

TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정



2018.03.29

26 일차

강사 - Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 - 신민철

akrn33@naver.com

sigaction1.c

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#include<signal.h>
```

```
struct sigaction act_new;
```

```
struct sigaction act_old;
```

```
void sigint_handler(int signo)
```

```
{
```

```
    printf("Ctrl + C\n");
```

```
    printf("if you push it one more time then exit\n");
```

```
    sigaction(SIGINT, &act_old, NULL);
```

```
}
```

```
int main(void)
```

```
{
```

```
    act_new.sa_handler = sigint_handler;
```

```
    sigemptyset(&act_new.sa_mask); //act_new.sa_mask 를 비워줌
```

```
    sigaction(SIGINT, &act_new, &act_old); // 이전에 시그널 동작한  
정보가 액트올드에 들어옴
```

```
    while(1)
```

```
    {
```

```
        printf("sigactino test\n");
```

```
        sleep(1);
```

```
    }
```

```
    return 0;
```

```
}
```

kill.c

```
#include<stdio.h>
```

```
#include<signal.h>
```

```
void gogogo(int voidv)
```

```
{
```

```
    printf("SIGINT Accur!\n");
```

```
    exit(0);
```

```
}
```

```
int main(void)
```

```
{
```

```
    signal(SIGINT ,gogogo);
```

```
    for(;;)
```

```
    {
```

```
        printf("kill Test\n");
```

```
        sleep(2);
```

```
    }
```

```
    return 0;
```

```
}
```

```
/*
```

```
gcc -o test test.c
```

```
./kill &
```

```
*/
```

kill2.c

```
#include<signal.h>
```

```
#include<stdio.h>
```

```
#include<unistd.h>
```

```
#include<fcntl.h>
```

```
int main(int argc, char* argv[])
```

```

{
    if(argc < 2)
        printf("Usage : ./exe pid\n");
    else
        kill(atoi(argv[1]),SIGINT);
    return 0;
}

```

```

/*
    gcc -o kill kill.c
    ./kill pid[숫자]
*/

```

```

thread.c
#include<stdio.h>
#include<pthread.h>

```

```

void* task1(void* X)
{
    printf("Thread A Complete\n");
}

```

```

void* task2(void* X)
{
    printf("Thread B complete\n");
}

```

```

int main(void)
{
    pthread_t ThreadA, ThreadB;

```

pthread_create(&ThreadA, NULL, task1, NULL); // 쓰레드 A에
이와 쓰레드 B의 값을 채워넣는다 2 번째는 3 번째인자 태스크 1 은 구동시

킬 함수다보이드포인터가 쓰이는이유는 무엇이든 인자로 받고 무엇이든 리턴한다쓰레드비가태스크 2 를 구동시킨다 쓰레드를 등록해주는거

pthread_create(&ThreadB, NULL, task2, NULL);// 조인을 하는순간 메모리에 올라가는거

```
pthread_join(ThreadA, NULL);
pthread_join(ThreadB, NULL);

return 0;
}
```

basic_client.c

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<unistd.h>
#include<arpa/inet.h>
#include<sys/socket.h>
```

```
typedef struct sockaddr_in si;
typedef struct sockaddr * sap;
```

```
void err_handler(char* msg)
{
    fputs(msg, stderr);
    fputc('\n', stderr);
    exit(1);
}
```

```
int main(int argc, char* argv[]){
```

```
    int serv_sock;
    int clnt_sock;
```

```

si serv_addr;
si clnt_addr;
socklen_t clnt_addr_size;

char msg[] = "Hello Network Programming";
if(argc != 2)
{
    printf("use: %s <port>\n",argv[0]);
    exit(1);
}

serv_sock = socket(PF_INET, SOCK_STREAM, 0);

if(serv_sock == -1)
    err_handler("socket() error");

memset(&serv_addr, 0, sizeof(serv_addr));
serv_addr.sin_family = AF_INET;
serv_addr.sin_addr.s_addr = htonl(INADDR_ANY);
serv_addr.sin_port = htons(atoi(argv[1]));
//스코프바인딩이란해당스코프지정하겠다
if(bind(serv_sock, (sap)&serv_addr, sizeof(serv_addr)) == -1)
    err_handler("bind() error");

if(listen(serv_sock, 5) == -1)//서브속리슨
    err_handler("listen() error");

clnt_addr_size = sizeof(clnt_addr);//32 나올것임
clnt_sock = accept(serv_sock, (struct sockaddr *)&clnt_addr,
& clnt_addr_size);//소켓이 클라이언트를 기다림

if(clnt_sock == -1)
    err_handler("accept() error");

```

```

        write(clnt_sock, msg, sizeof(msg)); //클라이언트소켓에다가 메세
지를 보낸다. 소켓은 원격에있는 파일인데 원격으로 세마포어를 만드는거임
// 원격에있는클라이언트에라이트함
        close(clnt_sock);
        close(serv_sock);

        return 0;
}

