

Xilinx Zynq FPGA, TI DSP, MCU기반의 프로그래밍 및 회로 설계 전문가 과정

강사 - Innov (이상훈)

gcccompil3r@gmail.com

학생 - 이유성

dbtjd1102@naver.com

gethostbyaddr.c ip주소로 호스트(도메인)찾기

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

typedef struct sockaddr_in si;

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

int main(int argc , char **argv)
{
    int i;
    si addr;
    struct hostent *host;

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

    memset(&addr , 0,sizeof(addr));
    addr.sin_addr.s_addr = inet_addr(argv[1]);
    host = gethostbyaddr((char *)&addr.sin_addr ,4,AF_INET);//실제 아이피값을 가지고
    호스트를 찾는다.

    if(!host)
        err_handler("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]);

    (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;
}

//gethost error 방화벽이 있다면 뜰 수 있다 kns.kornet.net 를쳐서 이 아이피를 넣어봐라.
```

```

mpecho_serv.c
// mpecho_serv.c
//포크를 이용해서 제어를 하는 방식

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

typedef struct sockaddr_in si;
typedef struct sockaddr *sap;

#define BUF_SIZE 32

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

void read_childproc(int sig)
{
    pid_t pid;
    int status;
    pid = waitpid(-1 , &status , WNOHANG); //자식상태를.
    printf("Removed proc id : %d\n " ,pid);
}

int main(int argc , char **argv)
{
    int serv_sock , clnt_sock;
    si serv_addr , clnt_addr;
    pid_t pid;
    struct sigaction act; // 시그널 대체.
    socklen_t addr_size;
    int str_len, state;
    char buf[BUF_SIZE] = {0};

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

```

```

act.sa_handler = read_childproc;
sigemptyset(&act.sa_mask);
act.sa_flags = 0;
state = sigaction(SIGCHLD,&act , 0);// 1인자를 받으면 2인자 실행.

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]));

if(bind(serv_sock, (sap)&serv_addr,sizeof(serv_addr)) == -1)
    err_handler("bind() error");

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

for(;;)
{
    addr_size = sizeof(clnt_addr);
    clnt_sock = accept(serv_sock, (sap)&clnt_addr , &addr_size);
    if(clnt_sock == -1)
        continue ;
    else
        puts("New Client Connected....");

    pid = fork();
    if(pid == -1)
    {
        close(clnt_sock);
        continue;
    }
    if(pid == 0 )//자식
    {
        close(serv_sock);//자식프로세스는 소켓안필요.해서 닫음.

        while((str_len = read(clnt_sock,buf,BUF_SIZE)) !=0)//다시짜줌echo
        클라이언트 들어올때까지 영원히 기다림..
            write(clnt_sock , buf ,str_len);
        //클라이언트에서 읽은 것을 다시 클라에 보냄
        close(clnt_sock);//
        puts("Client Disconnected...");
        return 0;
    }
    else
        close(clnt_sock);
}

```

```
    }  
    close(serv_sock);  
  
    return 0;  
}
```

mpecho_clnt.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;  
  
#define BUF_SIZE 32  
  
void err_handler(char *msg)  
{  
    fputs(msg,stderr);  
    fputc('\n',stderr);  
    exit(1);  
}  
  
void read_routine(int sock, char *buf)  
{  
    for(;;)  
    {  
        int str_len = read(sock,buf,BUF_SIZE);  
  
        if(str_len ==0)  
            return;  
        buf[str_len] = 0;  
        printf("msg from server : %s",buf);  
    }  
}  
  
void write_routine(int sock,char *buf)  
{  
    for(;;)  
    {  
        fgets(buf,BUF_SIZE,stdin);
```

```

        if(!strcmp(buf,"q\n") || !strcmp(buf,"Q\n")) //q나 Q면 끄고
        {
            shutdown(sock,SHUT_WR);
            return ;
        }
        write(sock,buf,strlen(buf));          //그 외는 적어.
    }
}
int main(int argc , char **argv)
{
    pid_t pid;
    int i , sock;
    si serv_addr;
    char buf[BUF_SIZE] = {0};

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

    if(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 = 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");
    else
        puts("connected....");

    pid = fork();

    if(pid ==0) //자식은 write 내가 키보드에 입력한 것을 서버에 송신.
        write_routine(sock,buf);
    else //부모는 read 서버에서 날아온거 읽어서 뿌려줌.
        read_routine(sock,buf);
    close(sock);

    return 0;
}

```