0. 목차

1. 네트워크 기초 이론

---

2. C 언어와 리눅스 네트워크 환경 준비

3. TCP 소켓 프로그래밍

- 3.1 TCP 서버 구현 (단일 접속)
- 3.2 TCP 클라이언트 구현
- 3.3 다중 클라이언트 처리: `fork` 기반
- 3.4 다중 클라이언트 처리: `pthread` 기반
- 3.5 접속 끊김, 오류 처리, 종료 처리
- 3.6 TCP 연결 유지 (`keep-alive`, `SO_LINGER` 옵션)

4. UDP 소켓 프로그래밍

- 4.1 UDP 서버/클라이언트 구현
- 4.2 `recvfrom`, `sendto` 함수 이해
- 4.3 클라이언트 주소 식별 및 처리
- 4.4 UDP 패킷 손실/순서 문제 대응
- 4.5 DNS 클라이언트 샘플 구현

5. 소켓 관련 주요 옵션 및 설정

- 5.1 `setsockopt`, `getsockopt`
- 5.2 `SO_REUSEADDR`, `SO_REUSEPORT`
- 5.3 소켓 타임아웃 설정 (`SO_RCVTIMEO`, `SO_SNDTIMEO`)
- 5.4 TCP_NODELAY, 버퍼 크기 조정
- 5.5 `fcntl`을 이용한 소켓 비동기 설정

6. 입출력 멀티플렉싱 (I/O Multiplexing)

- 6.1 `select` 함수 사용법
- 6.2 `poll` 함수 사용법
- 6.3 `epoll` 구조와 사용법
- 6.4 `epoll`의 LT(Level-Triggered) vs ET(Edge-Triggered)
- 6.5 고성능 서버를 위한 `epoll` 기반 구조 설계

7. 멀티프로세스와 멀티스레드 서버

- 7.1 `fork()` 기반 다중 접속 서버
- 7.2 `pthread` 기반 멀티스레드 서버
- 7.3 `thread pool` 개념 및 구현

- 7.4 race condition, mutex, semaphore 사용법
- 7.5 프로세스/스레드 간 통신 (pipe, socketpair, message queue 등)

---

8. 고급 I/O 및 논블로킹 처리

- 8.1 `non-blocking I/O` 설정 (`O_NONBLOCK`)
- 8.2 `read`, `write`의 반환값 처리
- 8.3 `recv`, `send`의 `MSG_DONTWAIT`, `MSG_PEEK` 옵션
- 8.4 `ioctl`을 이용한 소켓 제어
- 8.5 파일 디스크립터 기반 이벤트 루프 구성

---

9. 패킷 처리와 프로토콜 구현

- 9.1 C로 직접 구현하는 HTTP 서버 (GET/POST 지원)
- 9.2 FTP/SMTP와 같은 간단한 응용 계층 프로토콜 구현
- 9.3 RAW 소켓을 이용한 커스텀 패킷 생성
- 9.4 `libpcap`을 이용한 패킷 스니핑
- 9.5 ICMP 패킷 처리 (ping 프로그램 구현)

---

10. IPv6 네트워크 프로그래밍

- 10.1 IPv6 주소 구조 이해
- 10.2 `sockaddr_in6`, `inet_pton`, `inet_ntop` 사용법
- 10.3 IPv6 서버/클라이언트 구현
- 10.4 듀얼 스택 서버 구현

---

11. 보안 소켓 프로그래밍

- 11.1 TLS/SSL 기초 이론
- 11.2 OpenSSL을 이용한 C 언어 기반 SSL 서버/클라이언트 구현
- 11.3 인증서 발급, 키 교환, 암호화 처리
- 11.4 인증된 HTTPS 서버 구축

---

12. 네트워크 디버깅 및 성능 튜닝

- 12.1 `tcpdump`, `wireshark` 실습
- 12.2 RTT, 패킷 드롭, 리트랜스미션 분석
- 12.3 `strace`, `lsof`를 통한 시스템 콜 추적
- 12.4 커널 파라미터 (`/proc/sys/net/`)

- 12.5 `iperf`, `netperf`를 이용한 대역폭 측정

---

13. 실제 프로젝트 예제

- 13.1 채팅 서버/클라이언트 구현
- 13.2 멀티 클라이언트 Echo 서버
- 13.3 파일 업로드/다운로드 서버
- 13.4 RESTful API 프록시 서버
- 13.5 부하 테스트용 네트워크 트래픽 생성기

---

14. 고급 주제 및 최신 기술 연동

- 14.1 비동기 네트워크 라이브러리 (libevent, libuv) 사용법
- 14.2 C와 MQTT 연동 (mosquitto)
- 14.3 멀티코어 네트워크 처리 모델
- 14.4 커널 우회 네트워킹 (DPDK, XDP 개요)
- 14.5 IoT 디바이스 네트워크 연동 (C 기반 TCP/IP Stack 내장)