|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **주차** | 11 주차 | **기간** | 9.5~ 9.12 | **지도교수** | (서명) |
| 이번주 한일 요약 | 1. 네트워크 프로그래밍 책 공부 2. 게임서버 강의 시청을 통한 공부 | | | | |

<상세 수행내용>

1. 네트워크 게임 프로그래밍 교재를 통해 소켓 모델과 그 중 completion port 모델의 동작 원리 등을 공부 후 실습을 진행해 봤습니다.

텍스트, 스크린샷, 도표, 폰트이(가) 표시된 사진

자동 생성된 설명

Completion port 모델의 실행 순서를 바탕으로 비동기 입출력 모델에 대해 이해해 보고자 했습니다. 우선 CreateIoCompletionPort 함수를 통해 입출력 완료 포트를 생성할 수 있었으며, 소켓과 입출력 완료 포트를 연결해야 했습니다.

비동기 입출력에 대해 처리방법 등에 대해서는 GetQueuedCompletionStatus 함수를 사용해 입출력 처리를 할 수 있었습니다.

1. 게임 서버 온라인 강의를 시청했습니다. 지난 주 멀티스레드 프로그래밍 실습을 통해 생긴 부족한 부분들을 공부했습니다. 공유자원을 수정하기 시작했을 때의 문제들을 해결하는 연습을 했습니다. atomic을 통해 한 스레드가 공유자원에 접근했을 시 다른 스레드의 접근을 제한하는 방법 및 락을 통해 접근을 제한하는 방식에 대해 실습해봤으며, 락을 통해 사용할 때 발생할 수 있는 데드락의 문제점을 해결하는 실습을 해봤습니다. 추가적으로 스핀락을 구현해봤으며, 이 경우 CAS연산을 활용해봤습니다. 이를 활용해 Reader\_Writer 스핀락을 구현해봤습니다. 이 외에도 스핀락 방식이 아닌 이벤트와 컨디션 변수(notify\_one / wait)로 스레드간 통신 동기화를 실습해봤습니다.

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** | Iocp 모델에 대해 구조적으로 어떤식으로 실행되는지에 대한 이해만 있을 뿐 아직 미비한 것 같다. | | |
| **해결방안** | 위와 마찬가지로 부족한 부분은 차후 강의를 통해 채워나가도록 하겠다.. | | |
| **다음주차** | 12주차 | **다음기간** | 9.13 ~ 9.19 |
| **다음주 할일** | 1. 게임 서버 온라인 강의 시청(메모리 관리 +@ 프로토콜 연동)  2. 게임서버 교과서 책을 통해 iocp 데이터베이스 부분 공부 | | |
| **지도 교수**  **Comment** |  | | |