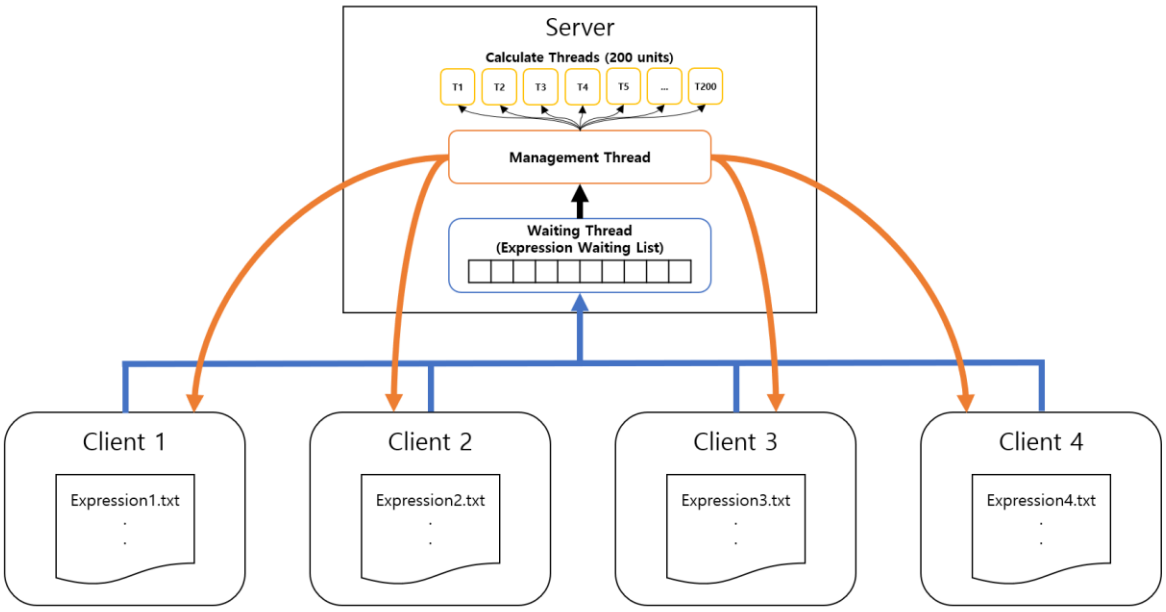
HW#3 데이터통신 프로그래밍 과제

Thread Pool 기반의 사칙연산을 위한 서버-클라이언트 간 작업 스케쥴러 구현

○ DUE: 2024.10.30.(수) 23:59,	(Assigned 2024.10.16)
○ How: 1개화일 제출 – G조이름HW3.zip (조별로 한명만 제출하면 됨)	
◆ 모든 소스화일 및 결과화일 전체를 압축한 G조이름HW3.zip으로 제출 ex) G1HW3.zip	

1. (시뮬레이션 환경 및 구성요소) 다음과 같은 실험환경을 구성한다. 1개의 Server와 4개의 Client는 thread간 Socket 통신으로 동작하며 연결 구조의 제약은 없다. 각각의 객체는 다음의 기능을 수행한다.



① 서버: 전역변수 혹은 임의의 객체인 System Clock을 유지, 관리, 업데이트 하고 프로그램이 수행되면 System Clock을 수행시킨다. 프로그램이 시작되면 서버는 202개의 thread를 생성하여 관리하게된다. 이때 thread는 1개의 "Waiting Thread", 1개의 "Management Thread", 200개의 "Calculate Thread"이다. "Waiting Thread"는 클라이언트에서 요청된 연산 작업을 해당 thread 내에서 통신으로 전달받는 역할을 하며, 작업을 최대 30개까지 리스트에 저장이 가능하다. 만약 리스트가 가득 차있다면 작업 거절 메세지를 클라이언트에 전송한다. "Management Thread"는 "Waiting Thread"의 리스트에 대기중인 작업을 하나 선택하여 "Calculate Thread" 하나에 작업을 전달하고 최종 연산 된 결과를 다시 해당 thread 내에서 직접 클라이언트와의 통신으로 전달하는 역할을 한다. 리스트에 대기 중인 작업 중 어떤 작업을 우선선택 할 것인지에 대한 제약은 없으며, 구현한 방식이나 알

고리즘에 대해 명시하여야한다. 200개의 "Calcuate Thread"는 각각 "Management Thread"에서 전달받은 연산 작업을 직접 수행한다. 이때 주어진 수식을 Parsing Tree로 변환하여 출력 후 주어진 수식을 postorder traversal 방식으로 계산하고 결과를 출력한다. 그 후 최종 결과는 다시 "Management Thread"로 전달한다. "Calcuate Thread"의 연산 수행 시간은 Parsing Tree로 변환된 수식의 leaf node 개수 msec 만큼 소요된다. 예를 들어 수식 "2-5*2"의 연산 시간은 그 leaf node의 개수인 3msec 소요된다. 서버내의 모든 thread간 통신은 mutex 또는 semaphor를 사용하여 전역변수가 아닌 heap 영역에 존재하는 공유 메모리를 통해 이루어질수 있도록 한다(제출 시 쓰레드간 통신을 구현한 규칙 및 알고리즘에 대해 명시하여야 한다). 모든 thread의 이벤트와 주어진 수식의 Parsing Tree, 각 수식의 연산 소요 시간을 System Clock과 함께 log로 기록한다.

