

네트워크 프로그래밍 프로젝트 평가 항목

항목	점수	설명
네트워크 기술의 활용 및 난이도	20	<ul style="list-style-type: none"> ● 클라이언트-서버 구조, TCP/UDP 등 네트워크 개념이 잘 적용되었는가? ● 멀티스레딩, 논블로킹 I/O, 비동기 통신 등 고급 기술을 적절히 활용했는가? ● 도전적인 기술을 적용하려 하였는가?
기술적 완성도	20	<ul style="list-style-type: none"> ● 스스로 설정한 구현 목표를 얼마나 달성했는가? ● 기능들이 정상적으로 작동하며 일관성 있게 구현되었는가? ● 예외 상황에서도 안정적으로 동작하는가?
성능 검증 노력	20	<ul style="list-style-type: none"> ● 정상 및 예외 상황을 고려한 테스트가 잘 구성되었는가? ● 여러 클라이언트 연결, 지연 등 실제 네트워크 환경을 고려한 실험이 이루어졌는가?
구조적 코드 작성	10	<ul style="list-style-type: none"> ● 함수 분리, 일관된 네이밍, 주석 등 유지보수 가능하고 읽기 쉬운 코드를 작성했는가?
기술 문서 작성	10	<ul style="list-style-type: none"> ● 실행 방법, 주요 기능 설명, 사용 기술 요약 등이 문서로 잘 정리되어 있는가?
발표 및 시연	20	<ul style="list-style-type: none"> ● 프로젝트 내용을 명확하게 설명하고 질문에 논리적으로 답변했는가? ● 구현한 기능의 작동 과정을 테스트 계획에 따라 효과적으로 시연했는가? ● 다른 평가 항목에서 강조된 기술적 강점을 발표를 통해 잘 전달했는가?
추가 점수	5	<ul style="list-style-type: none"> ● Git을 사용한 개발 이력 관리를 충실히 수행했는가? ● 이미 구현된 부분에 대해서는 평가기준 이후 변경된 사항에 대해서만 관리

발표 시 반드시 포함되어야 할 발표 항목

1. 프로젝트 개요

- 프로젝트의 목적 및 배경
- 주요 기능 요약

2. 사용한 네트워크 기술 설명

- 클라이언트-서버 구조 또는 기타 통신 모델 설명
- TCP/UDP, 멀티스레딩, 논블로킹 I/O, 비동기 처리 등 사용 기술
- 도전적인 기술 선택 이유와 그 구현 방식

3. 기능 구현 설명

- 처음 설정한 목표와 실제 구현된 기능 비교
- 주요 기능 시연
- 예외 상황 처리 방법 (예: 잘못된 요청, 연결 끊김 등)

4. 테스트 및 성능 검증

- 테스트 시나리오 개요 (정상 + 예외 상황)
- 성능 검증을 위해 수행한 실험
 - 다수 클라이언트 접속 테스트
 - 네트워크 지연/속도 측정
 - 부하 발생 시 동작 설명
- 테스트 시나리오(성능 검증 포함) 수행 결과

5. 코드 구조 소개

- 전체 코드 구성도 또는 주요 모듈 구조
- 함수 분리 방식, 클래스/모듈 간 관계
- 가독성/유지보수성 확보를 위해 고려한 요소 (네이밍, 주석 등)

6. 기술 문서 요약

- 문서로 정리한 내용 중 핵심 요약 (슬라이드에 간단히 정리)
- 문서 위치(GitHub 링크 등) 안내

7. 개발 관리 및 Git 사용 이력 (→ 관련 평가: 추가 점수)

- Git을 통한 버전 관리 흐름 (commit log, 브랜치 전략 등)
- 개인/팀의 작업 내역 추적 예시