the client. The client prints a message "file transferred" after the file is successfully transferred.
e. **put <filename>** - copies the file from the current directory **on the client side** to the current directory **on the server side** with the same filename. If a file with the same name exists on the server side, it is overwritten. The server sends a message "ok" to the client after the file is successfully written. The client prints a message "file transferred" after the file is successfully transferred.
f. **exit –** quits the connection.

Note that multiple clients can try to write to the same server file. To see how to get the current working directory and change the directory from within a C program, look up the file and directory related function in the C library functions (link to list of all C library function given in course website).

To check for login-passwords, first create a text file manually on the server side with the five login names and corresponding passwords. The format of the file should be as follows:

```
user1   user1Password
user2   user2Password
user3   user3Password
```

Name the C files client.c and server.c. Tar all the programs into a single tar file Assgn6.tar and mail it to course email together with the subject line "Assignment 8 for 07CS60**".

.