② 클라이언트: 클라이언트는 각각 Expression 텍스트 파일에서 1000개씩 주어진 문제를 하나씩 순차적으로 서버의 "Waiting Thread"에게 전달하여 작업을 요청하게 된다. 그 후 연산에 대한 결과는 "Management Thread"에서 결과에 대한 정보를 전달받는다. 이전 작업의 결과가 도착하지 않아도 클라이언트는 서버에게 System Clock 1msec 간격으로 계속하여 작업을 요청해야 한다. 만약 서버에서 작업 거절 메세지가 도착한다면 도착 즉시 해당 연산 작업을 최우선순위로 해결해야 한다. 모든 작업을 요청 후, 모든 연산 결과가 도착한다면 각 클라이언트는 모든 이벤트와 연산 작업에 따른 연산 결과를 System Clock과 함께 log로 기록하고, 각 작업의 요청부터 결과가 도착할때까지의 평균 대기 시간, 작업이 거절된 횟수를 log로 기록한다.

※ Server는 반드시 AWS나 구글클라우드 등 Physically 외부서버에 구현 되어야함!!
※ Server와 모든 Client의 통신은 반드시 Socket으로 구현되어야 함!!
※ 모든 구성요소는 Thread를 통해 구현되어야 함!!
(해당 세가지 조건은 **과제수행의 전제조건**으로 만족되지 않으면 과제수행 인정불가)

2. (시뮬레이션 시나리오) 다음과 같은 실험 시나리오를 수행하여 각각의 구성요소 별 Log를 기록한다. 즉, 서버와 4개의 클라이언트는 각각 모든 이벤트와 연산 정보를 Server.txt, Client1.txt, Client2.txt, Client3.txt, Client4.txt에 기록한다.

① 시나리오: 프로그램 시작 후, 서버는 모든 클라이언트와의 통신 환경을 초기화 후 가상의 system clock을 생성합니다. System clock은 서버에서 관리하며, 실제 시간이 아닌 operation에서 소요되는 가상의 연산 시간과 가상의 network delay를 측정한다. 서버는 1개의 "Waiting Thread", 1개의 "Management Thread", 200개의 "Calculate Thread"를 생성

하여 관리한다. 모든 사전작업이 끝난다면 System Clock의 시작과 함께 각각의 클라이언트는 주어진 Expression 파일에서 1 line씩 읽어, 서버에게 작업을 1 msec 간격으로 요청한다. 서버는 클라이언트에서 요청된 수식을 "Waiting Thread"의 리스트에 저장이 가능하며, "Management Thread"는 리스트에서 하나의 작업을 선정 후 "Calcuate Thread"에 할당하여 연산 작업을 수행한다. 이때 어떤 작업을 우선으로 수행할지에 대한 제약은 없으며, 클라이언트의 작업 대기시간이 최소가 되며, 최대한 모든 작업을 수용할 수 있는 방식을 설계하여 제출 시 구현된 규칙이나 알고리즘에 대해 명시하여야 한다. "Calcuate Thread"는 할당 받은 수식을 Parsing Tree로 변환하여 postorder traversal 방식으로 연산한다. 이때 연산에 소요된 시간은 tree의 leaf node 개수(msec) 만큼 소요된다. 수식의 연산이 끝나게 된다면 해당 수식의 결과는 다시 "Management Thread"로 전달되며 해당 thread에서 클라이언트에게 결과를 전송한다. 만약 클라이언트의 작업 요청이 있을 때, "Waiting Thread"의 리스트에 공간이 없다면 작업 거절 메세지를 클라이언트에 전송하며 클라이언트는 다음 요청에서 거절된 작업을 최우선 순위로 재요청 하게 된다. 모든 작업을 요청 후 모든 연산 결과가 도착하였다면, 모든 클라이언트와의 연결을 종료한다(모든 node는 Gracefully Termination 되어야 함). 프로그램 수행동안 서버는 모든 이벤트(thread의 업데이트, Parsing Tree와 postorder traversal 순서를 포함한 개별 연산 정보, 각 수식의 연산 소요 시간)를 System Clock과 함께 log로 기록 하며, 모든 클라이언트는 모든 이벤트(각 수식에 따른 결과와 대기시간)을 System Clock과 함께 log로 기록한다. 모든 클라이언트는 log의 마지막에 모든 작업의 요청부터 결과를 받을때까지의 평균 대기시간과 작업이 거절된 횟수와 파일 내의 작업이 끝날때까지 소요된 시간을 출력한다.

3. (과제제출) 다음 화일을 조별로 기한안에 제출한다.

   - G조이름HW3.zip (ex. G1HW3.zip)
     - 모든 소스 화일들
     - AllDefinedLogs.txt
       - 조별로 프로그램에서 정의한 모든 Log 메시지 명세 및 설명
     - 서버 및 클라이언트별 출력된 모든 Log 화일들
     - download.txt
       - G조이름HW3.mp4 (ex. G1HW3.mp4) : 5분이내의 설명 동영상을 제작하여, 해당동영상을 다운로드 할 수 있는 link를 포함하는 화일

- 반드시 동영상이 다운되는 링크를 삽입할 것
- 포함된 링크로 동영상 다운이 안되거나, 공유권한이 없거나, 다운 후 동영상 실행이 안되거나 하는 등의 에러는 프로그램 수행이나 포함내용의 미포함처럼 모두 해당조의 과실로 감점대상임

◆ Readme.txt

· 조 이름, 모든 조원 학번&이름
  - 학생 별 역할 명시
· 프로그램 구성요소 설명
· 소스코드 컴파일방법 명시
· 프로그램 실행환경 및 실행방법 설명
· 서버의 thread 관리 및 작업 대기 리스트의 선정 알고리즘에 대한 설명 작성
· Error or Additional Message Handling에 대한 사항 설명
· Additional Comments: 추가로 과제제출관련 언급할 내용 작성