```

basic_client.c

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<unistd.h>
#include<arpa/inet.h>
#include<sys/socket.h>

typedef struct sockaddr_in si;
typedef struct sockaddr * sap;

void err_handler(char* msg)
{
    fputs(msg, stderr);
    fputc('\n', stderr);
    exit(1);
}

int main(int argc, char** argv)
{
    int sock;
    int str_len = 0;
    si serv_addr;
    char msg[32] = {0};

```

```
int idx = 0, read_len = 0;
```

```
if(argc != 3)
{
    printf("use : %s <IP> <port>\n", argv[0]);
    exit(1);
}
```

```
sock = socket(PF_INET, SOCK_STREAM, 0);
```

```
if(sock == 0)
    err_handler("socket() error");
```

```
memset(&serv_addr, 0, sizeof(serv_addr));
serv_addr.sin_family = AF_INET;
serv_addr.sin_addr.s_addr = inet_addr(argv[1]);
serv_addr.sin_port = htons(atoi(argv[2]));
```

```
if(connect(sock, (sap)&serv_addr, sizeof(serv_addr)) == -1)
    err_handler("connect() error");
```

```
while(read_len = read(sock, &msg[idx++], 1))
{
    if(read_len == -1)
        err_handler("read() error!");

    str_len += read_len;
}
```

```
//중간에 끊기는 현상때문에 이 코드를 삽입함. 데이터 손실 없이 보내
```

려고

```
printf("msg from serv: %s\n", msg);
printf("read count: %d\n", str_len);
close(sock);
```



```
    return 0;
```

```
}
```

```
basic_client.c
```

```
#include<stdio.h>
```

```
#include<fcntl.h>
```

```
#include<stdlib.h>
```

```
#include<string.h>
```

```
#include<arpa/inet.h>
```

```
#include<sys/socket.h>
```

```
#include<unistd.h>
```

```
typedef struct sockaddr_in si;
```

```
typedef struct sockaddr * sap;
```

```
void err_handler(char* msg)
```

```
{
```

```
    fputs(msg, stderr);
```

```
    fputc('\n', stderr);
```

```
    exit(1);
```

```
}
```

```
int main(int argc, char* argv[])
```

```
{
```

```
    int sock;
```

```
    int str_len;
```

```
    si serv_addr;
```

```
    char msg[32];
```

```
    if(argc != 3)
```

```
    {
```

```
        printf("use : %s <IP> <port>\n", argv[0]);
```

```
        exit(1);
```

```
    }
```

```

//내가통신할수있는파일디스크립터를가져와서 소켓에다가 집어넣음
sock = socket(PF_INET, SOCK_STREAM,0);

if(sock == -1)
{
    err_handler("socket() error");
}
memset(&serv_addr, 0, sizeof(serv_addr));//멤셋 초기화
serv_addr.sin_family = AF_INET;//티씨피쓸건지유디피쓸건지
serv_addr.sin_addr.s_addr = inet_addr(argv[1]);
serv_addr.sin_port = htons(atoi(argv[2]));

if(connect(sock, (sap)&serv_addr, sizeof(serv_addr)) == -1)//
커넥트하면 서버의리슨에서 받아서 액셉트함
    err_handler("connect() error");

str_len = read(sock, msg, sizeof(msg) -1);

if(str_len == -1)
    err_handler("read() error!");
printf("msg from serv : %s\n",msg);
close(sock);

return 0;
}

```

inet_addr.c

```
#include<stdio.h>
```

```
#include<arpa/inet.h>
```

//엔디안크로스매칭할때 사람이쓰는거말고 기계가 알아듣게 형을 변환해주는코드

```
int main(int argc, char*argv[])
```

```
{
```

```
    char* addr1 = "3.7.5.9";
```

```

char* addr2 = "1.3.5.7";

unsigned long conv_addr = inet_addr(addr1);
if(conv_addr == INADDR_NONE)
    printf("Error!\n");
else
    printf("Network  Ordefed  Integer  Addr:  %#lx\n",
conv_addr);

conv_addr = inet_addr(addr2);
if(conv_addr == INADDR_NONE)
    printf("Error!\n");
else
    printf("Network      Ordered      Integer      Addr:
%#lx\n",conv_addr);

return 0;
}

```

sock_fd.c//소켓도 fd 처럼 반환해준다는 것을 보여주는 코드

```

#include<stdio.h>
#include<fcntl.h>
#include<unistd.h>
#include<sys/socket.h>

int main(void)
{
    int fd[3];
    int i;

    fd[0] = socket(PF_INET, SOCK_STREAM, 0);//TCP/IP
    fd[1] = socket(PF_INET, SOCK_DGRAM,0);//UDP
    fd[2] = open("test.txt",O_CREAT|O_WRONLY|O_TRUNC);

    for(i = 0; i < 3; i++)

```

```
{  
    printf("fd[%d] = %d\n",i , fd[i]);  
}  
for(i = 0; i < 3; i++)  
    close(fd[i]);  
return 0;  
}
```