

12주차 2차시 소켓 프로그래밍 기초

【학습목표】

1. 소켓의 기본 개념을 이해한다.
2. 소켓 프로그래밍의 기본 개념을 설명할 수 있다.

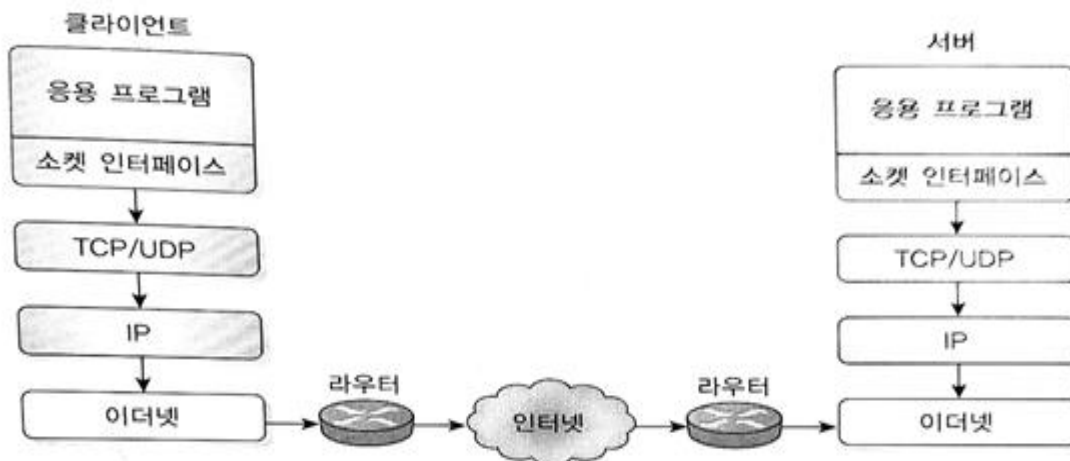
학습내용1 : 소켓

1. 소켓의 종류

* 개념

- 응용 계층과 전송 계층을 연결 기능 제공
- 네트워크에 대한 사용자 수준의 인터페이스를 제공
- 소켓은 양방향 통신 방법으로 클라이언트-서버 모델을 기반으로 프로세스 사이의 통신에 매우 적합하다

* 소켓 인터페이스



- 응용 프로그램과 TCP 계층을 연결하는 역할
- 네트워크 프로그래밍 작성 용이

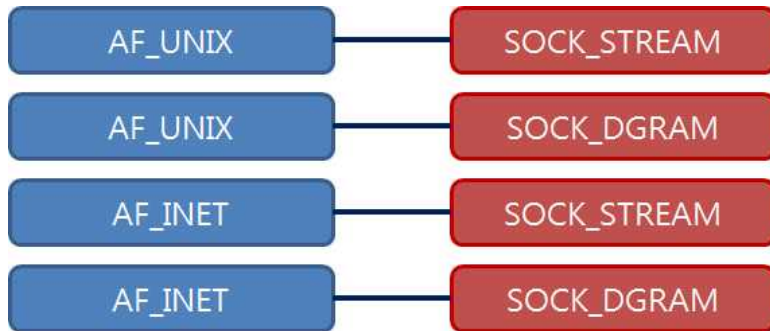
* 종류

- AF_UNIX : 유닉스 도메인 소켓 (시스템 내부 프로세스간 통신)
- AF_INET : 인터넷 소켓 (네트워크를 이용한 통신)

2. 소켓의 통신 방식

SOCK_STREAM : TCP 사용

SOCK_DGRAM : UDP 사용



학습내용2 : 소켓 프로그래밍

1. 소켓 주소 구조체

* 유닉스 도메인 소켓의 주소 구조체

```

struct sockaddr_un {
    sa_family_t
    sun_famly;
    char
    sun_path[108]
};
  
```

- sun_family : AF_UNIX
- sun_path : 통신에 사용할 파일의 경로명

* 인터넷 소켓의 주소 구조체

```

struct sockaddr_in {
    sa_family_t sin_family;
    in_port_t sin_port;
    struct in_addr sin_addr;
};

struct in_addr {
    in_addr_t s_addr;
};
  
