Assessment 1

Computer Networks - 1

November 14, 2022

1 Problem statement

Demonstration of simple calculator using connectionless socket programming. Here, a client must be able to perform simple calculator tasks such as addition, subtraction, multiplication, power, and division.

2 Server Side

2.1 Commands

Server must be able to perform following functions based on the command received:

Command	**Function**	**Return value**
ADD	Add given numbers	answer if successful, suitable error code otherwise
SUB	Subtract given numbers	answer if successful, suitable error code otherwise
MULT	Multiple given numbers	answer if successful, suitable error code otherwise
DIV	Divides two numbers	answer if successful, suitable error code otherwise
POW	Returns a power b	answer if successful, suitable error code otherwise

2.2 Error codes

The following error codes must be sent by the server in case of invalid scenarios

Error code	Scenario
INV_ARGS	Invalid arguments
INSUF_ARGS	Insufficient arguments
DZERO	Divide by zero
INV_CMD	Invalid command

Note: You are free to use other error codes for any other exceptions, if required.

3 Client side

A client must be able to perform the following activities by sending commands in a particular format. A special variable called "ANS" can be used to add the current result of the server to the given numbers.

3.1 Commands

1. **Addition**

   `ADD -n <numberOfArgs> <args..>`

   Here, -n represents the number of arguments (numbers) to be added. The minimum number of arguments must be 2.

   Cases to check:

   - Invalid command
   - Insufficient arguments
   - Invalid arguments

Example 1: C: `ADD -n 3 100 300 200`
S: 600
C: The result is 600

Example 2: C: `ADD -n 3 100 200`
S: INSUF_ARGS
C: Insufficient arguments.

Example 3: C: `ADD -n 3 ANS 1000 2000`
S: 3600 //Assuming the previous operation performed in the server yielded result 600
C: The result is 3600

2. **Subtraction**

   `SUB -n <noOFArgs> <args...>`

   Here, -n represents the number of arguments (numbers) to be subtracted. The minimum number of arguments must be 2.

   Cases to check:

   - Invalid command
   - Insufficient arguments
   - Invalid arguments

Example 1: C: `SUB -n 3 100 200 300`
S: -400
C: The result is -400

Example 2: C: `SUB -n 2 ANS -600`
S: 200 //Previous answer in the server is -400
C: The result is 200

Example 3: C: `CREATE -n 1 400`
S: INSUF_ARGS
C: Insufficient arguments.

3. **Multiplication**

   Same as addition and subtraction

4. **Division**

   `DIV <numerator> <denominator>`

   Divides first argument by the second. Note that, the second argument cannot be 0. The result could be in decimal as well. Display up to six decimal places

5. **Power**

   `POW <base> <exp>`

   Returns the value of the power to the base.

Example 1: C: `POW 4 2`
S: 16
C: The result is 16

Example 2: C: `POW 4 0.5`
S: 2
C: The result is 2

4 Submission instructions

ANY one caught plagiarizing the code, ALL people involved will be given ZERO as their internal assessment marks. No excuses shall be accepted.

1. Create two files with names `<rollNo>_client.c` and `<rollNo>_server.c`
2. Put them into ONE zip file with name `<rollNo>.zip`
3. Upload it in the LMS before the deadline.