Spencer Reed
11/14/22

CS 270 – Program #4 Design Sheet

Objective:

Creating a simple client/server program that will read a line of data from the client that is an equation in prefix notation. After reading this data it will write the data to the server which wil perform the operation and write the result back to the client.

Data Structures/Methods Needed:

- A method to open of the socket
- A method to conenct the client and server
- A method to write from the client to the server
- A method to read the client data on the server
- A method to interperet the user data
- A method to calculate the equation
- A method to write the result back to the client
- A method to read the data on the client from the server
- A method to display the result on the client.

Steps Needed:

1. Using the code given from https://www.linuxhowtos.org/C_C++/socket.htm, read the line piece by piece on the server side.
2. Using this line find which operator the client wants to use.
3. Write a switch method that will perform different operations on the two integers based on the operator sent.
4. Send the result back to the client.
5. Receive the correct data on the client and display it.

Spencer Reed
11/14/22

Program 4 Server Code

```
/* server.c

   CS 270.Bolden.........Compiler version..........Spencer Reed
   11/14/22..MacBook Air, 1.1GHz quad-core Intel Core i5..reed7385@vandals.uidaho.edu

   This program opens a socket at the port designated, then reads input from
   the client as an operation in prefix notation. After this it solves the
   equation and writes it to the client.

   Original code taken from https://www.linuxhowtos.org/C_C++/socket.htm
 *-------------------------------------------------------------------------
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

//Prints an error message when called and stops the program
void error(const char *msg)
{
    perror(msg);
    exit(1);
}

int main(int argc, char *argv[])
{
     int sockfd, newsockfd, portno;
     socklen_t clilen;
     char buffer[256];
     int result;
     struct sockaddr_in serv_addr, cli_addr;
     int n;
     if (argc < 2) {
         fprintf(stderr,"ERROR, no port provided\n");
         exit(1);
     }
     sockfd = socket(AF_INET, SOCK_STREAM, 0);
     if (sockfd < 0)
        error("ERROR opening socket");
     bzero((char *) &serv_addr, sizeof(serv_addr));
     portno = atoi(argv[1]);
     serv_addr.sin_family = AF_INET;
     serv_addr.sin_addr.s_addr = INADDR_ANY;
     serv_addr.sin_port = htons(portno);
     if (bind(sockfd, (struct sockaddr *) &serv_addr,
              sizeof(serv_addr)) < 0)
              error("ERROR on binding");
     listen(sockfd,5);
     clilen = sizeof(cli_addr);
     newsockfd = accept(sockfd,
                 (struct sockaddr *) &cli_addr,
                 &clilen);
     if (newsockfd < 0)
          error("ERROR on accept");
     bzero(buffer,256);
```

```
    n = read(newsockfd,buffer,255);
    if (n < 0) error("ERROR reading from socket");
    printf("Here is the problem: %s\n",buffer);
        int firstInt, secondInt;
        char operator;
        sscanf(buffer, "%c %i %i", &operator, &firstInt, &secondInt);
        printf("Operator: %c\n", operator);
        printf("First Integer: %i\n", firstInt);
        printf("Second Integer: %i\n", secondInt);
        printf("%i %c %i = ", firstInt, operator, secondInt);
        result = 0;
        int i = 0;
        switch(operator)
        {
           case '+':
              result = (firstInt + secondInt);
              n = write(newsockfd, &result, sizeof(int));
              printf("%i\n", result);
              break;
           case '-':
              result = (firstInt - secondInt);
              n = write(newsockfd, &result, sizeof(int));
              printf("%i\n", result);
              break;
           case '*':
              result = (firstInt * secondInt);
              n = write(newsockfd, &result, sizeof(int));
              printf("%i\n", result);
              break;
           case '/':
              result = (firstInt / secondInt);
              n = write(newsockfd, &result, sizeof(int));
              printf("%i\n", result);
              break;
           case '%':
              result = (firstInt % secondInt);
              n = write(newsockfd, &result, sizeof(int));
              printf("%i\n", result);
              break;
           default:
              printf("Your operator is not valid\n");
        }
        printf("\n");
    if (n < 0) error("ERROR writing to socket");
    close(newsockfd);
    close(sockfd);
    return 0;
}
```

