

자기주도 프로젝트

멀티스레드 채팅 서버 구현



계절학기

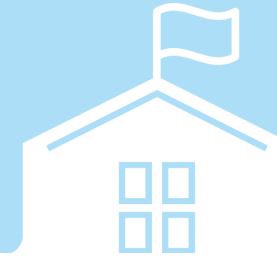
자기주도 PRJ : 멀티스레드 채팅 서버 구현

한 기 철 Project consultant

- KTNET 글로벌 비즈실
- 티맥스소프트 연구소 BP실
- 쌍용정보통신 KT IPTV 프로젝트
- Sun Microsystems Java Professional course
- 서강대학교 컴퓨터공학 석사



프로그램




- ✓ 개발자가 작성한 명령어들의 집합군
- ✓ 크롬, 메모장 등은 하나의 프로그램
- ✓ SSD/HDD 같은 보조기억장치에 저장되어짐
- ✓ 내가(초보자) 직접 작성한 C언어 소스코드 컴파일 결과물인 실행파일 하나도 프로그램



- ★ Chrome, Notepad ...
- ★ HelloWorld

프로세스

- ✓ '운영체제가 관리하는 하나의 실행 단위'
- ✓ 보통 GUI환경에서 아이콘 마우스 더블클릭시 해당 '프로그램'을 운영체제의 메커니즘에 기반한 단위로 메모리에 상주시킴
- ✓ 각 운영체제별 프로세스 조회 및 관리 방법 제공됨

 작업 관리자

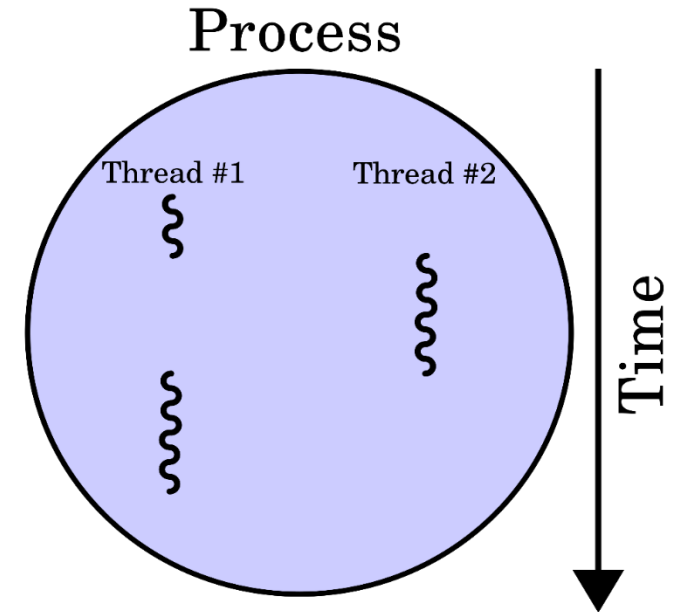
파일(F) 옵션(O) 보기(V)

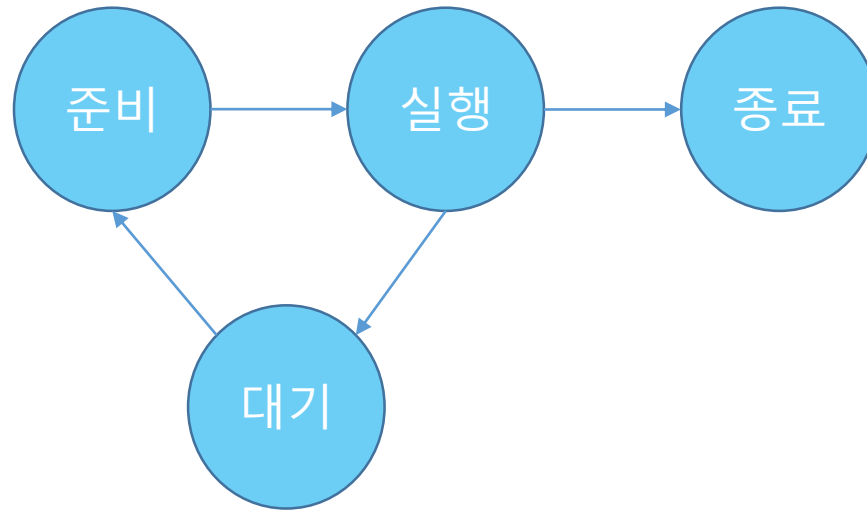
프로세스 성능 앱 기록 시작프로그램 사용자 세부 정보 서비스

이름	상태	11%	85%
		CPU	메모리
> Google Chrome(82)		3.4%	2,483.5MB
> WebStorm(5)		0%	693.7MB
> PyCharm(6)		0%	511.8MB

