

프로젝트 명세서

멀티 스레드 채팅 서버 구현

목차

1. 프로젝트 개요.....	3
2. 과제	4
3. 심화 과제	6
4. 산출물 제출.....	7
5. 추천 학습 방법.....	9

1. 프로젝트 개요

4 차 산업 기술을 필두로 SW 관련 분야가 세간의 뜨거운 관심을 얻고 있습니다. 시대의 흐름을 읽고 SSAFY 에 지원한 여러분은 "역량 있는 SW 인재가 부족하다." 와 같은 기사를 마주한적이 있으실 겁니다. 또한 실제로 웹 기술을 기반으로 프로젝트를 진행하다 보면 다음과 같은 내용들을 접하게 됩니다. "자바스크립트는 싱글 스레드로 동작한다. Node.js 는 비동기 처리에 강한 장점을 가지며 단일 스레드로 작동한다. 스프링 웹 프레임워크는 멀티 스레드로 동작한다." 정확하게 어떤 의미인지 파악이 완전치 않더라도 어딘가에서 얻은 정보로 원하는 기능을 1 차 구현하는 데에는 큰 어려움이 없을 수 있습니다. 그렇지만 시스템이 복잡해지고 성능 이슈가 발생하는 등의 문제에 봉착하게 되면 내가 다루고 있던 모듈, 프레임워크를 구성하는 CS(Computer Science) 관련 기초 지식이 필요해지는 순간을 마주하게 됩니다.

CS 과목을 이수하지 않았더라도 일반적인 SW 개발자로서의 여정에서 큰 장애물이 없습니다. 작은 규모의 프로그램을 구현할 때에도 동작 과정을 하나하나 설명할 수 있도록 코드 한 줄 한 줄에 끊임 없는 관심만 쏟을 수 있다면 누구나 역량을 갖춘 SW 개발자가 될 수 있습니다.

단숨에 모든 CS 기술 요소를 풀어낼 수 있는 프로젝트를 만들기란 불가능에 가깝습니다. 운영체제의 기본 개념 및 프로세스, 스레드에 대한 정리와 더불어 네트워크에 대한 주요 지식을 정리 할 수 있다면 CS 첫 걸음으로는 최고의 선택이라 말할 수 있습니다. 채팅 서버 구현 프로젝트는 이런 핵심 요소들을 아우를 수 있는 훌륭한 주제가 될 수 있기에 적극 추천 드립니다.

2. 과제

1) 운영체제에 관한 학습 (프로세스와 스레드)

어떤 프로젝트를 진행하시더라도 운영체제에 대한 기본적인 이해는 필수입니다. 일반적으로 개발자는 궁극적으로 운영체제에게 작업 지시를 효율적으로 내려주는 일을 담당하기에 이에 대한 이해도가 부족하다면 많은 어려움을 겪을 수 밖에 없습니다. 운영체제의 핵심 기능에는 어떤 것들이 있는지? 우리가 평소 사용하는 운영체제에서 이런 핵심 요소들을 살펴볼 수 있는 방법은 무엇인지? 내가 사용하는 프로그래밍 언어는 어떤 메커니즘을 통해 운영체제에게 해야 할 일들을 지시하는지? 이러한 핵심 개념을 이해해야만 개발자로서 능동적인 시스템 설계와 구현이 가능하기 때문입니다.

운영체제가 관리하는 작업의 단위인 ‘프로세스’에 대한 이해는 매우 중요합니다. 일반적으로 개발자가 프로그래밍 언어를 이용해서 원하는 작업의 모음을 기술 한 이후 컴파일 과정을 거쳐 운영체제에게 이것을 실행하도록 명령을 내리면 ‘프로세스’를 생성하여 관리하게 됩니다. 내가 작성한 프로그램이 어떤 상태에 있는지 모니터링 하기 위해서라도 이에 대한 이해는 필수라 할 수 있습니다. 하나의 프로세스가 수행되는 동안에도 여러 개의 하위 작업들을 동시에 수행해야만 하는 경우가 존재합니다. 예를 들어 동영상 플레이어를 직접 만들어야 하는 경우 영상과 사운드 두 가지 자료를 동시에 디코딩해야 하는 과정이 필연적으로 요구되는데 이럴 때에 스레드를 활용한 적절한 자원 스케줄링을 통해 사람이 느끼기에는 동시에 해당 작업들이 이루어지는 듯한 효과를 낼 수 있습니다.

아래의 참고 링크 뿐만 아니라 스스로 프로세스 및 스레드의 개념에 대해 설명 할 수 있을 때까지 집중력을 가지고 학습하셨으면 합니다.

