프로젝트 정의서

과목명	네트워크 프로그래밍
학과(학부)	소프트웨어 학부
주제	실시간 소스코드 공유
팀원	20170334 주성진, 20170286 박세준

1. 프로젝트의 목표 및 필요성

프로그래밍을 공부하다 보면 여러 가지 문제상황에 직면하게 된다. 예를 들면, 어떤 에러가 발생했는데 해결하지 못하고 있거나 코딩을 하며 전혀 예상치 못한 결과를 마주하게 되는 경우들이 있을 것이다. 보통은 검색이나 다른 사람에게 질문을 통해 마주친 문제들을 해결하겠지만, 그렇지 못하는 경우도 있다. 어떻게 검색해야 할 지를 모르거나, 어떻게 질문해야 할 지를 모르는 경우 등이 그렇다. 이럴 땐 누군가가 직접 코드를 보며 도움을 받았으면 하는 생각이 들곤 한다. 이러한 상황에 놓일 때, 가장 적절한 해결책은 페어 프로그래밍을 하는 것이다.

페어 프로그래밍(Pair programming) 이란 다수의 사람이 하나의 소스 코드를 작성하는 프로그래밍 기법을 의미한다. 한 명은 드라이버의 역할을 맡아 실제 타이핑을 진행하고, 나머지는 네비게이터의 역할을 맡아 드라이버가 올바른 방향으로 코드를 작성할 수 있도록 돕는 것이다. 이러한 프로그래밍 방식의 장점은 지식 공유가 수월하다는 점이다. 예를 들어, 여러 명이서 알고리즘을 공부한다고 가정해보자. 누구는 분할정복에 관한 알고리즘만을 잘 알고 있고, 누구는 동적 계획 알고리즘만을 잘 알고 있는 등 개인의 역량에 많은 차이가 있을 것이다. 이런 상황에서 페어 프로그래밍을 진행하게 되면, 서로가 가진 지식을 공유하는 것과 동시에 개개인마다 자신이 가진 약점을 보완할 수 있다.

또 다른 상황으로 팀 단위의 개발을 진행한다고 가정해보자. 개발팀에 신입 멤버가 합류하면 해당 멤버는 팀에 적응할 시간이 필요하다. 문서나 구두를 통해 팀의 개발 환경이나 개발 스타일 등을 전달할 수도 있을 것이다. 그러나 "백문불여일견(百聞不如一見)" 이라는 말이 있듯이, 기존 멤버들에게 전해 듣는 것보다 페어 프로그래밍을 통해 실제로 경험해보는 것이 보다 더 빠르게 팀에 적응할 수 있을 것이다.

따라서 본 프로젝트에서는 페어 프로그래밍을 보다 수월하게 실천할 수 있도록 하는 것을 목표로 한다. 다수의 사용자가 하나의 서버에 접속하여 하나의 소스코드를 작성하는 프로그램을 만들 것이며, 작성한 소스 코드는 간단한 명령 입력을 통해 클라이언트 측에서 간단한 컴파일 과정을 거쳐 하나의 실행파일을 가질 수 있도록 한다.

2. 프로젝트의 특징 및 장점

본 프로젝트는 Linux 환경에서 동작하는 서버 – 클라이언트 구조의 프로그램이다. 또, 서버와 클라이언트는 스레드를 사용하여 주요한 기능들을 동작시키도록 하였다. 물론 스레드가 아닌 프로세스를 생성하고, 생성된 프로세스에서 작업하도록 할 수도 있었지만 스레드를 사용하는 것이 메모리를 조금 더 효율적으로 사용할 수 있으므로, 다중 스레드를 사용하여 기능들을 구현하였다.

또, 클라이언트에서는 ncurses 라는 라이브러리를 사용하여 분할 윈도우 화면을 구현하였다. ncurses 라이브러리는 CNU프로젝트에서 개발한 오픈소스 라이브러리로 X11 License를 따른다. 해당 라이브러리는 터미널 환경에서 화면을 window 라는 단위로 구분하여 각 window 별로 화면을 갱신하고 컨트롤 할 수 있다. 본 프로젝트에서는 해당 라이브러리를 사용해 소스코드가 표시되는 부분, 소스코드를 입력하는 부분을 개별의 window 로 나누었고 각각의 window가 서버와 클라이언트의 상황에 따라 개별로 갱신될 수 있도록 하였다.

본 프로젝트의 장점은 혼자서 소스 코드를 작성하는 것이 아니라 다수의 사람들과 함께 작성할 수 있다는 것이다. 또, 소스코드의 작성 및 컴파일 과정을 하나의 프로세스 내에서 모두 수행할 수 있다.

