

Chapter 07.네트워크 프로그래밍 기초

Mingyu Lim

Collaborative Computing Systems Lab.

(<http://ccslab.konkuk.ac.kr>)

School of Internet & Multimedia Engineering

Konkuk University, Seoul, Korea

1. 소켓이란

□ 소켓

socket [미] [sa'kit] [영] [sɒk-]

명사

꽂는 [끼우는] 구멍, 장뚝구멍, 축 (軸) 받이
(전구 등을 꽂는) 소켓, 벽소켓; (춧대의) 초꽂이

【해부】 와 (窩) , 강 (腔)

the socket of the eye 눈구멍

【골프】 클럽의 힐

□사용자에게 네트워크에 접근할 수 있는 인터페이스를 제공

□소켓 생성, 소켓을 통한 송수신, 소켓 소멸

□TCP vs UDP

□ 소켓형식

– SOCK_STREAM

– SOCK_DGRAM

– SOCK_RAW

1. 소켓이란

□SOCK_STREAM (TCP)

- 바이트를 주고받을 수 있는 스트림 통신을 구현할 수 있게 해주는 소켓으로 양방향 통신 가능

□SOCK_DGRAM (UDP)

- 데이터그램 통신용 소켓으로 양방향 통신 가능

□SOCK_RAW

- 보안상 지원하지 않음
- 좀더 높은 수준의 제어를 하고 싶은 사용자를 위한 것
- 패킷 전달시 패킷 전달 경로도 지정 가능

ACHROMATIC COLOR

1. 소켓이란

□ 인터넷 주소와 포트

– 인터넷 주소

- ◆ 인터넷에서 유일하게 컴퓨터를 구분
- ◆ IP 주소 사용

– 포트

- ◆ 한 컴퓨터 내 여러 소켓을 구분하기 위해 필요
- ◆ 0 ~ 1023: well-known port
- ◆ 일반 사용자는 1023 이후의 포트 사용

CHROMATIC COLOR
TEXTURE

ACHROMATIC COLOR
TEXTURE

2. InetAddress 클래스 활용

□ InetAddress

- 도메인 주소와 IP 주소간 변환
- IP주소에 대한 정보
- 현재 컴퓨터의 이름

□ InetAddress 객체 생성 관련 메소드

- Static InetAddress[] getAllByName(String host)
 - ◆ 호스트의 모든 IP주소에 대한 정보를 InetAddress 배열형태로 반환
- Static InetAddress getByAddress(byte[] addr)
 - ◆ 바이트로 표현된 addr에 해당하는 IP정보를 InetAddress 객체 형태로 반환
- Static InetAddress getLocalHost()
 - ◆ 로컬호스트의 IP주소에 대한 정보를 InetAddress객체 형태로 반환

2. InetAddress 클래스 활용

□ InetAddress 중요 메소드

- Byte[] getAddress()
 - ◆ IP 주소를 바이트 형태로 반환
- String getHostAddress()
 - ◆ 호스트의 IP주소를 점으로 구분되는 10진수 형태로 반환
- String getHostName()
 - ◆ 호스트의 도메인명을 문자열로 반환
- Boolean isMulticastAddress()
 - ◆ 주소가 멀티캐스트 주소인지를 확인. 멀티캐스트 주소일 경우 true 반환 (224.0.0.0 ~ 239.255.255.255)
- String toString()
 - ◆ IP 주소를 문자열 형태로 반환 (host name / IP address)

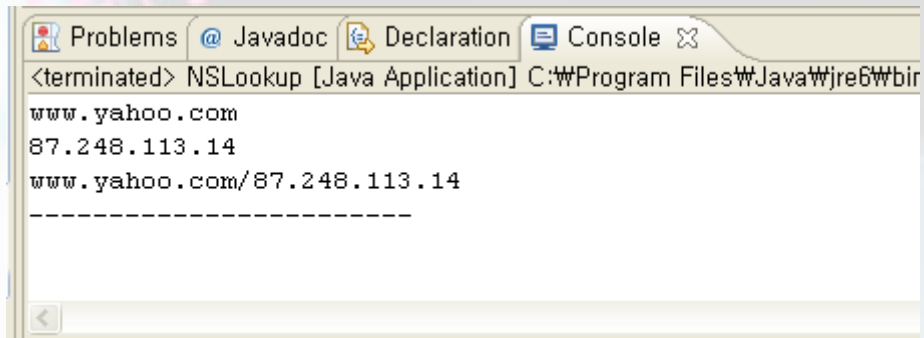
2. InetAddress 클래스 활용

□ InetAddress 클래스를 이용한 nslookup 명령 구현

```
public class NSLookup {  
  
    public static void main(String[] args) {  
        if(args.length != 1){  
            System.out.println("IP주소나 도메인 주소를 인자로 지정하세요.");  
            System.exit(0);  
        }  
        InetAddress inetaddr[] = null;  
        try {  
            inetaddr = InetAddress.getAllByName(args[0]);  
        } catch (UnknownHostException e) {  
            e.printStackTrace();  
        }  
        for(int i = 0; i < inetaddr.length; i++){  
            System.out.println(inetaddr[i].getHostName());  
            System.out.println(inetaddr[i]..getHostAddress());  
            System.out.println(inetaddr[i].toString());  
            System.out.println("-----");  
        }  
    }  
} // manin
```

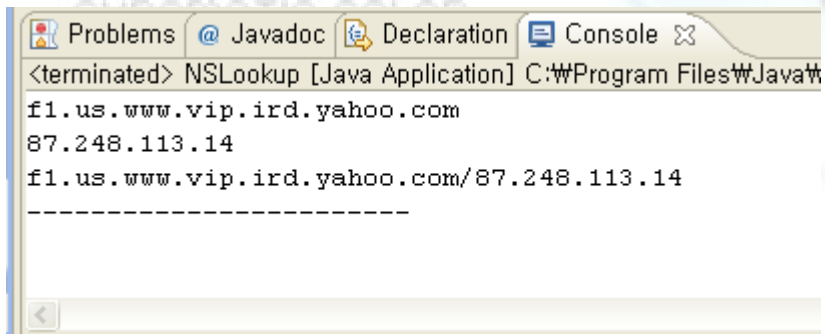
2. InetAddress 클래스 활용

□Java NSLookup www.yahoo.com



```
<terminated> NSLookup [Java Application] C:\Program Files\Java\jre6\bin
www.yahoo.com
87.248.113.14
www.yahoo.com/87.248.113.14
-----
```

□Java NSLookup 87.248.113.14



```
<terminated> NSLookup [Java Application] C:\Program Files\Java\
f1.us.www.vip.ird.yahoo.com
87.248.113.14
f1.us.www.vip.ird.yahoo.com/87.248.113.14
-----
```


2. InetAddress 클래스 활용

□ InetAddress 이용한 로컬컴퓨터명과 IP구하기

```
public class NSLookupLocal {  
  
    public static void main(String[] args) {  
        InetAddress inetaddr = null;  
        try {  
            inetaddr = InetAddress.getLocalHost();  
        } catch (UnknownHostException e) {  
            e.printStackTrace();  
        }  
        System.out.println(inetaddr.getHostName());  
        System.out.println(inetaddr.getHostAddress());  
        System.out.println("byte[] 형식의 ip 주소 값의 출력.");  
        byte[] ip = inetaddr.getAddress();  
        for(int i = 0; i < ip.length; i++){  
            System.out.print((int)ip[i]);  
            if(i != ip.length - 1)  
                System.out.print(" ");  
        }  
    }  
}
```

