**Multicore Programming Project 2**

담당 교수 : 박성용

이름 : 백상욱

학번 : 20190388

1. **개발 목표**이번 프로젝트는 여러 클라이언트들이 접속을 하고 server에 요청을 보냈을 때, 이 동시 접속 및 요청에 대해서 하나의 서버가 처리하는 concurrent서버를 구축하는 것이 목표이다. 이를 구현하기 위해 소켓 프로그래밍, 에코 서버 프로그래밍, Rio에 대한 내용을 활용한다. 서버는 주식 정보를 이진 탐색 트리 형태로 저장하여 입력 및 읽으며 여러 클라이언트들의 조회, 판매, 구매 요청을 처리한다. 그리고 이를 event-driven 방식과 thread-based 방식으로 각각 구현한다.   
   최종적으로는 구현한 두 서버 각각의 성능을 평가하고 분석한다.
2. **개발 범위 및 내용**
   1. **개발 범위**

* **아래 항목을 구현했을 때의 결과를 간략히 서술**

1. Task 1: Event-driven Approach : I/O multiplexing을 기반으로, select 함수를 이용하여 구현한다. 서버 시작과 함께 stock.txt의 내용을 읽고 이진트리에 저장 후 클라이언트가 연결 종료를 요청할 때까지 주식 조회, 매매 요청을 처리한다. 이를 위해 각각의 클라이언트는 fd를 트리거하고, 서버는 select로 fd를 모니터링하여 동작을 시작시킨다. 서버 종료시 업데이트된 최 종 변경 사항을 stock.txt에 저장한다.
2. Task 2: Thread-based Approach
3. Task 3: Performance Evaluation
   1. **개발 내용**

* **Task1 (Event-driven Approach with select())**
  + Multi-client 요청에 따른 I/O Multiplexing 설명
  + epoll과의 차이점 서술
* **Task2 (Thread-based Approach with pthread)**
  + Master Thread의 Connection 관리
  + 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
  1. **개발 방법**
* **B.의 개발 내용을 구현하기 위해 어느 소스코드에 어떤 요소를 추가 또는 수정할 것인지 설명. (함수, 구조체 등의 구현이나 수정을 서술)**

1. **구현 결과**

* **2번의 구현 결과를 간략하게 작성**
* **미처 구현하지 못한 부분에 대해선 디자인에 대한 내용도 추가**

1. **성능 평가 결과 (Task 3)**

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