Homework 4

Dec 21, 2018 (Fri)

Eunggu Yun

1. Introduction

1) Problem 1 – Controlling Drones

There are two programs: client and server. These two programs are the drone control client program and the drone control server program which are using TCP protocol to communicate.

The client program sends several drone control commands (Move_UP, Move_DOWN, Move_LEFT, and Move_RIGHT) to the server and receives the current drone position from the server.

The server program receives several drone control commands from the client and moves the drone within the boundary. For each command, it sends the drone position to the client.

2) Problem 2 – Online Calculator

There are two programs: client and server. These two programs are the online calculator client program and the online calculator server program which are using TCP protocol to communicate.

The client program sends the expression consisting of + and - to the server and receives the result of the expression from the server.

The server program receives the expression from the client and calculates the expression. Then it sends the result to the client.

3) Problem 3 – Simple DNS Simulator

There are three programs: client, lserver and gserver. These three programs are the DNS client program, the local DNS server program, and the global DNS server program which are using UDP and TCP protocols to communicate.

The client program sends the domain name to the local DNS server using UDP and receives the IP address from it using UDP.

The local DNS server program receives the domain name from the client using UDP and search local DNS information to find the IP address that corresponds the domain name. If there is an information, it sends the IP address to the client using UDP. If there isn't an information, it sends the domain name to the global DNS server using TCP and receives the IP address from it using TCP. Then it sends the IP address back to the client using UDP.

The global DNS server program receives the domain name from the local DNS server using TCP and search global DNS information to find the IP address that corresponds the domain name. Then it sends the IP address to the local DNS server using TCP.

2. Development Environment

1) Platform (OS): Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-31-generic x86_64)

2) Compiler: gcc 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.10)

(This is the SKKU server's development environment – swye.skku.edu)

3. How to build and run

1) Problem 1 – Controlling Drones

① Build

To build the server and the client programs:

```
$ make all
```

To build the server program only:

```
$ make server
```

To build the client program only:

```
$ make client
```

To clean up output files:

```
$ make clean
```

② Run

Make sure there is the moves.txt file in the directory with this format:

```
Move_UP
Move_UP
Move_DOWN
Move_LEFT
```

To run the server program:

```
$ ./server [port]
```

For example,

```
$ ./server 1234
```

To run the client program:

```
$ ./client [server_ip] [port]
```

For example,

```
$ ./client 127.0.0.1 1234
```

2) Problem 2 – Online Calculator

① Build

To build the server and the client programs:

```
$ make all
```

To build the server program only:

```
$ make server
```

To build the client program only:

```
$ make client
```

To clean up output files:

```
$ make clean
```

② Run

Make sure there is the expression.txt file in the directory with this format:

```
1+5-2+7-3
```

To run the server program:

```
$ ./server [port]
```

For example,

```
$ ./server 1234
```

To run the client program:

```
$ ./client [server_ip] [port]
```

For example,

```
$ ./client 127.0.0.1 1234
```

3) Problem 3 – Simple DNS Simulator

① Build

To build the server and the client programs:

```
$ make all
```

To build the gserver program only:

```
$ make gserver
```

To build the lserver program only:

```
$ make lserver
```

To build the client program only:

```
$ make client
```

To clean up output files:

```
$ make clean
```

② Run

Make sure there is the domain.txt file in the directory with this format:

```
www.skku.edu
```

Make sure there are the local_dns.txt, global_edu_dns.txt, and global_com_dns.txt with this format

```
www.skku.edu 212.124.11.20
www.seoul.edu 212.124.11.14
```

To run the gserver program:

```
$ ./gserver
```

To run the lserver program:

```
$ ./lserver
```

To run the client program:

```
$ ./client
```

4. Project Structure

```
Project Folder
├ Problem 1 – Controlling Drones
│ ├ client.c
│ ├ Makefile
│ ├ moves.txt
│ └ server.c
├ Problem 2 – Online Calculator
│ ├ client.c
│ ├ expression.txt
│ ├ Makefile
│ └ server.c
└ Problem 3 – Simple DNS Simulator
  ├ client.c
  ├ domain.txt
  ├ global_com_dns.txt
  ├ global_edu_dns.txt
  ├ gserver.c
  ├ local_dns.txt
  ├ lserver.c
  └ Makefile
```

1) Problem 1 – Controlling Drones

① Makefile

A file for GNU Make. It gives the knowledge of how to build the client and the server programs to GCC compiler.

② client.c

A source code of the drone control client program.

③ server.c

A source code of the drone control server program.