- 프로세스에 대한 이해

<https://ko.wikipedia.org/wiki/%ED%94%84%EB%A1%9C%EC%84%B8%EC%8A%A4>

- 스레드에 대한 이해

[https://ko.wikipedia.org/wiki/%EC%8A%A4%EB%A0%88%EB%93%9C_\(%EC%BB%B4%ED%93%A8%ED%8C%85\)](https://ko.wikipedia.org/wiki/%EC%8A%A4%EB%A0%88%EB%93%9C_(%EC%BB%B4%ED%93%A8%ED%8C%85))

2) 네트워크 기본 개념과 socket 통신에 대한 이해

네트워크가 없는 격리된 시스템은 현대 사회에서 찾아보기 어렵습니다. 컴퓨터 네트워크는 모든 개발자에게 선택이 아닌 필수 소양이며 이에 대한 기본 개념은 주니어 개발자에게도 예외일 수 없습니다. 계층적으로 분리해 놓은 OSI 7 계층을 시작으로 간 단계에서 정의하고 있는 내용들은 어떤 것들인지 실제 장치간 통신은 어떤 식으로 전개되는지 스스로 설명이 가능하도록 집중적으로 학습하도록 합니다. 더불어 채팅서버 구현에 사용될 socket 통신이 어떤 개념인지 그리고 운영체제에서는 통신 관련된 내용을 어디까지 커버하는지도 확인하시기 바랍니다.

3) 실시간 멀티 채팅 서버 프로그램 구현

최대 5명이 동시에 접속하여 사용자간 단순 텍스트 교환을 통한 채팅 기능이 가능하도록 서버를 구현합니다. 처음 단계에서는 책, 참고할 만한 사이트 등을 통해 실행이 가능하게 만드는데 집중하고 1차 기능 구현에 성공하였다면 소스 코드 한 줄이 주는 의미가 무엇인지 설명이 가능하도록 정리해 나갑니다.

4) 채팅 클라이언트 프로그램 구현

별도의 GUI 없이 커맨드창을 통해 입력과 출력이 이루어지도록 클라이언트 프로그램을 구현합니다.

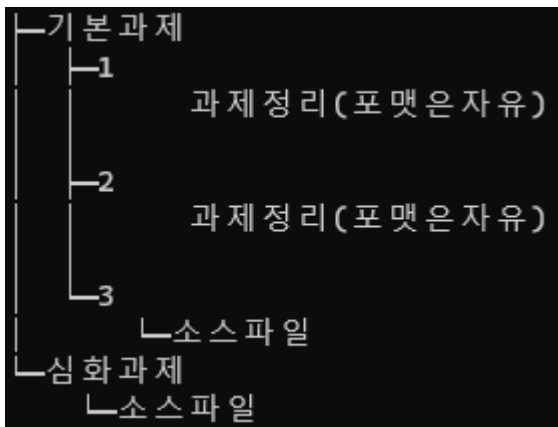
3. 심화 과제

- 1) ‘커맨드 창’에서의 채팅이 아닌 GUI 환경을 지원하는 채팅 클라이언트 프로그램을 구현합니다. 내가 선택한 프로그래밍 언어, 프레임워크에서 인기가 있는 GUI 모듈을 빠르게 살펴보고 적용해 봅니다.

4. 산출물 제출

- <https://lab.ssafy.com/s10-study/self-project/>의 “산출물 제출 가이드”.docx 참조
- 작성 내용

아래의 트리처럼 과제 번호에 맞는 폴더에 문서 및 소스파일을 첨부하고 전체를 압축하여 제출해주세요.



1) 운영체제 내에서의 프로세스와 스레드에 대한 학습 내용 정리

운영체제의 핵심 요소 중 하나인 프로세스에 대해 학습합니다. 그리고 프로세스 내 여러 명의 작업자라 불릴 수 있는 스레드에 대해 학습하고 이해한 내용을 정리하도록 합니다.

2) 네트워크에 대한 학습 내용 정리

OSI 7 계층을 필두로 네트워크에 대해 학습한 내용(소켓 통신에 관한 정리는 필수)을 정리합니다. 분량 제한은 없으나 내가 이해한 내용을 정리하도록 합니다.

3) 실시간 멀티 채팅 서버, 클라이언트 프로그램 구현

익숙한 프로그래밍 언어를 활용하여 다음 요건을 만족하는 채팅 프로그램을 구현합니다.

- 커맨드 창을 활용한 단순 텍스트 전송 (GUI 필수 아님, socket 통신)
- 최대 5명 동시 접속 가능하며 채팅 룸은 하나만 존재

4) 심화 과제(GUI 프로그래밍)

앞선 3 번의 결과물을 기반으로 GUI 환경을 지원하는 채팅 클라이언트 프로그램을 구현합니다

5. 추천 학습 방법

- 1) 모든 프로젝트가 그러하듯 정해진 기한 내에 기획한 서비스가 온전히 수행되는 것이 우선되어야 합니다. 책, 블로그, 공개 소스 등 참고할 만한 자료를 활용하여 기본적인 채팅 서버, 클라이언트 프로그램을 구현합니다. 인터넷 검색만으로도 실행 가능한 샘플 코드를 쉽게 얻을 수 있기에 1차적인 완성에는 큰 어려움이 없을 것입니다.
- 2) 본 프로젝트의 진정한 목적은 1)번 이후, 즉 간단한 채팅 프로그램 완성 이후에 있습니다. 작성한 코드 각 줄마다 어떤 명령을 운영체제에게 주고 있는 것인지 스스로 설명이 가능하도록 학습하고 정리 해봅니다.
 - 네트워크 개념 정리, OSI 7 계층, socket 통신
 - 프로세스와 스레드