3. 개발 환경 및 관련 기술

Editor: Vscode

OS : Ubuntu Linux 20.04

Language : C (Standard C 17)

Complier : gcc 9.3.0

사용 기술 : Tcp/ip socket, thread, mmap, ncurses

ncurses 라이브러리를 사용하기 위해, 클라이언트 PC에는 libncurses5-dev 가 설치되어야 한다. 또, thread, mmap, ncurses와 관련된 라이브러리들은 컴파일 시에 자동으로 링크되는 것이 아니므로, 컴파일 시에 관련 옵션을 명시해주어야 한다.

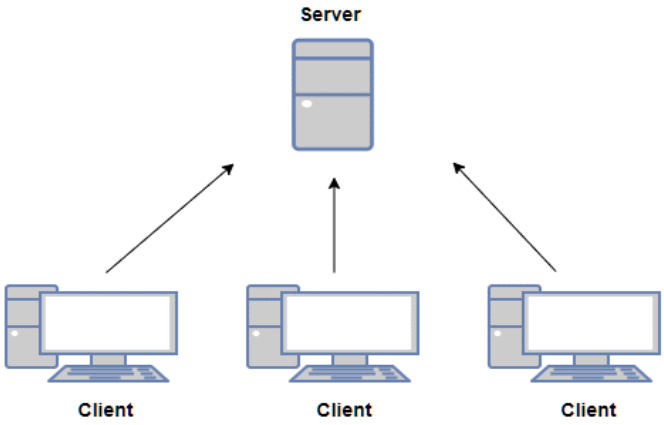
서버측 코드를 컴파일할 때에는 다음과 같은 형식으로 gcc 명령을 수행해야 한다. 각각 Posix thread와 shm_open(mmap을 열기 위한 함수)를 링크시키는 옵션이다.

gcc –o sample sampe.c -lpthread -lrt

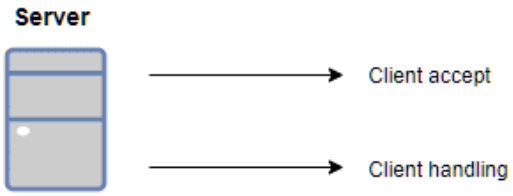
또, 클라이언트 코드를 컴파일 할 때에는 다음과 형식으로 gcc 명령을 수행해야 한다. -lptrhead와 –lcurses 옵션은 각각 Posix thread와 ncurses를 링크시키는 옵션이다.

gcc –o sample sample.c -lpthread -lcurses

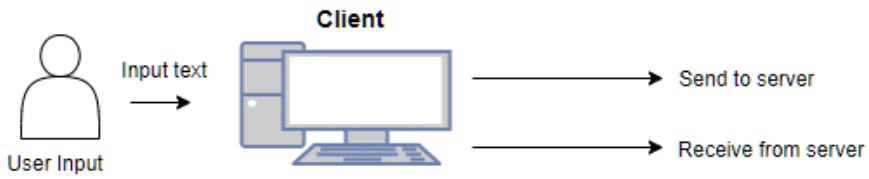
4. Architecture



서버 – 클라이언트 형식의 프로그램이다 보니, 다수의 클라이언트가 서버에 접근하고, 각 클라이언트의 요청을 서버가 처리하는 방식으로 전체 시스템이 동작한다.



서버가 수행하는 일은 크게 두 가지이다. 하나는 클라이언트의 연결 요청을 수락하는 것이고, 또 하나는 요청된 클라이언트를 하나의 스레드에 연결하여 클라이언트의 요청을 처리하게 하는 것이다.



클라이언트에서 수행하는 일도 크게는 두 가지이다. 이는 두 개의 스레드로 나누어 처리한다. 하나의 스레드는 사용자로부터 입력 받은 텍스트를 서버로 전송하는 것이고, 또 하나의 스레드는 서버로부터 전송 받은 텍스트를 콘솔 화면에 띄우는 것이다.

5. 수업과의 연관성

네트워크 프로그래밍 한 학기 수업동안 OSI 7 Layer, Socket 등과 같은 기초인 사항부터 TCP, UDP, ARP, HTTP,DDNS 와 같은 통신 프로토콜, 스레드, 프로세스, epoll, select 와 같은 멀티 프로세스, 멀티 스레드 및 semaphore, Mutex와 같은 동기화 기법들에 대해서 다루었다. 본 프로젝트에서는 상기한 요소들 가운데 socket, tcp, thread, mutex, semaphore를 활용하여 동시성이 보장되는 프로그램을 작성하였다. 서버에서는 1개의 클라이언트당 1개의 스레드를 할당하여 각각의 클라이언트들에 대한 동시성을 보장하였고, 클라이언트에서는 데이터를 수신하는 쓰레드와 데이터를 송신하는 쓰레드 및 그에 대한 작업 쓰레드를 나눠 사용자가 코드를 작성 중이더라도 서버로부터 데이터 전송이 오면 화면을 갱신 할 수 있도록 하였다. 서버와 클라이언트 간의 통신은 신뢰성이 보장되는 Tcp Sokect을 통해 이루어지고 Mutex와 semaphore를 통해 여러 스레드 간에 생길 수 있는 Race Condition을 방지하였다.