④ moves.txt

A text file for the drone control client program which stores drone movement commands.

2) Problem 2 – Online Calculator

① Makefile

A file for GNU Make. It gives the knowledge of how to build the client and the server programs to GCC compiler.

② client.c

A source code of the online calculator client program.

③ server.c

A source code of the online calculator server program.

④ expression.txt

A text file for the online calculator client program which stores an expression to be calculated.

3) Problem 3 – Simple DNS Simulator

① Makefile

A file for GNU Make. It gives the knowledge of how to build the client, the lserver and the gserver programs to GCC compiler.

② client.c

A source code of the DNS client program.

③ lserver.c

A source code of the local DNS server program.

④ gserver.c

A source code of the global DNS server program.

⑤ domain.txt

A text file for the DNS client program which stores a domain name to be queried.

⑥ local_dns.txt

A text file for the local DNS server program which stores all local DNS information.

⑦ global_com_dns.txt

A text file for the global DNS server program which stores DNS information with ".com".

⑧ global_edu_dns.txt

A text file for the global DNS server program which stores DNS information with ".edu".

5. Implementation Details

1) Problem 1 – Controlling Drones

The drone control server program gets one argument: port number. When it starts, it creates a TCP socket with given port, binds the socket, and listens for client connection. When a client makes a connection, it accepts a client and opens new TCP socket for data transmission. Because the server no longer needs to wait for another client, it closes the listen socket. After the socket for data transmission created, it receives drone control commands (Move_UP, Move_DOWN, Move_LEFT, Move_RIGHT) (Receive 16 bytes) from the client until it receives "exit" command. For each command, it moves the drone within the boundary of 10x10 area and sends the current drone position to the client (Send 16 bytes). After receiving "exit" command, it closes the socket and terminates.

The drone control client program gets two arguments: server IP address and port number. When it starts, it creates a TCP socket and connects to the drone control server. When the connection is accepted by the server, it reads the "moves.txt" file and gets the drone control commands one by one. For each command, it sends the command to the server (Send 16 bytes) and receives the current drone position from the server (Receive 16 bytes). Then it saves the drone position to the "position.txt" file. After that, it sends "exit" command to the server and terminates.

2) Problem 2 – Online Calculator

The online calculator server program gets one argument: port number. When it starts, it creates a TCP socket with given port, binds the socket, and listens for client connection. When a client

makes a connection, it accepts a client and opens new TCP socket for data transmission. Because the server no longer needs to wait for another client, it closes the listen socket. After the socket for data transmission created, it receives the expression consisting of + and - from the client (Receive 1024 bytes). Then it calculates the expression and sends the result to the client (Send 32 bytes). After sending the result, it closes the socket and terminates.

The online calculator client program gets two arguments: server IP address and port number. When it starts, it creates a TCP socket and connects to the online calculator server. When the connection is accepted by the server, it reads the "expression.txt" file and gets the expression to be calculated. It sends the expression to the server (Send 1024 bytes) and receives the result from the server (Receive 32 bytes). Then it saves the result to the "result.txt" file and terminates.

3) Problem 3 – Simple DNS Simulator

The global DNS server program gets no argument. When it starts, it creates a TCP socket with port 9000, binds the socket, and listens for the local DNS server connection. When a local DNS server makes a connection, it accepts it and opens new TCP socket for data transmission. Because the server no longer needs to wait for another local DNS server, it closes the listen socket. After the socket for data transmission created, it receives the domain name from the local DNS server (Receive 128 bytes). Then it reads the "global_com_dns.txt" and the "global_edu_dns.txt" files to find corresponding IP address of the domain name. If it finds the IP address, it sends the IP address to the local DNS server (Send 16 bytes). If not, it sends "Not found" message to the local DNS server (Send 16 bytes). After that, it closes the socket and terminates.

