



윤성우의 열혈 TCP/IP 소켓 프로그래밍

윤성우 저 열혈강의 TCP/IP 소켓 프로그래밍 개정판

Chapter 08. 도메인 이름과 인터넷 주소



Chapter 08-1. Domain Name System

윤성우 저 열혈강의 TCP/IP 소켓 프로그래밍 개정판

도메인 이름과 DNS 서버

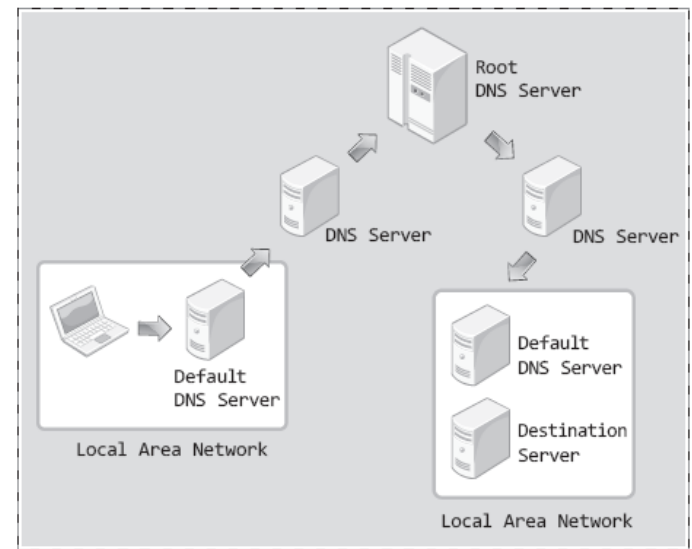
▶ 도메인 이름

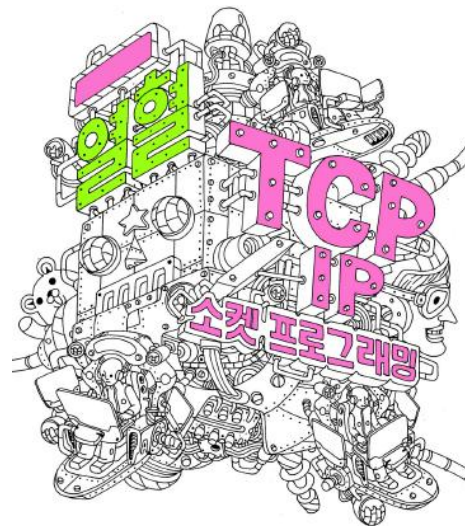
- ▶ IP를 대신하는 서버의 주소 ex) www.naver.com
- ▶ 실제 접속에 사용되는 주소는 아니다. 이 정보는 IP로 변환이 되어야 접속이 가능하다.

▶ DNS 서버

- ▶ 도메인 이름을 IP로 변환해주는 서버
- ▶ DNS는 일종의 분산 데이터베이스 시스템이다.

도메인 이름을 이용해서 서버에 접속하면, 접속 이전에 DNS 서버에 해당 도메인의 IP 주소를 묻게 되고, 그 결과로 얻게 된 IP를 이용해서 서버에 접속하게 된다.





Chapter 08-2. IP주소와 도메인 이름 사이의 변환

윤성우 저 열혈강의 TCP/IP 소켓 프로그래밍 개정판

도메인 이름을 이용해서 IP주소 얻어오기

```
#include <netdb.h>
```

```
struct hostent * gethostbyname(const char * hostname);
```

➔ 성공 시 hostent 구조체 변수의 주소 값, 실패 시 NULL 포인터 반환

gethostbyname 함수의 인자로 도메인의 이름 정보를 전달하면, 해당 도메인의 서버정보가 hostent 구조체 변수에 채워지고, 그 변수의 주소 값이 반환된다.

```
struct hostent
{
    char * h_name;           // official name
    char ** h_aliases;      // alias list
    int h_addrtype;         // host address type
    int h_length;           // address length
    char ** h_addr_list;    // address list
}
```

IP는 도메인 이름에 비해 상대적으로 변동이 심하다. 때문에 프로그램 코드상에서 서버의 IP를 직접 코드로 입력한다면, 서버의 IP가 변경될 때마다 컴파일을 다시해야 하는 번거로운 상황이 발생한다. 그러나 상대적으로 변동이 덜한 도메인 이름을 이용해서 서버가 실행될 때마다 IP를 얻어오게 구현한다면, 서버의 코드를 재 컴파일 할 필요가 없다.