```

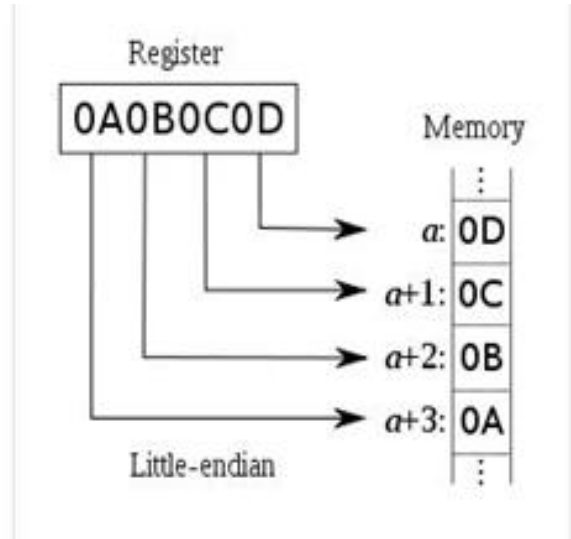
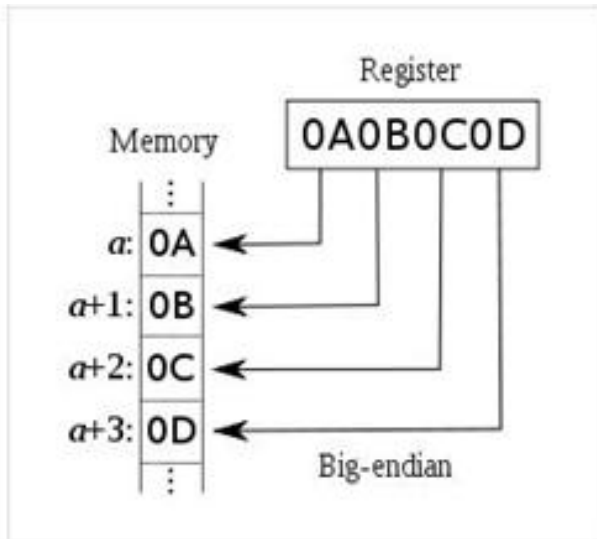
- sin_family : 주소 패밀리
- sin_port : 주소 포트 번호
- sin_addr : IP 주소

2. 바이트 순서 함수

* 정수를 저장하는 방식 : 빅엔디안, 리틀엔디안

- 빅엔디안 : 메모리의 낮은 주소에 정수의 첫 바이트를 위치 -> 모토로라, 썬
- 리틀엔디안 : 메모리의 높은 주소에 정수의 첫 바이트를 위치 -> 인텔

종류	1234	12345678
빅 엔디안	1234	12345678
리틀 엔디안	4321	87654321



- TCP/IP 네트워크에서 바이트 순서 표준 : 빅엔디안
- 호스트 바이트 순서(HBO¹⁾) : 시스템에서 사용하는 바이트 순서
- 네트워크 바이트 순서(NBO²⁾) : 네트워크에서 사용하는 바이트 순서

```
#include <sys/types.h>
#include <netinet/in.h>
#include <inttypes.h>
uint32_t htonl(uint32_t hostlong);
uint16_t htons(uint16_t hostshort);
uint32_t ntohl(uint32_t netlong);
uint16_t ntohs(uint16_t netshort);
```

- htonl : 32비트 HBO를 32비트 NBO로 변환
- htons : 16비트 HBO를 16비트 NBO로 변환
- ntohl : 32비트 NBO를 32비트 HBO로 변환
- ntohs : 16비트 NBO를 16비트 HBO로 변환

1) HBO : Host Byte Order

2) NBO : Network Byte Order

* NBO를 HBO로 변환하기

```

01 #include <netdb.h>
02 #include <stdio.h>
03
04 int main(void) {
05     struct servent *port;
06     int n;
07
08     setservent(0);
09
10     for (n = 0; n < 5; n++) {
11         port = getservent();
12         printf("Name=%s, Port=%d\n", port->s_name,
13             ntohs(port->s_port));
14     }
15     endservent();
16
17     return 0;
18 }

```

NBO를 HBO로 변환하기 위한 함수 호출

```

# ex11_3.out
Name=tcpmux, Port=1
Name=echo, Port=7
Name=echo, Port=7
Name=discard, Port=9
Name=discard, Port=9