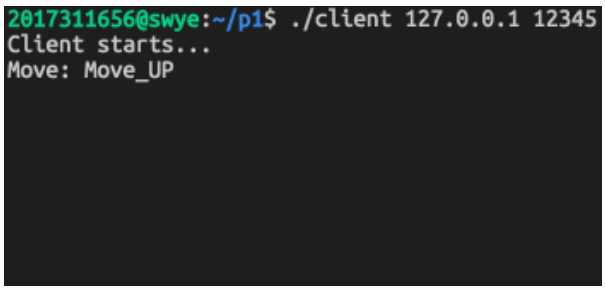
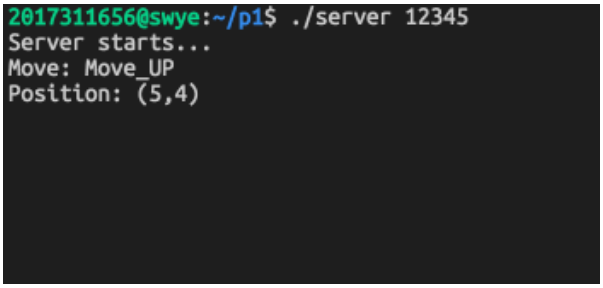
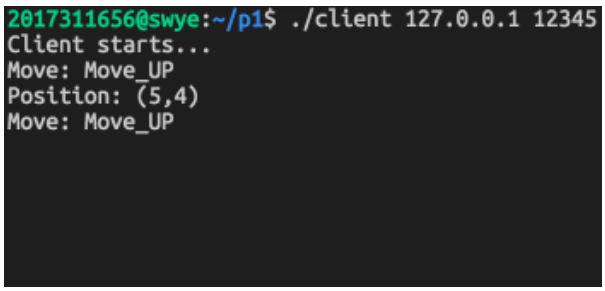
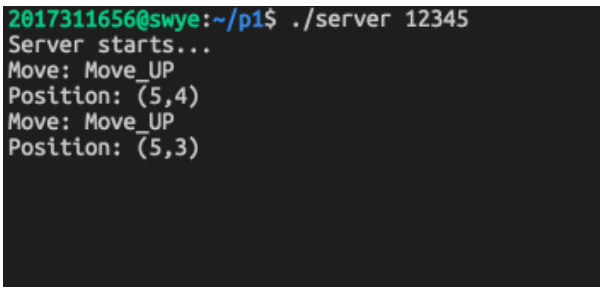
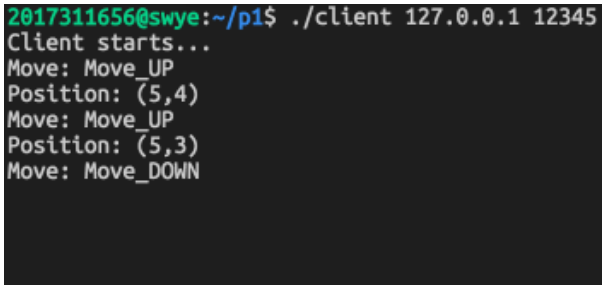
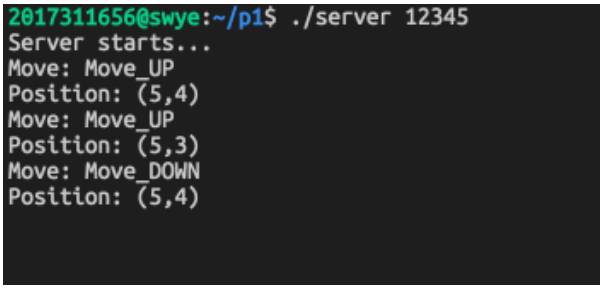
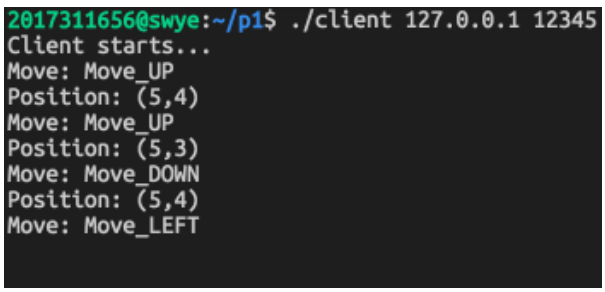
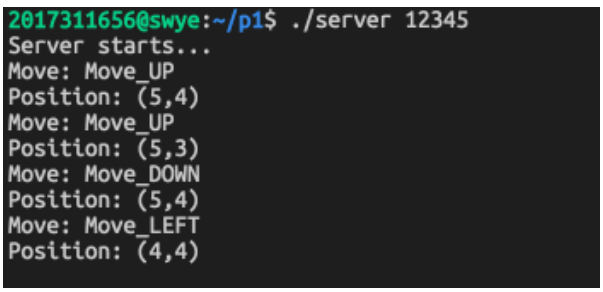
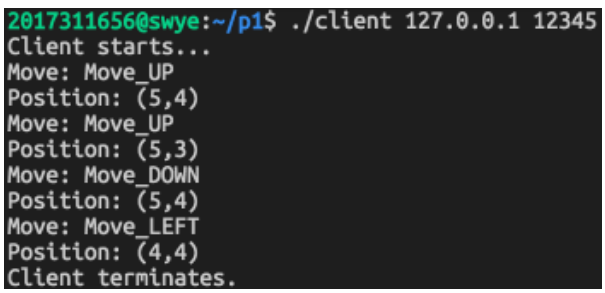
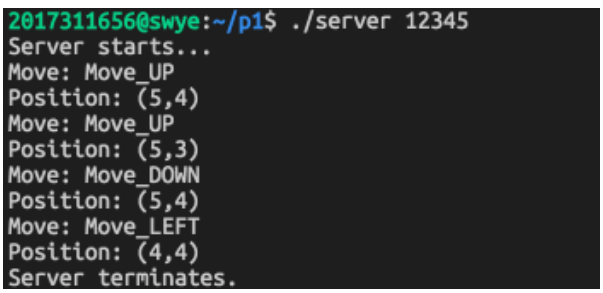
The local DNS server program gets no argument. When it starts, it creates a UDP socket with port 8000, and binds the socket. When a DNS client sends the domain name, it receives the domain name (Receive 128 bytes) and reads the "local_dns.txt" file to find corresponding IP address of the domain name. If it finds the IP address, it sends the IP address to the client (Send 16 bytes) and terminate. If not, it creates a TCP socket and connects to the global DNS server. When the connection is accepted by the global DNS server, it sends the domain name (Send 128 bytes) and receives the response from it (Receive 16 bytes). After that it closes the TCP socket, sends the response to the client (Send 16 bytes), and terminates.