스레드

- ✓ '프로세스 내에서 실행되는 작업 흐름의 단위'
- ✓ CPU 는 한번에 하나의 명령 수행하게 되는데 스레드를 활용하여 동일 프로세스 내 여러 독립된 작업을 동시에 수행하는 듯한 효과를 볼 수 있음





★ Priority(우선 순위) 근거 상태 전이

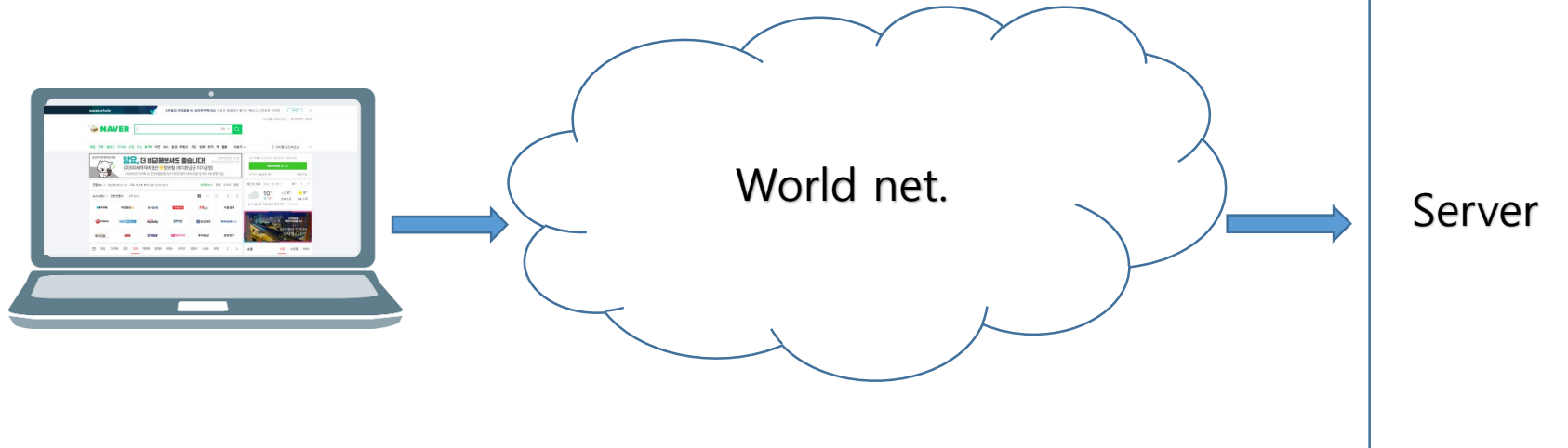
스레드 예시 - 동영상 플레이어



- ✓ 동일 프로세스 내 영상과 사운드 데이터를 동시에 풀어내어 각 장치에 전송해줘야 하는 미션
- ✓ 스레드 프로그래밍을 통해 동시(성) 작업이 가능



