시스템 프로그래밍 실습

[Assignmen3-2]

Class : D 반(실습 2 금 56)

Professor : 최상호 교수님

Student ID : 2022202104

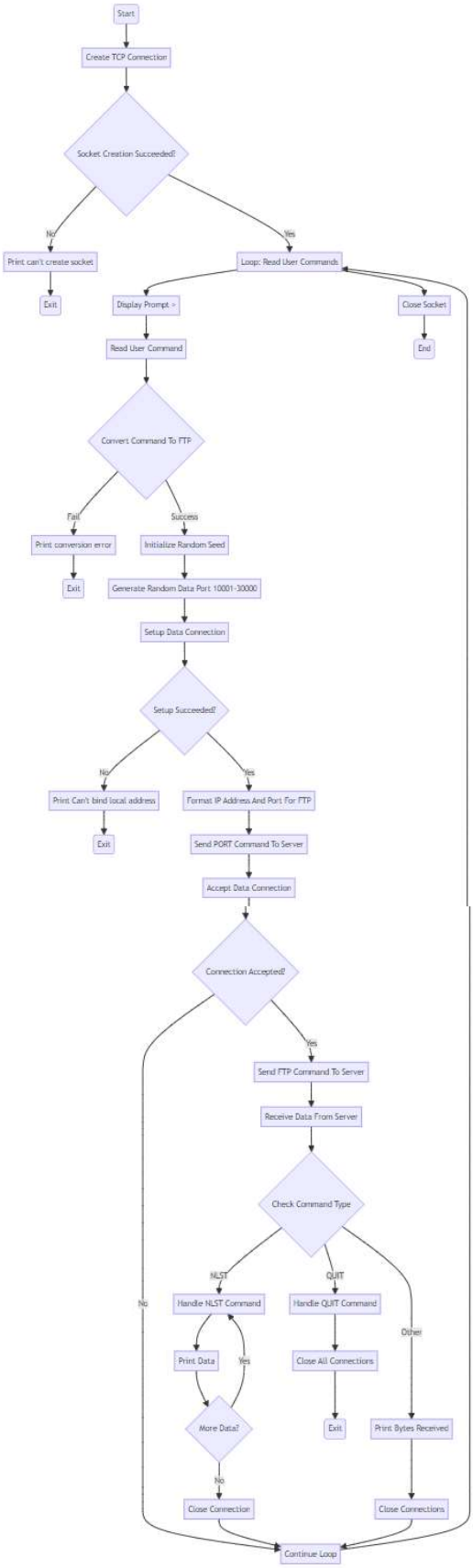
Name : 김유찬

Introduction

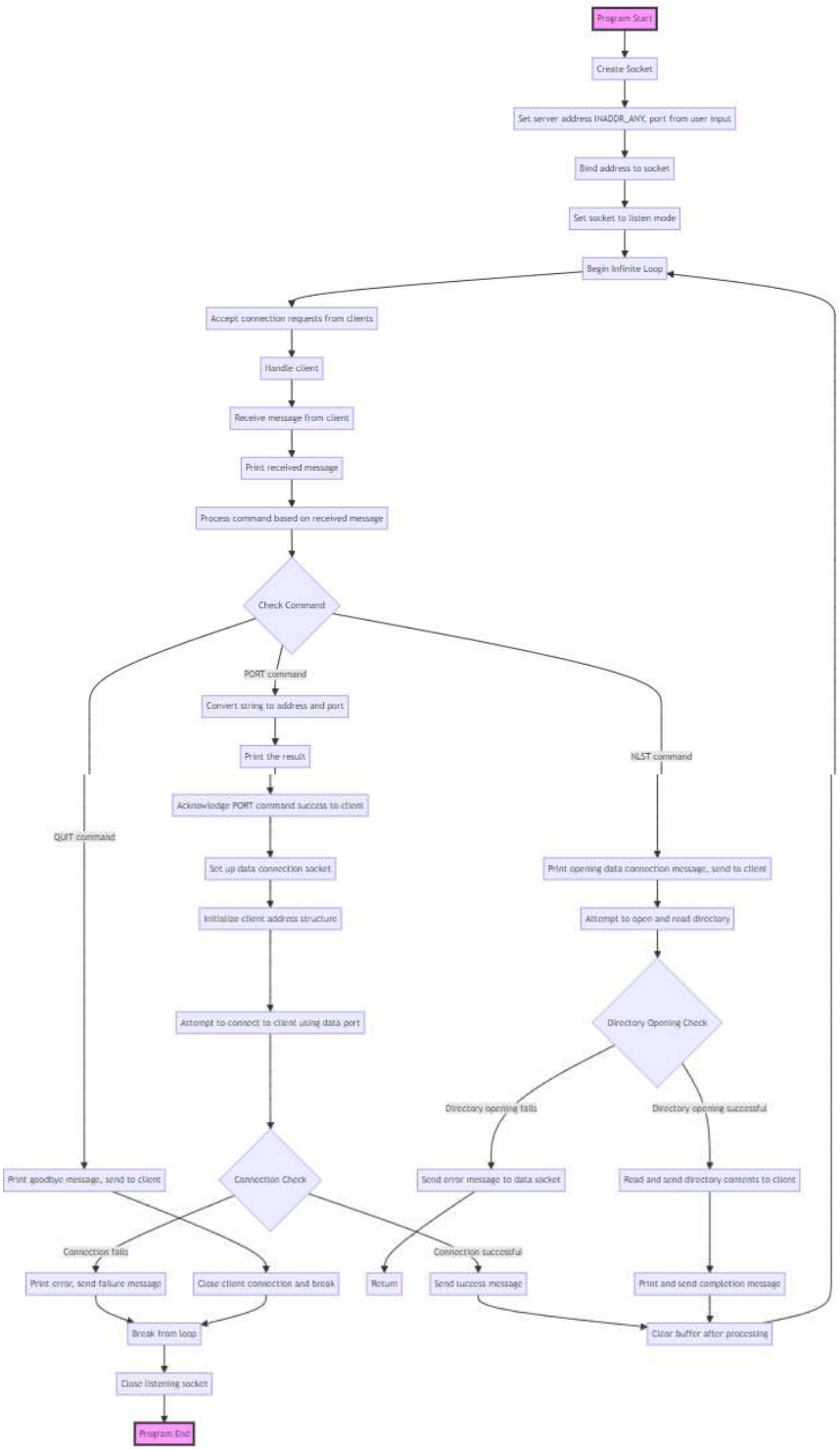
저번 과제에서는 FTP 에서의 Authorization 을 구현했다. 이번에는 FTP 에서 많이 쓰는 Control connection 과 data connection 을 구현해 볼 것이다. 이를 사용함으로써 Control connection 기능과 data connection 기능을 각 port 마다 다르게 두어서 관리가 쉽고 속도 향상이 된다. 또한 보안성 역시 높일 수 있는 장점이 있다. 이번 시간에 이를 구현해볼 것이다.

Flow chart

1) client


Start
Create TCP Connection
Socket Creation Succeeded?
No
Yes
Print can't create socket
Exit
Loop: Read User Commands
Display Prompt >
Read User Command
Close Socket
End
Convert Command To FTP
Fail
Success
Print conversion error
Exit
Initialize Random Seed
Generate Random Data Port 10001-30000
Setup Data Connection
Setup Succeeded?
No
Yes
Print Can't bind local address
Exit
Format IP Address And Port For FTP
Send PORT Command To Server
Accept Data Connection
Connection Accepted?
No
Yes
Send FTP Command To Server
Receive Data From Server
Check Command Type
NLST
QUIT
Other
Handle NLST Command
Print Data
Yes
More Data?
No
Close Connection
Handle QUIT Command
Close All Connections
Exit
Print Bytes Received
Close Connections
Continue Loop

2) server


Program Start
Create Socket
Set server address INADDR_ANY, port from user input
Bind address to socket
Set socket to listen mode
Begin Infinite Loop
Accept connection requests from clients
Handle client
Receive message from client
Print received message
Process command based on received message
Check Command
PORT command
Convert string to address and port
Print the result
Acknowledge PORT command success to client
QUIT command
Set up data connection socket
Initialize client address structure
Attempt to connect to client using data port
NLST command
Print opening data connection message, send to client
Attempt to open and read directory
Directory Opening Check
Directory opening fails
Directory opening successful
Print goodbye message, send to client
Connection Check
Send error message to data socket
Read and send directory contents to client
Connection fails
Connection successful
Print error, send failure message
Close client connection and break
Return
Send success message
Print and send completion message
Break from loop
Clear buffer after processing
Close listening socket
Program End

