

TI DSP, MCU 및 Xilinx Zynq FPGA 프로그램 전문가 과정

강사 - Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 - 은태영

zero_bird@naver.com

file server

```
tewill@tewill-B85M-D3H: ~/Downloads
#include <fcntl.h>
#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);
}

int main(int argc, char **argv)
{
    int serv_sock, clnt_sock, fd;
    char buf[BUF_SIZE] = {0};
    int read_cnt;

    si serv_addr, clnt_addr;
    socklen_t clnt_addr_size;

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

1,1 Top

```
tewill@tewill-B85M-D3H: ~/Downloads

fd = open("c_01.c", O_RDONLY);
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");

clnt_addr_size = sizeof(clnt_addr);

clnt_sock = accept(serv_sock, (sap)&clnt_addr, &clnt_addr_size);
```

35,0-1 58%

- ❖ 기존 서버 형식과 동일하다.
- ❖ 추가적으로 fd 를 통하여 파일을 열고 인덱스를 받는다.

file server

```
tewill@tewill-B85M-D3H: ~/Downloads
for(;;)
{
    read_cnt = read(fd, buf, BUF_SIZE);

    if(read_cnt < BUF_SIZE)
    {
        write(clnt_sock, buf, read_cnt);
        break;
    }

    write(clnt_sock, buf, BUF_SIZE);
}

shutdown(clnt_sock, SHUT_WR);
read(clnt_sock, buf, BUF_SIZE);
printf("msg from client : %s\n", buf);

close(fd);
close(clnt_sock);
close(serv_sock);

return 0;
}
```

56,0-1 Bot

- ❖ read 를 통하여 파일 내용을 BUF_SIZE 만큼 buf 에 저장하고, 읽은 값을 리턴 한다.
- ❖ write 를 통해 buf 의 내용을 clnt_sock 에 쓴다.
- ❖ shutdown() : close 와 동일하지만, 특정 기능만 닫을 수 있다.
- ❖ SHUT_WR 을 닫아서 clnt_sock 의 쓰기를 사용하지 못하게 된다.
- ❖ 그 후, read 를 통해 문자열을 buf 에 저장 후, 출력한다.

file client

```
tewill@tewill-B85M-D3H: ~/Downloads
#include <fcntl.h>
#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);
}

int main(int argc, char **argv)
{
    char buf[BUF_SIZE] = {0};
    int fd, sock, read_cnt;
    si serv_addr;

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

1,1 Top

- ❖ 기존 client 와 동일한 형식이다.
- ❖ 추가적으로 파일을 받을 fd 가 존재한다.

file client

```
tewill@tewill-B85M-D3H: ~/Downloads

fd = open("receive.txt", O_CREAT | O_WRONLY);
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, (&serv_addr), sizeof(serv_addr)) == -1)
    err_handler("connect() error");
else
    puts("Connected.....");

while((read_cnt = read(sock, buf, BUF_SIZE)) != 0)
    write(fd, buf, read_cnt);

puts("Received File Data");
write(sock, "Thank you", 10);
close(fd);
close(sock);

return 0;
```

58,1 Bot

- ❖ 쓰기할 파일 “receive.txt” 를 생성 후 연다.
- ❖ 기존 client 와 동일한 방식으로 처리한다.
- ❖ connect 후, read 와 write 를 통하여 fd 에 쓰기를 시작한다.
- ❖ 쓰기 작업을 완료 후, 텍스트를 출력하고, “thank you” 를 서버에 보낸다.

IP 주소

```
tewill@tewill-B85M-D3H: ~/Downloads
#include <unistd.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <stdio.h>
#include <stdlib.h>

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

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

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

    host = gethostbyname(argv[1]);

    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]);

    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;
}
-- INSERT --
```

- ❖ host 를 만든 후, 인자로 받은 호스트 이름을 gethostbyname() 으로 저장한다.
- ❖ 가장 먼저 host 이름을 출력한다.
- ❖ 그 후, 별칭이 있는지 h_aliases 를 통해 확인하고 출력한다.
- ❖ addrtype 을 통하여 IP 의 버전을 확인한다.
- ❖ 해당 IP 주소를 출력한다.

