

# System Programming Project 2

담당 교수 : 김영재

이름 : 이제현

학번 : 20170624

## 1. 개발 목표

여러 client들이 동시에 접속하는 concurrent한 server를 event driven 방식과 thread-based 방식으로 구현한다.

## 2. 개발 범위 및 내용

### A. 개발 범위

- 아래 항목을 구현했을 때의 결과를 간략히 서술

### 1. Task 1: Event-driven Approach

여러 client가 동시에 실행해도 concurrency하게 실행된다.

### 2. Task 2: Thread-based Approach

event driven과 마찬가지로 concurrency 하지만, client의 수의 따라 시간이 일정하게 늘지 않고 급수적으로 증가한다.

### 3. Task 3: Performance Evaluation

## B. 개발 내용

- 아래 항목의 내용만 서술
- (기타 내용은 서술하지 않아도 됨. 코드 복사 붙여 넣기 금지)
- **Task1 (Event-driven Approach with select())**

- ✓ Multi-client 요청에 따른 I/O Multiplexing 설명

I/O Multiplexing은 single process에서 echo server가 동시에 여러 client들의 입력을 받아야 하기 때문에 필요하다. 예를 들어, 동시에 한명은 네트워크 연결을 시도하고, 다른 한명은 명령어를 입력하고 있다면 서버가 각각의 client 들에 동시에 출력을 못하기 때문에 필요함 .

- ✓ epoll과의 차이점 서술

epoll은 file descriptor에 대해 반복문을 사용하지 않는데 select는 모든 fd들을 순회하는 반복문을 사용한다.

- **Task2 (Thread-based Approach with pthread)**

- ✓ Master Thread의 Connection 관리

client와의 연결을 통한 fd를 저장하기 위해 sbuf\_t 형식인 sbuf를 통해 fd를 삽입한다.

- ✓ Worker Thread Pool 관리하는 부분에 대해 서술

- **Task3 (Performance Evaluation)**

- ✓ 얻고자 하는 metric 정의, 그렇게 정한 이유, 측정 방법 서술
- ✓ Configuration 변화에 따른 예상 결과 서술

## C. 개발 방법

### -Event-driven Approach with select

기본적으로 강의자료를 참고하되, stock 정보를 저장하는 자료구조는 Item이라는 type을 가지는 구조체로 선언하였다. 이것을 이진 트리 형식으로 정보를 가져올 수 있도록 전체적인 구조를 짰다. 기본적인 이진 트리의 node들의 삽입, 검색, 삭제, 출력의 기능을 만들었고 client가 입력하는 show, buy, sell 명령어는 이 기능들을 활용하여 요청을 수행한다.

### -Thread-based Approach with pthread

전체적인 구조는 event-driven 방식과 유사하지만 mutex의 변경을 다른 사용자가 못하도록, exclusion을 위해 내용이 수정되는 부분에 P,V operation을 기능들에 삽입했다.

## 3. 구현 결과

thread based가 제대로 구현되지 않았지만 client의 요청을 받는 server.c의 main 함수 쪽을 손보면 될것 같다.

## 4. 성능 평가 결과 (Task 3)

- 강의자료 슬라이드의 내용 참고하여 작성 (측정 시점, 출력 결과 값 캡처 포함)