

네트워크 프로그래밍

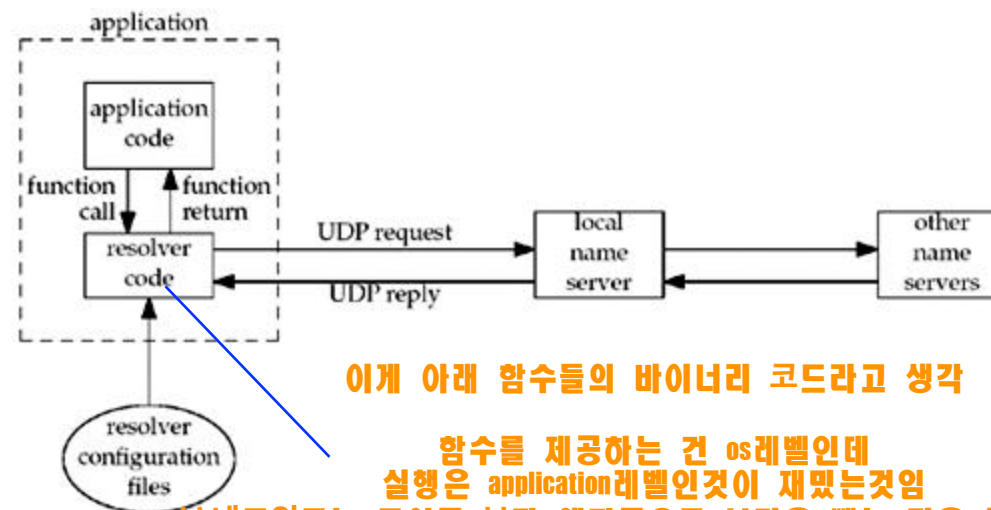
11. Domain Name Service

도메인 이름

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

네임 변환 서비스

- 도메인 네임 서비스(DNS, Domain Name Service)
 - ▣ 도메인 이름을 IP로 변환해주는 서버
 - ▣ DNS는 일종의 분산 데이터베이스 시스템이다.
- 지역 설정 정보(local configuration database)
 - ▣ 유닉스 시스템: /etc/hosts
 - ▣ 윈도우 시스템: c:\windows\system32\drivers\etc\hosts



이게 아래 함수들의 바이너리 코드라고 생각

함수를 제공하는 건 os레벨인데
실행은 application레벨인것이 재밌는것임

>>네트워크는 코어쪽 보다 엣지쪽으로 부담을 빼는 것을 좋아하므로
이것도 마찬가지로 좀더 높은 레벨쪽으로 부담을 뺐다고 생각하면됨

도메인 이름을 이용해서 IP주소 얻어오기

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

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

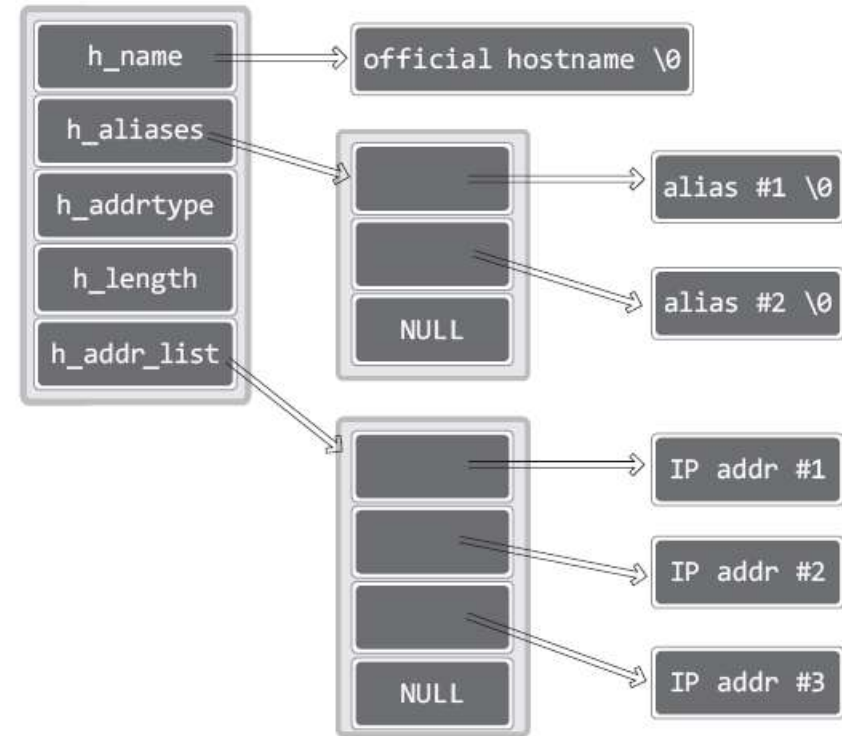
```
struct hostent * gethostbyname(const char * hostname);
```

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

```
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
}
```

구조체 hostent에 채워지는 정보의 형태

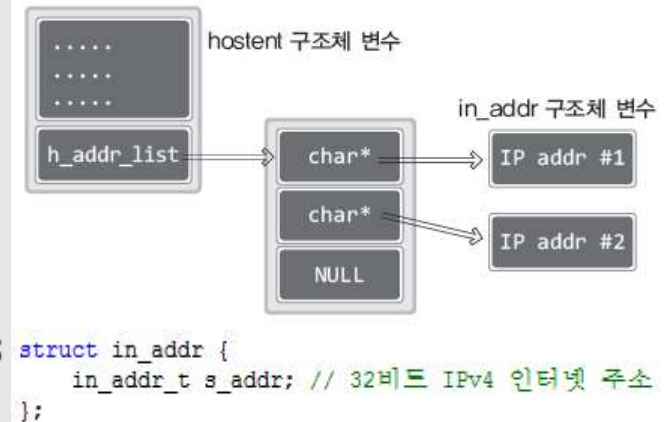
- **h_name**
 - ▣ 공식 도메인 이름
- **h_aliases**
 - ▣ 별칭의 도메인 이름
- **h_addrtype**
 - ▣ 반환된 IP의 정보가 IPv4인 경우, AF_INET이 반환
- **h_length**
 - ▣ 반환된 IP 정보의 크기, IPv4의 경우 4, IPv6의 경우 16이 저장
- **h_addr_list**
 - ▣ IP의 주소정보, 둘 이상의 경우 모두 반환



gethostbyname 함수의 호출 예

```
host=gethostbyname(argv[1]);   gethostbyname.c의 일부
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;   ip는 항상 이런 구조체 형식으로 저장한다.
```



```
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

여러 경우를 지원해주려고 포인터로 썼지만, 옛날이라 void가 아닌 char를 사용했다. 요즘에 만들었으면 void로 만들었을듯

```
#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.

gethostbyaddr 함수의 호출 예

```
memset(&addr, 0, sizeof(addr));          gethostbyaddr.c의 일부
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;
```

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