```

* HBO를 NBO로 변환하기

```

01 #include <netdb.h>
02 #include <stdio.h>
03
04 int main(void) {
05     struct servent *port;
06
07     port = getservbyname("telnet", "tcp");
08     printf("Name=%s, Port=%d\n", port->s_name, ntohs(port->s_port));
09
10     port = getservbyport(htons(21), "tcp");
11     printf("Name=%s, Port=%d\n", port->s_name, ntohs(port->s_port));
12
13     return 0;
14 }

```

이름으로 서비스 포트번호 검색

HBO를 NBO로 변환하여 포트번호 검색

```

# ex11_4.out
Name=telnet, Port=23
Name=ftp, Port=21

```

3. IP주소 변환 함수

* IP주소의 형태

- 192.168.10.1과 같이 점(.)으로 구분된 형태
- 시스템 내부 저장 방법 : 이진값으로 바뀌어서 저장
- 외부적 사용 형태 : 문자열로 사용

* 문자열 형태의 IP주소를 숫자형태로 변환 : inet_addr(3)

```

#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
in_addr_t inet_addr(const char *cp);

```

* 구조체 형태의 IP주소를 문자열 형태로 변환: inet_ntoa(3)

```

#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
char *inet_ntoa(const struct in_addr in);

```

* IP 주소 변환하기

```

...
09 int main(void) {
10     in_addr_t addr;
11     struct hostent *hp;
12     struct in_addr in;
13
14     if ((addr = inet_addr("218.237.65.4")) == (in_addr_t)-1) {
15         printf("Error : inet_addr(218.237.65.4\n");
16         exit(1);
17     }
18
19     hp = gethostbyaddr((char *)&addr, 4, AF_INET);
20     if (hp == NULL) {
21         (void) printf("Host information not found\n");
22         exit(2);
23     }
24
25     printf("Name=%s\n", hp->h_name);
26
27     (void) memcpy(&in.s_addr, *hp->h_addr_list, sizeof (in.s_addr)); 0;
28     printf("IP=%s\n", inet_ntoa(in));
29
30     return 0;
31 }

```

문자열 행태를 이진형태로 변환

주소로 호스트명 검색

구조체 형태에서 문자열로 변환하여 출력

```

# gcc -o ex11_5.out ex11_5.c -lsocket -lnsl
# ex11_5.out
Name=www.hanb.co.kr
IP=218.237.65.4

```

【학습정리】

1. 소켓의 종류

* 개념

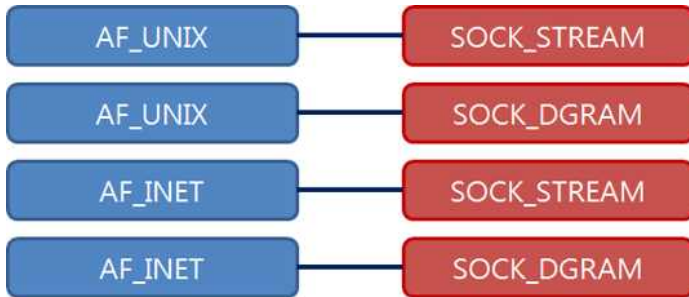
- 응용 계층과 전송 계층을 연결 기능 제공
- 네트워크에 대한 사용자 수준의 인터페이스를 제공
- 소켓은 양방향 통신 방법으로 클라이언트-서버 모델을 기반으로 프로세스 사이의 통신에 매우 적합하다.

2. 종류

- AF_UNIX : 유닉스 도메인 소켓 (시스템 내부 프로세스간 통신)
- AF_INET : 인터넷 소켓 (네트워크를 이용한 통신)

3. 소켓의 통신 방식

- SOCK_STREAM : TCP 사용
- SOCK_DGRAM : UDP 사용



4. 소켓 프로그래밍

- * 소켓 주소 구조체 : 유닉스 도메인 소켓의 주소 구조체

```
struct sockaddr_un {
    sa_family_t
    sun_famly;
    char
    sun_path[108]
};
```

- sun_family : AF_UNIX
- sun_path : 통신에 사용할 파일의 경로명

5. 소켓 프로그래밍

- 인터넷 소켓의 주소 구조체

```
struct sockaddr_in {
    sa_family_t sin_family;
    in_port_t sin_port;
    struct in_addr sin_addr;
};

struct in_addr {
    in_addr_t s_addr;
};
```

- sin_family : 주소 패밀리
- sin_port : 주소 포트 번호
- sin_addr : IP 주소