

Socket Programming Assignment 4 – C Implementation of Client-Server Communications

Goal: Practice makes perfect! Socket programming assignments are to help you review and apply your conceptual knowledge from this class. This one is the **C implementation** of client-server communications, per the following instructions. It will greatly help you gain deeper understanding of the computer's socket operations.

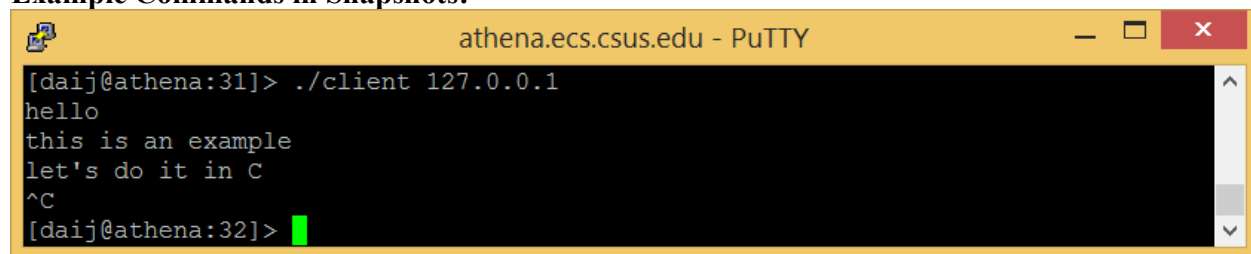
- Client reads a line of characters (data) from its keyboard and sends the data to the server;
- The server receives the data and displays it to its screen.

Attention: Code plagiarism is absolutely **NOT** allowed! Please prepare for a **demonstration** of running your program in front of the instructor/grader and answer their questions upon request.

Instructions: Please repeat what's done in the course slides about implementing the **TCP** client/server interactions with C.

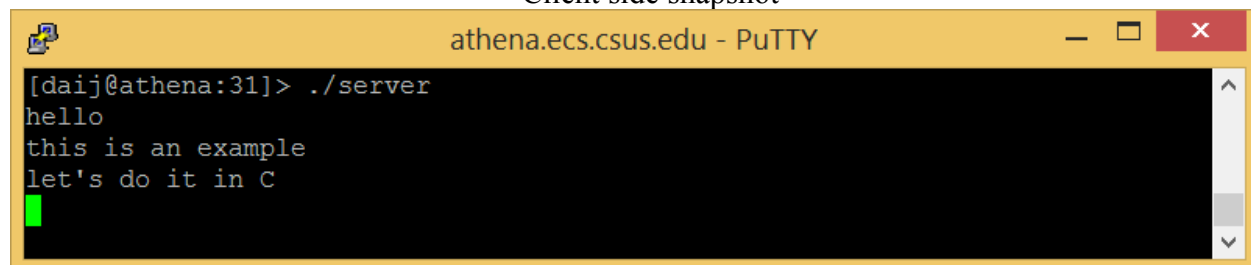
Server command: `./server`
Client command: `./client 127.0.0.1`

Example Commands in Snapshots:

A screenshot of a PuTTY terminal window titled "athena.ecs.csus.edu - PuTTY". The terminal shows a user prompt "[daij@athena:31]> ./client 127.0.0.1". The user then types "hello", "this is an example", and "let's do it in C". The prompt changes to "[daij@athena:32]>".

```
athena.ecs.csus.edu - PuTTY
[daij@athena:31]> ./client 127.0.0.1
hello
this is an example
let's do it in C
^C
[daij@athena:32]>
```

Client side snapshot

A screenshot of a PuTTY terminal window titled "athena.ecs.csus.edu - PuTTY". The terminal shows a user prompt "[daij@athena:31]> ./server". The user then types "hello", "this is an example", and "let's do it in C". The prompt changes to a green cursor.

```
athena.ecs.csus.edu - PuTTY
[daij@athena:31]> ./server
hello
this is an example
let's do it in C
```

Server side snapshot

Deliverable: A project report, an **electronic submission** to Canvas, is expected to include both your **source code** and some **screenshots** that can help you demonstrate your work (**commands, operations, results and analysis**). Code plagiarism is absolutely **NOT** allowed! Please also prepare for a **demonstration** of running your program in front of the instructor/grader and answer their

questions (which are about your code). Your grade will be based on both the report and your performance during demonstration.

Requirement: The report will all be evaluated based on the following grading criteria.

Functionality Correctness	60%
Security/Robustness	40%

Secure Programming Notes:

A “**clinician**” will help you with improvements regarding robustness. **For each assignment, you can have two submission opportunities: one to catch the first deadline, and the other one after correction for the second deadline. Only students who visit the clinic will be given the opportunity to get lost points back.** Specifically, the second submission could earn back **100%** of the lost points in the first one due to non-robustness. Hence, you are strongly encouraged to see the clinicians for correction. The clinicians will hold regular office hours during which students can come and consult on a drop-in basis. The information about the clinician will be announced to class via separate email through Canvas. **The clinician is NOT the grader for this course.**

Clinician: Priyanka Makwana

Clinician Office Hours at Riverside Hall 2009

Monday	Wednesday	Friday
2 pm to 4 pm	12 pm to 4 pm	12 pm to 4 pm