|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **주차** | 16주차 | **기간** | 2023.10.10~  2023.10.16 | **지도교수** | (서명) |
| 이번주 한일 요약 | 미뤄왔던 게임 서버 프로그래밍 책을 다시 읽기 시작함.(4 ~ 9장) | | | | |

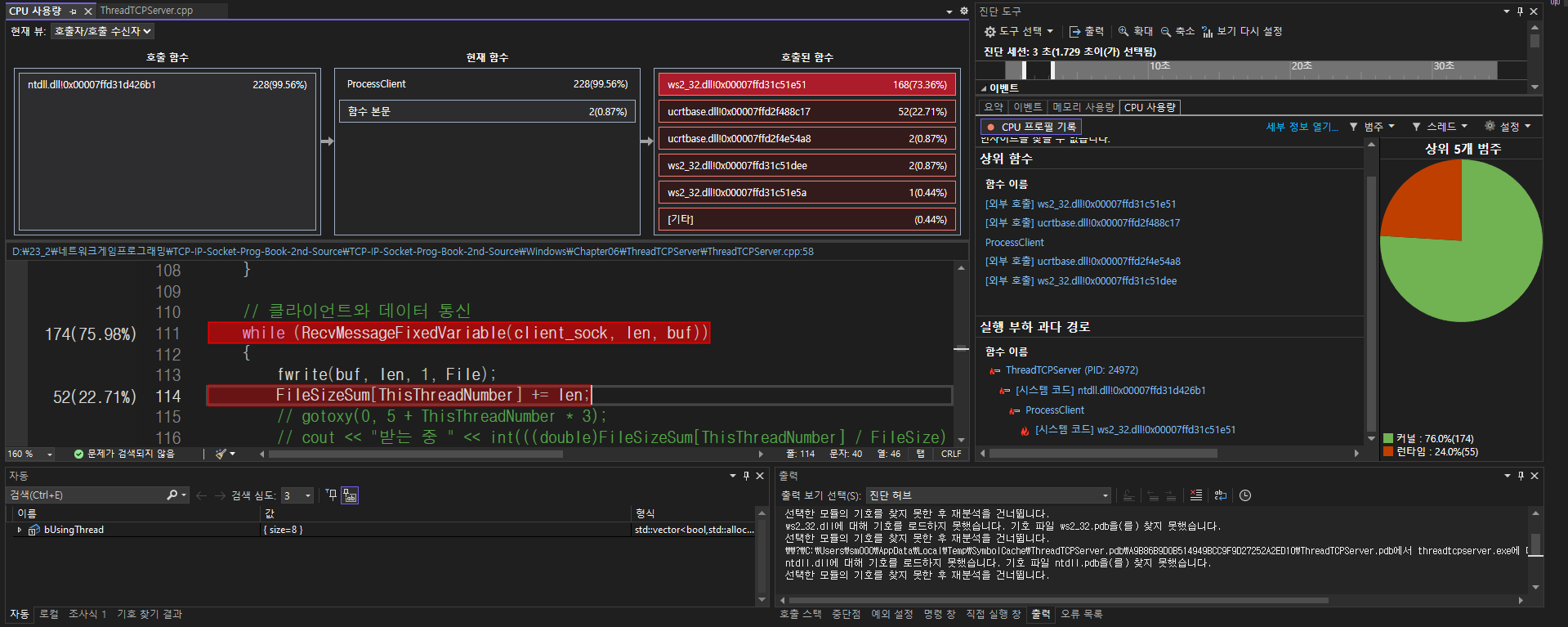
<상세 수행내용>

게임 프로그래밍 교과서 책을 읽으며 정리해봄

4장 게임 서버와 클라이언트 부터해서 9장 분산 서버 구조까지 책의 절반 이상을 읽음.

결국은 이론 덩어리기 때문에 간략하게 알면 좋을것들을 적어봄.

코드 프로파일링: Visual studio에서 지원하는 기능으로 어느 범위 내에서 함수가 호출되는 빈도를 알 수 있는 기능. 모든 프로그램 성능의 80%는 20%의 코드에서부터 나온다는 80:20 법칙에 따라 20% 코드는 성능 최적화를 하고 나머지 80퍼는 유지보수하기 쉽게 짤 것을 책에서는 추천한다. 또한 서버 분산 처리는 어느 부분에서 부하가 오는지 알아내어 분산한다. 그렇기에 코드 프로파일링 기능으로 어느 부분을 최적화할지, 어느 로직부분을 분산할지 결정하는데 도움이 될 거 같음.



게임 서버 엔진에서 쓰는 기능

RMI(원격 함수 호출): 상대방 컴퓨터 안에 있는 프로그램의 특정 함수를 멀리서 실행. 수동으로 만들었어야 하는 코드를 자동으로 만들어 줌.

P2P 그룹: P2P 그룹을 만들어 그 그룹내에 있는 클라끼리만 통신 가능(서버에서 처리). 클라이언트를 해킹하여 아무 클라에 P2P 연결을 맺고 잘못된 데이터를 전송하는 것을 막기위함.

게임플레이 네트워킹 방식은 총 4가지다

1. 모든 역할을 서버에서 하기: 서버 부하로 인해 지금은 안쓰는 추세
2. 렌더링은 클라에서 하기: 메시지를 통해 서버 동기화

이는 레이턴시가 1/60보다 현저히 낮아야하기에 월드 변화를 1/30이나 1/10으로 낮춰서 변화값을 보간하는 방식을 씀

1. 추측항법: 서버와 클라 사이의 레이턴시를 이용하여 캐릭터의 위치를 추측하는 방법.
2. 레이턴시 마스킹: ‘사소한것들은 클라에서 판단하기’와 ‘일단 보여주고 나중에 얼렁뚱땅하기’ 방법이 있음.

사소한것들은 클라에서 판단하기: 매우 쾌적함. 캐릭터 위치 계산을 클라에서하고 서버에 결과 전송. 하지만 해킹 문제가 존재

일단 보여주고 나중에 얼렁뚱땅하기: 클라에서 서버에 메시지를 보내면서 그 행동에 맞는 일부 모습을 클라에서 시작. 그 후 서버에서 다시 메시지가 오면 연출해야하는 나머지 부분을 클라에서 보여주는 방식. <https://www.youtube.com/shorts/g_ygEYTYNHM>

진행 상황 유튜브 링크:

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** |  | | |
| **해결방안** |  | | |
| **다음주차** | 17주차 | **다음기간** | 2023.10.17~2023.10.23 |
| **다음주 할일** | 게임 서버 프로그래밍 교과서 책 마지막 10장 읽기  프로토타입 게임 외부에서 접속 가능하게 해결하기  시험공부 살짝하기 | | |
| **지도 교수**  **Comment** |  | | |