

시스템 프로그래밍 프로그램 보고서

◆ 프로젝트 제목

TCP 기반 파일 저장 서버/클라이언트 시스템 구현

◆ 프로젝트 목표

- TCP 기반의 안정적인 네트워크 통신을 바탕으로, 사용자가 입력한 파일 이름과 내용을 서버로 전송하고, 서버에서 해당 내용을 실제 파일로 저장하도록 하는 기능 구현.
- 최대 3개의 클라이언트 동시 접속 처리.
- 일정 시간(30초) 동안 무응답한 클라이언트를 타임아웃으로 종료.
- 서버 종료 시 모든 클라이언트와의 연결도 정리되도록 설계.

◆ 프로젝트 개요

본 프로젝트는 TCP/IP 소켓 프로그래밍을 기반으로 한 클라이언트-서버 모델을 구현한 시스템입니다.

사용자는 클라이언트 프로그램을 통해 파일 이름과 내용을 입력하면, 서버는 해당 정보를 받아 실제로 파일로 저장합니다.

이 과정에서 서버는 멀티스레딩을 활용하여 여러 클라이언트를 동시에 처리하며, poll() 시스템 호출을 통해 클라이언트의 응답 시간을 체크하여 30초 이상 응답이 없을 경우 해당 연결을 종료합니다.

또한 서버는 SIGINT 시그널을 감지하여 종료되며, 이때 모든 클라이언트의 소켓을 정리한 뒤 안전하게 종료됩니다.

◆ 시스템 구조 및 동작 흐름도

```
+-----+   TCP 연결   +-----+
| Client A | <-----> | Server  |
| Client B | <-----> | (멀티스레드)
| Client C | <-----> +-----+
```

Client 동작:

1. 사용자로부터 파일명/내용 입력 받음
2. 서버로 전송 (REQ_PACKET)
3. 서버 응답 수신 (RES_PACKET)

Server 동작:

1. 클라이언트 요청 수신 (poll()으로 30초 타이머)
2. 자식 프로세스를 fork하여 파일 생성/쓰기

3. 결과 응답 전송
4. END 수신 or 무응답 시 연결 종료

◆ 결과 예시

클라이언트 측 출력:

파일 이름 입력 (또는 /exit): *test.txt*
 저장할 파일 내용 입력: *Hello, TCP!*
[Tx] cmd: 1, filename: test.txt, filedata: Hello, TCP!
[Rx] cmd: 2, result: 1. 파일 저장 완료 (exit code: 0)

서버 측 출력:

Connected client IP: 127.0.0.1
Received filename: test.txt
Received content: Hello, TCP!
Child: wrote to file test.txt

타임아웃 발생 시:

Client 4 timed out (no data for 30 seconds).
Client 4 disconnected.

클라이언트 비정상 종료 감지:

[서버 연결 종료 감지됨]

◆ 주요 구현 사항 요약

항목	구현 내용
통신 방식	TCP 소켓 (IPv4)
파일 저장 방식	서버에서 자식 프로세스가 직접 파일 작성
멀티 클라이언트 처리	pthread (최대 3개)
타임아웃 처리	poll()을 통한 30초 감지 후 종료
서버 종료 처리	SIGINT 시 모든 클라이언트 연결 종료
클라이언트 종료 감지 및 처리	read() 반환값 검사로 종료 감지 후 pthread_cancel()

◆ 참고 자료

1. 1. 구글 서치
2. 2. Linux Man Pages: poll(2), fork(2), signal(2), pthread(3)
3. 3. 수업 강의자료: 시스템 프로그래밍 실습 PPT
4. 4. <https://man7.org/linux/man-pages/>