Spencer Reed
11/14/22

Program 4 Client Code

```
/* client.c

   CS 270.Bolden.........Compiler version..........Spencer Reed
   11/14/22..MacBook Air, 1.1GHz quad-core Intel Core i5..reed7385@vandals.uidaho.edu

   This program writes an equation to a server using the port designated
   in prefix notaion, and will read the result from the server after it
   is solved and will print it

   Original code taken from https://www.linuxhowtos.org/C_C++/socket.htm
 *-------------------------------------------------------------------
 */

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

void error(const char *msg)
{
    perror(msg);
    exit(0);
}

int main(int argc, char *argv[])
{
    int sockfd, portno, n;
    struct sockaddr_in serv_addr;
    struct hostent *server;

    char buffer[256];
    int result;
    if (argc < 3) {
       fprintf(stderr,"usage %s hostname port\n", argv[0]);
       exit(0);
    }
    portno = atoi(argv[2]);
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");
    server = gethostbyname(argv[1]);
    if (server == NULL) {
        fprintf(stderr,"ERROR, no such host\n");
        exit(0);
    }
    bzero((char *) &serv_addr, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    bcopy((char *)server->h_addr,
         (char *)&serv_addr.sin_addr.s_addr,
         server->h_length);
    serv_addr.sin_port = htons(portno);
    if (connect(sockfd,(struct sockaddr *) &serv_addr,sizeof(serv_addr)) < 0)
        error("ERROR connecting");
    printf("Enter the problem you would like to solve using prefix notation: ");
    bzero(buffer,256);
```

```
    fgets(buffer,255,stdin);
    n = write(sockfd,buffer,strlen(buffer));
    if (n < 0)
         error("ERROR writing to socket");
    bzero(buffer,256);

    n = read(sockfd, &result, sizeof(int));
    int i = result;
    if (n < 0)
         error("ERROR reading from socket");
    printf("The answer is: ");
    printf("%i\n", result);

    close(sockfd);
    return 0;
}
```

Spencer Reed
11/14/22

Program 4 Output

Client Output:

Addition:

```
cs270 $ gcc client.c –o client
cs270 $ ./client localhost 4501
Enter the problem you would like to solve using prefix notation: +
3 16
The answer is: 19
```

Subtraction:

```
cs270 $ ./client localhost 4502
Enter the problem you would like to solve using prefix notation: -
6 7
The answer is: –1
```

Multiplication:

```
cs270 $ ./client localhost 4503
Enter the problem you would like to solve using prefix notation: *
4 11
The answer is: 44
```

Division:

```
cs270 $ ./client localhost 4504
Enter the problem you would like to solve using prefix notation: /
7 7
The answer is: 1
```

Mod:

```
cs270 $ ./client localhost 4505
Enter the problem you would like to solve using prefix notation: %
5 3
The answer is: 2
```

Server Output:

Addition:

```
cs270 $ gcc server.c –o server
cs270 $ ./server 4501
Here is the problem: + 3 16

Operator: +
First Integer: 3
Second Integer: 16
3 + 16 = 19
```

Spencer Reed
11/14/22

Subtraction:

```
cs270 $ ./server 4502
Here is the problem: - 6 7

Operator: -
First Integer: 6
Second Integer: 7
6 - 7 = -1
```

Multiplication:

```
cs270 $ ./server 4503
Here is the problem: * 4 11

Operator: *
First Integer: 4
Second Integer: 11
4 * 11 = 44
```

Division:

```
cs270 $ ./server 4504
Here is the problem: / 7 7

Operator: /
First Integer: 7
Second Integer: 7
7 / 7 = 1
```

Mod:

```
cs270 $ ./server 4505
Here is the problem: % 5 3

Operator: %
First Integer: 5
Second Integer: 3
5 % 3 = 2
```

Spencer Reed
11/14/22

Programming Log

Time Spent:

- Finding a method to separate the client data into three pieces: 15 minutes.
- Creating the switch: 20 minutes.
- Sending the result of the operation to the client: 1 hour 15 minutes.
- Receiving the data on the client and displaying it: 30 minutes
- Overall time spent: 2 hours and 20 minutes

Aspects of Code Learned:

- How to open sockets and connect a server and its clients.
- How to write and read different things between the server and a client.
- How to split a line of data into multiple parts with sscanf();.

Problems Encountered:

- At first I could not figure out how to pass the result to the client as it would simply not print the result, but I found it was a simple error on my part.