

시스템/소켓 프로그래밍 기반 지능형 실시간 명령 응답 시스템 개발 계획서

2021115744_권구태

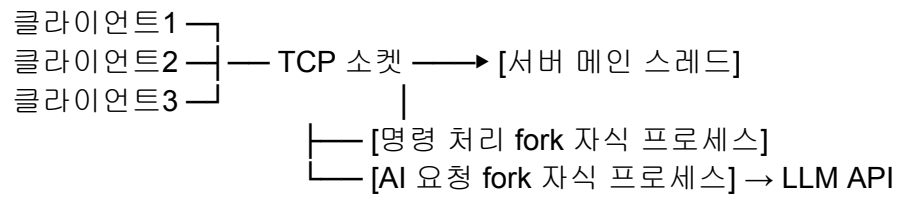
1. 과제 개요

본 프로젝트는 시스템 프로그래밍과 네트워크 소켓 프로그래밍, 그리고 현대적인 LLM API를 통합하여, 다중 프로세스 기반의 실시간 명령어 응답 시스템을 개발하는 것이다. 사용자는 키보드를 통해 명령을 입력하고, 이 명령은 서버에 전송되어 처리된다. 서버는 명령을 해석한 뒤 실행 결과 또는 AI의 응답을 클라이언트에 다시 반환한다.

2. 주요 목표 기능

분야	기능 설명
시스템 프로그래밍	<ul style="list-style-type: none">- <code>fork()</code>를 이용해 서버 내 명령 처리 프로세스와 AI 처리 프로세스를 분리- <code>signal(SIGALRM)</code>을 이용한 명령 타이머 (예: 10초 안에 응답 없으면 종료)- <code>pipe</code>, <code>message queue</code> 등의 IPC를 이용해 서버 내 프로세스 간 통신- <code>pthreads</code> 기반의 멀티스레드로 클라이언트 다중 처리 지원
소켓 프로그래밍	<ul style="list-style-type: none">- TCP 기반 클라이언트-서버 구조- 명령 요청 및 응답을 실시간으로 주고받음- 서버는 복수의 클라이언트와 동시 통신 가능
키보드 제어	<ul style="list-style-type: none">- 클라이언트에서 키보드 입력 감지 및 전송- 특수 명령어(예: <code>/exit</code>, <code>/ai [질문]</code>)를 통해 LLM API 연동 기능 활성화
셸 스크립트	<ul style="list-style-type: none">- <code>Makefile</code>을 통한 컴파일 자동화- <code>start.sh</code>, <code>stop.sh</code> 등을 통한 서버 및 클라이언트 자동 실행 및 로그 관리
LLM API 연동	<ul style="list-style-type: none">- <code>/ai [질문]</code> 형식의 명령어가 입력되면 서버는 OpenAI API 또는 로컬 LLM 서버에 HTTP 요청을 보내고, 응답을 클라이언트로 전달- 비동기 처리로 응답 지연 시에도 시스템이 동작 가능하도록 설계

3. 시스템 구성도



4. 기대 효과 및 학습 요소

- 시스템 프로그래밍 전반 (프로세스, IPC, 시그널, 스레드)에 대한 실제 구현 역량 강화
- 소켓 통신의 실질적인 구조 이해 및 동시 연결 처리 능력 배양
- 실제 LLM API (예: OpenAI) 연동을 통한 현대적 AI 시스템 구성 경험
- 셸 스크립트 및 Makefile 작성 실습으로 자동화 기술 습득