구조체 hostent에 채워지는 정보의 형태

h_name :

공식 도메인 이름

h_aliases :

별칭의 도메인 이름

h_addrtype :

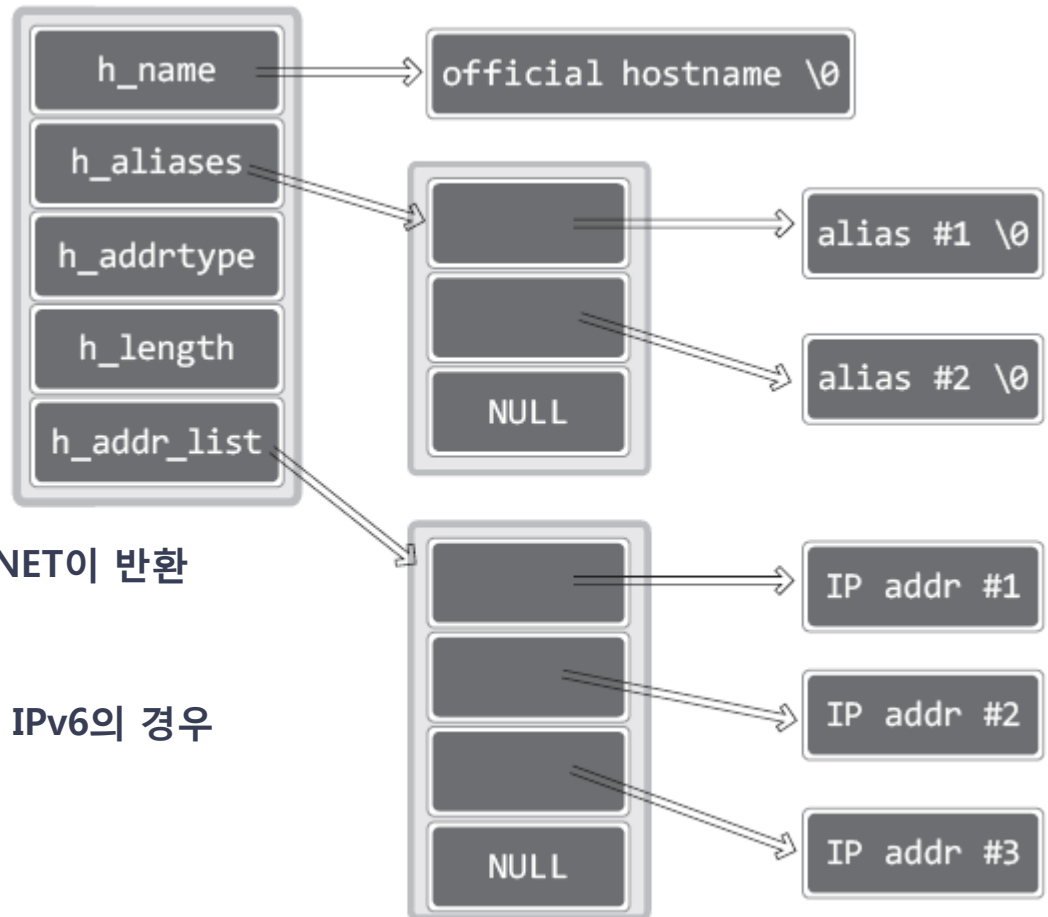
반환된 IP의 정보가 IPv4인 경우, AF_INET이 반환

h_length :

반환된 IP 정보의 크기, IPv4의 경우 4, IPv6의 경우 16이 저장

h_addr_list

IP의 주소정보, 둘 이상의 경우 모두 반환

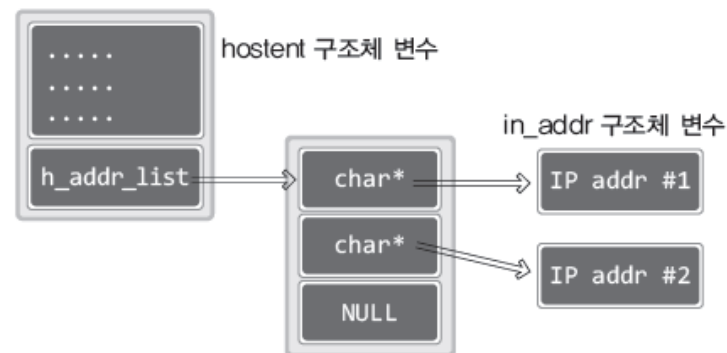


gethostbyname 함수의 호출 예

예제 gethostbyname.c의 일부

```
host=gethostbyname(argv[1]);
if(!host)
    error_handling("gethost... error");

printf("Official name: %s \n", host->h_name);
for(i=0; host->h_aliases[i]; i++)
    printf("Aliases %d: %s \n", i+1, host->h_aliases[i]);
printf("Address type: %s \n",
    (host->h_addrtype==AF_INET)?"AF_INET":"AF_INET6");
for(i=0; host->h_addr_list[i]; i++)
    printf("IP addr %d: %s \n", i+1,
        inet_ntoa(*(struct in_addr*)host->h_addr_list[i]));
return 0;
```



실행 결과

반복문의 구성을 통해서 반환된 모든 정보를 출력하고 있다. 정보의 끝은 NULL로 표시가 된다는 사실을 기억하자!

```
root@my_linux:/tcpip# gcc gethostbyname.c -o hostname
root@my_linux:/tcpip# ./hostname www.naver.com
Official name: www.g.naver.com
Aliases 1: www.naver.com
Address type: AF_INET
IP addr 1: 202.131.29.70
IP addr 2: 222.122.195.6
```