네트워크 기초 - OSI 7계층



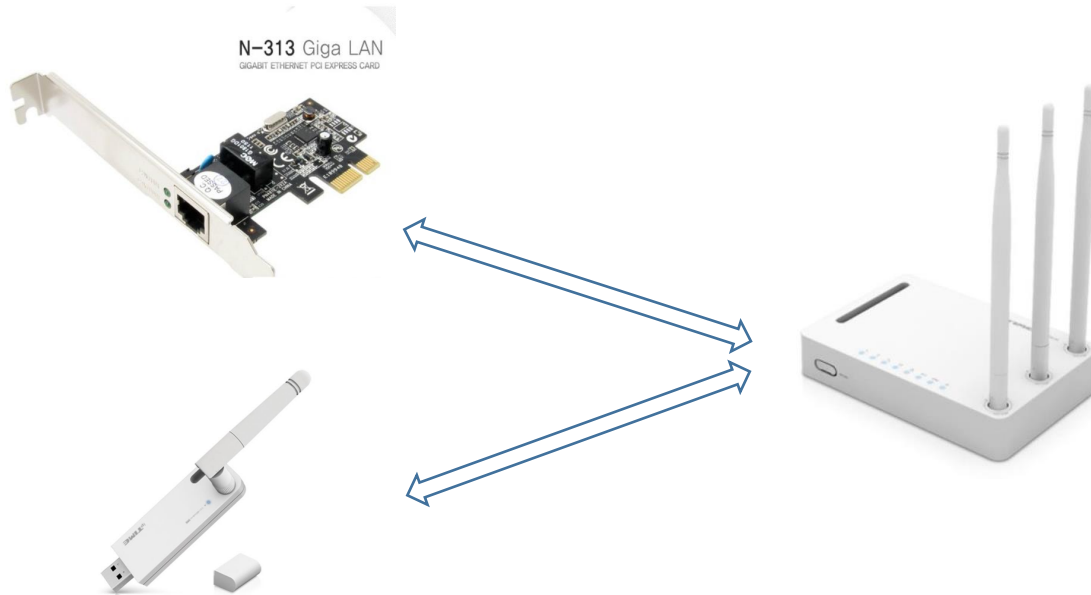
- ✓ 어플리케이션부터 통신 단말 장치까지 모든 과정을 계층적으로 구분하여 정의

OSI 7계층 - 물리 계층 (L1)



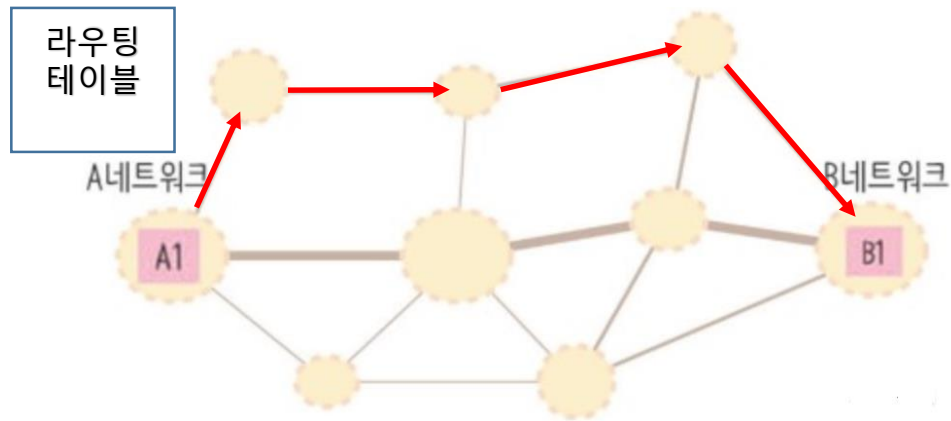
물리적 bit 신호 송수신
MAC 주소 부여 38-00-25-07-72-4A

OSI 7계층 - 데이터 링크 계층 (L2)



물리적 신호 오류 감지, 수정 및 전송
Ethernet(이더넷) 이 대표적

OSI 7계층 - 네트워크 계층 (L3)

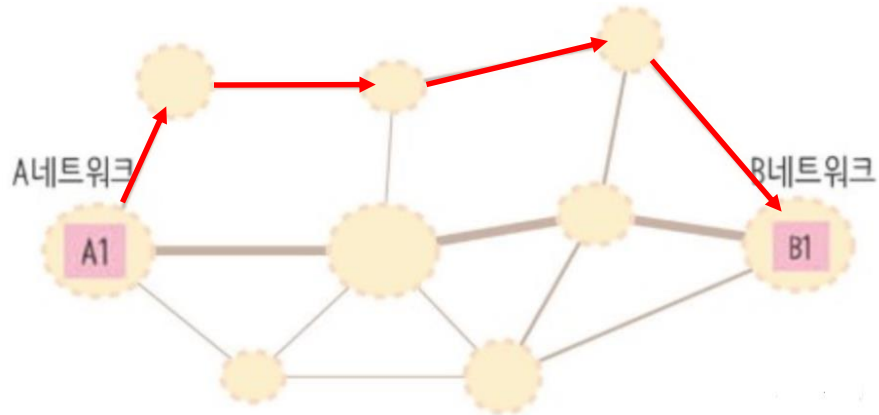


패킷 전송 경로 설정

Router의 라우팅 테이블을 근거로 경로 결정

Internet Protocol이 대표적

OSI 7계층 - 전송 계층 (L4)