The DNS client program gets no argument. When it starts, it creates a UDP socket. Then it reads the "domain.txt" file and gets the domain name to be queried. It sends the domain name to the local DNS server (Send 128 bytes) and receives the response from it (Receive 16 bytes). Then it saves the response to the "result.txt" file and terminates.

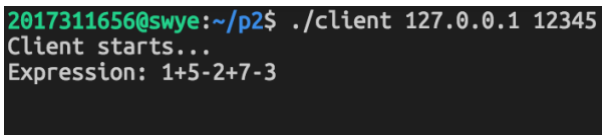
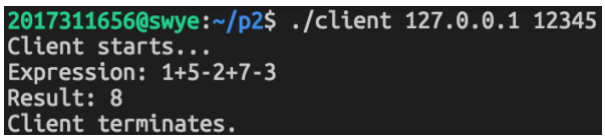
(Screen shots on next page)

6. Screen Shots

1) Problem 1 – Controlling Drones

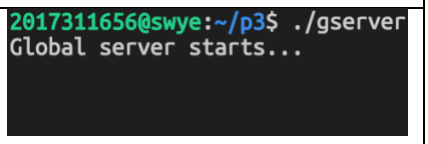
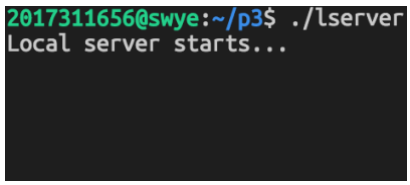
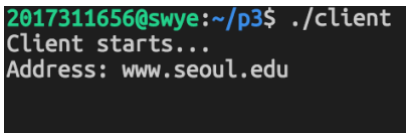
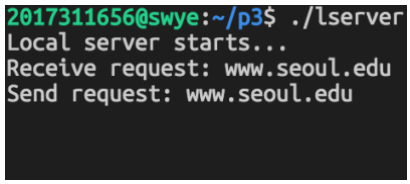
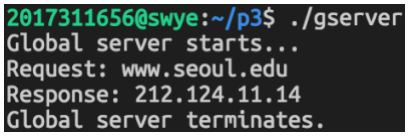
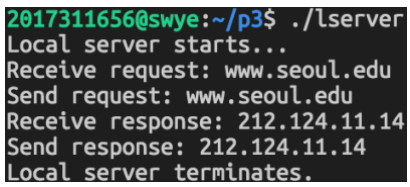
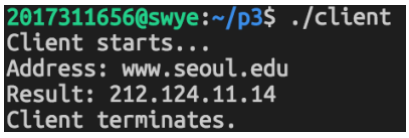
Drone Control Client	Drone Control Server
	
	
	
	
	
	

2) Problem 2 – Online Calculator

Online Calculator Client	Online Calculator Server
	
	
	

3) Problem 3 – Simple DNS Simulator

① DNS information is not in the local DNS server

DNS Client	Local DNS Server	Global DNS Server
		
		
		
		
		

②DNS information is in the local DNS server

DNS Client	Local DNS Server	Global DNS Server