IP주소를 이용해서 도메인 정보 얻어오기

```
#include <netdb.h>
```

```
struct hostent * gethostbyaddr(const char * addr, socklen_t len, int family);
```

➔ 성공 시 hostent 구조체 변수의 주소 값, 실패 시 NULL 포인터 반환

- addr IP주소를 지니는 in_addr 구조체 변수의 포인터 전달, IPv4 이외의 다양한 정보를 전달받을 수 있도록 일반화하기 위해서 매개변수를 char형 포인터로 선언.
- len 첫 번째 인자로 전달된 주소정보의 길이, IPv4의 경우 4, IPv6의 경우 16 전달.
- family 주소체계 정보 전달. IPv4의 경우 AF_INET, IPv6의 경우 AF_INET6 전달.

gethostbyname 함수의 반대 기능 제공



gethostbyaddr 함수의 호출 예

예제 gethostbyaddr.c의 일부

```
memset(&addr, 0, sizeof(addr));
addr.sin_addr.s_addr=inet_addr(argv[1]);
host=gethostbyaddr((char*)&addr.sin_addr, 4, AF_INET);
if(!host)
    error_handling("gethost... error");

printf("Official name: %s \n", host->h_name);
for(i=0; host->h_aliases[i]; i++)
    printf("Aliases %d: %s \n", i+1, host->h_aliases[i]);
printf("Address type: %s \n",
    (host->h_addrtype==AF_INET)?"AF_INET":"AF_INET6");
for(i=0; host->h_addr_list[i]; i++)
    printf("IP addr %d: %s \n", i+1,
        inet_ntoa(*(struct in_addr*)host->h_addr_list[i]));
return 0;
```

실행 결과

```
root@my_linux:/tcpip# gcc gethostbyaddr.c -o hostaddr
root@my_linux:/tcpip# ./hostaddr 74.125.19.106
Official name: nuq04s01-in-f106.google.com
Address type: AF_INET
IP addr 1: 74.125.19.106
```



Chapter 08-3. 윈도우 기반으로 구현하기

윤성우 저 열혈강의 TCP/IP 소켓 프로그래밍 개정판

윈도우 기반 도메인 관련 함수

```
#include <winsock2.h>
```

```
struct hostent * gethostbyname(const char * name);
```

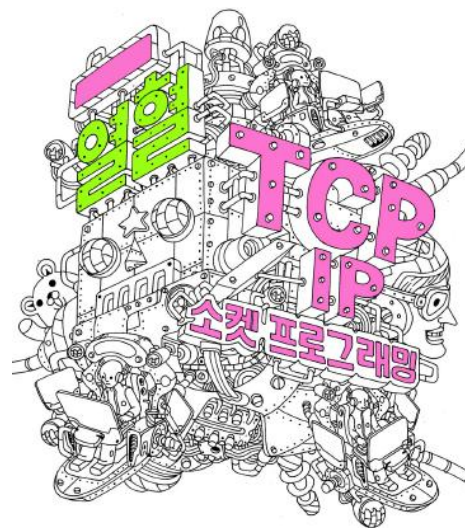
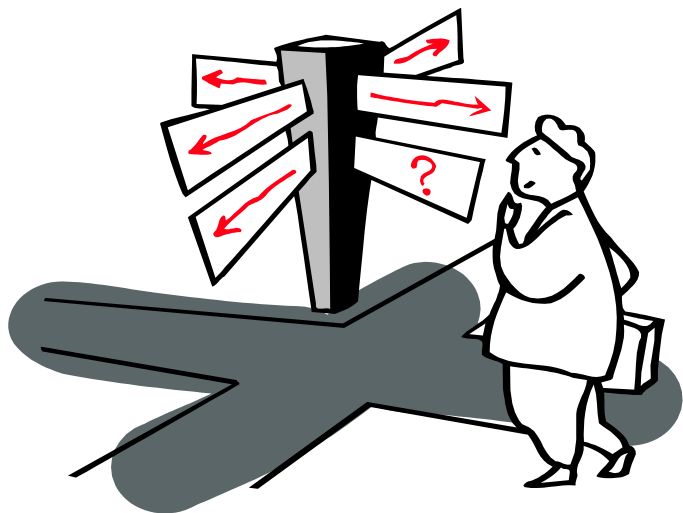
➔ 성공 시 hostent 구조체 변수의 주소 값, 실패 시 NULL 포인터 반환

```
#include <winsock2.h>
```

```
struct hostent * gethostbyaddr(const char * addr, int len, int type);
```

➔ 성공 시 hostent 구조체 변수의 주소 값, 실패 시 NULL 포인터 반환

전달인자의 이름에서만 차이가 있을 뿐 그 이외의 모든 것은 Linux의 함수와 동일하다.



Chapter 08이 끝났습니다. 질문 있으신지요?