Pseudo code

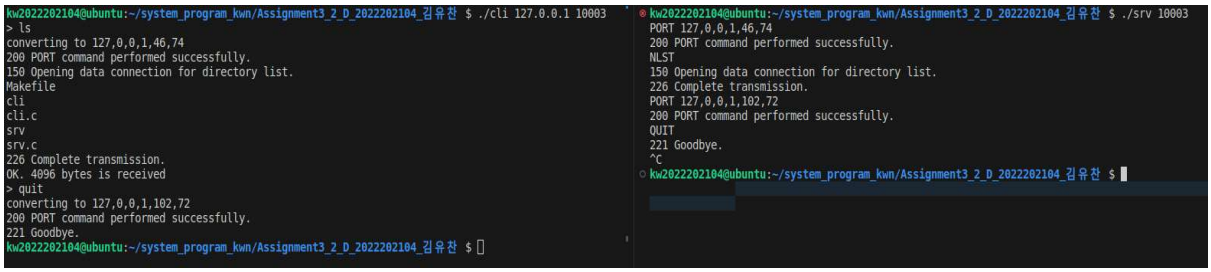
1) client

```
FTPClientSession(IP address, port number)
{
    sockfd = CreateTCPConnection(IP address, port number);
    if (socket creation fails) {
        Print "can't create socket";
        return -1;
    }

    while (true) {
        Display prompt "> ";
        Read user command into buffer;

        if (ConvertCommandToFTP(buffer, cmd_buff) fails) {
            Print "conversion error";
            Exit;
        }

        Initialize random seed;
        data_port = Generate a random port between 10001 and 30000;

        listenfd = SetupDataConnection(data_port);
        if (SetupDataConnection fails) {
            Print "Can't bind local address";
            return 1;
        }

        hostport = FormatIPAddressAndPortForFTP(temp.sin_addr.s_addr, servaddr.sin_port);
        Print "converting to ", hostport;

        SendPORTCommandToServer(sockfd, hostport);

        connfd = AcceptDataConnection(listenfd);
        if (connection acceptance fails) {
            Continue;
        }

        SendFTPCommandToServer(sockfd, cmd_buff);
        count = ReceiveDataFromServer(sockfd, buffer);

        if (Command equals "NLST") {
            do {
                count += ReceiveDataFromServer(sockfd, buffer);
                Print buffer;
            } while (additional data available);
        } else if (Command equals "QUIT") {
            HandleQuitCommand(sockfd, connfd, listenfd);
            return 0;
        }

        Print "OK. ", count, " bytes is received";
        CloseConnection(connfd, listenfd);
    }
    Close sockfd;
}
```

2) server

```
Program Start

Create Socket
Set server address (INADDR_ANY, port from user input)
Bind address to socket
Set socket to listen mode

Begin Infinite Loop
    Accept connection requests from clients
    Handle client:
        Repeat
            Receive message from client
            Print received message

            Process command based on received message:
                If ("PORT" command) {
                    Convert string to address and port, print the result
                    Acknowledge PORT command success to client
                    Set up data connection socket
                    Initialize client address structure
                    Attempt to connect to client using data port
                    If (connection fails) {
                        Print error, send failure message, and break
                    }
                    If (connection successful) {
                        Send success message
                    }
                }

                If ("NLST" command) {
                    Print opening data connection message, send to client
                    Attempt to open and read directory
                    If (directory opening fails) {
                        Send error message to data socket
                        Return
                    }
                    Read and send directory contents to client
                    Print and send completion message
                }

                If ("QUIT" command) {
                    Print goodbye message, send to client
                    Close client connection and break
                }

            Clear buffer after processing

    Close listening socket
    Program End

    Function convert_str_to_addr {
        Allocate memory for address storage
        Extract IP and port parts using sscanf
        If (format is incorrect) {
            Print error and return NULL
        }
        Format IP address into dot notation
        Calculate port number (p1 * 256 + p2)
        Return formatted IP address
    }
```

결과화면



ls 명령어를 client 에서 입력했으면 먼저 Control connection 이 실행되어서 그와 관련된 ip 와 port 에 대한 결과값이 출력이 되고 그 다음은 Data connection 을 통해 ls 에 대한 출력값이 출력된다.

고찰

Control connection 과 data connection 개념이 생소해서 처음에 어떻게 구현해야할지 해맸다. 하지만 client 가 server 가 되고 server 가 client 가 되는 것만 구현할 줄 알면 쉽게 구현할 수 있다. 이번에는 client 와 server 와의 write, read 갯수를 서로 맞추질 못해서 원하지 않는 출력이 나왔다. 그 밖에 ls 구현, quit 구현은 이미 짜논 코드가 있어서 쉽게 구현할 수 있었다.

Reference

시스템프로그래밍 실습 강의자료