전송 제어 (오류검출, 재전송등)

TCP, UDP가 대표적

OSI 7계층 - 세션 계층 (L5)



응용 프로세스에서의 통신 관리(유지)

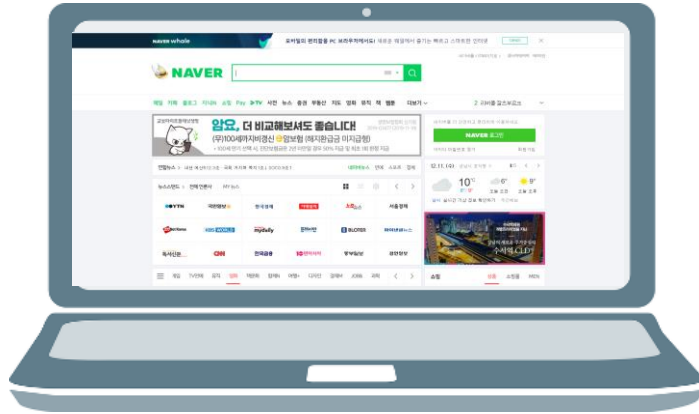
포트 바인딩

OSI 7계층 - 표현 계층 (L6)



데이터 압축, 암호화

OSI 7계층 - 응용 계층 (L7)



End application

HTTP, FTP 등

소켓 통신



- ✓ 소프트웨어로 작성된 **추상적인 개념**의 통신 접속점
- ✓ 운영체제 레벨에서 제공어진다고 볼 수 있음

소켓 통신 구현



✓ 내가 선택한 언어에서 **소켓 통신**을 어떻게 구현할 수 있는가?

소켓 통신 구현



java.net

Class ServerSocket

```
public class ServerSocket  
extends Object  
implements Closeable
```

This class implements server sockets. A server socket waits for requests to come in over the network. It performs some operation based on that request, and then possibly returns a result to the requester.

파이썬 소켓 프로그래밍 - 클라이언트 / 서버 예제

Python/Python 소켓 프로그래밍 / webnautes / 2019. 9. 27. 16:09

```
import socket

# 접속할 서버 주소입니다. 여기에서는 루프백(loopback) 인터페이스 주소 즉 localhost를 사용합니다.
HOST = '127.0.0.1'
# 클라이언트 접속을 대기하는 포트 번호입니다.
PORT = 9999

# 소켓 객체를 생성합니다.
# 주소 체계(address family)로 IPv4, 소켓 타입으로 TCP 사용합니다.
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

✓ 공식 API 문서를 참고하거나 도서 혹은 블로그를 검색해보면 어렵지 않게 소켓 예제를 찾을 수 있다

스레드 프로그래밍



- ✓ 클라이언트 프로세스는 서버로부터 소켓 통신을 통해 메시지를 읽어들이고 콘솔에 출력하는 작업과
- ✓ 콘솔창의 키보드 입력 텍스트를 소켓 통신을 통해 서버로 전송하는 작업
- ✓ 채팅 서버 역시 최대 5명까지의 소켓 세션이 유지되고 있는 클라이언트들로부터의 메시지 수신과 송신을 동시에 수행해야 함
- ✓ 메시지 수신/송신 두 가지 기능을 동시에 처리할 수 있어야 하며 이 때 필요한것이 각각의 작업을 동시에 별도 처리할 수 있는 스레드 프로그래밍

스레드 프로그래밍



```
// 메시지 전송 thread
Thread sendMessage = new Thread(new Runnable() {
    @Override
    public void run() {
        while (true) {
            String msg = scn.nextLine();
            try {
                dos.writeUTF(msg);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
});
```

3.8.14 3.8.14 Documentation » The Python Standard Library » Concurrent Execution »

threading — Thread-based parallelism ¶

Source code: [Lib/threading.py](#)

This module constructs higher-level threading interfaces on top of the lower level `_thread` module. See also the `queue` module.

- ✓ 소켓과 마찬가지로 선택 언어에서 어떻게 스레드를 활용 할 수 있는지와 같은 기본 사용법을 찾아보는 건 개발자의 기본 덕목임