TI DSP, MCU 및 Xilinx Zynq FPGA
프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 – 문한나

mhn97@naver.com

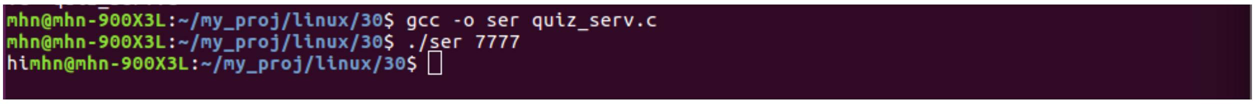
예제 1)

<quiz_serv.c>

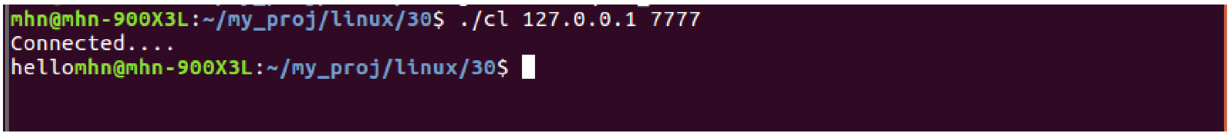
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <signal.h>
#include <sys/wait.h>
#include <arpa/inet.h>
#include <sys/socket.h>

typedef struct sockaddr_in si;
typedef struct sockaddr *sap;

//#define BUF_SIZE 32

struct ssend{

	int a;
	float b;

};
typedef struct ssend SSend;

void err_handler(char *msg){

	fputs(msg,stderr);
	fputc('\n',stderr);
	exit(1);

}

int main(int argc,char **argv){

	int serv_sock,clnt_sock,len;
	si serv_addr,clnt_addr;
	socklen_t addr_size;
	socklen_t clnt_addr_size;

	int strlen,clmsg;
//	char buf[BUF_SIZE] = {0};
	char buf[1024];
	char msg[] = "hello";

	if(argc != 2){

		printf("use: %s <port>\n",argv[0]);
		exit(1);
	}
	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]));

	serv_sock = socket(PF_INET, SOCK_STREAM, 0);

	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);

	clmsg = read(clnt_sock,buf,sizeof(buf)); //클라이언트에서 온 메시지 읽고
	write(1,buf,clmsg); //출력한다

	write(clnt_sock,msg,sizeof(msg)); //클라이언트에게 메시지 전송

	close(clnt_sock);
	close(serv_sock);

	return 0;

}
```

<quiz_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;

//struct SSend *msg;

//#define BUF_SIZE 32

void err_handler(char *msg){

	fputs(msg,stderr);
	fputc('\n',stderr);
	exit(1);

}

int main(int argc,char **argv){
        int header[2];

        int i,sock,msg1;
        si serv_addr;
//      char buf[BUF_SIZE] = {0};
        char buf[1024];

        char sermsg[] = "hi";

        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(atol(argv[2]));

        if(connect(sock, (sap)&serv_addr, sizeof(serv_addr)) == -1)
                err_handler("connect() error");
        else
                puts("Connected....");

        write(sock,sermsg,sizeof(sermsg)); //서버에게 메지지를 보낸다

        msg1 = read(sock,buf,sizeof(buf)); //서버에서 온 메시지를 읽고
        write(1,buf,msg1); //출력한다

        close(sock);

        return 0;
}
```



```
mhn@mhn-900X3L:~/my_proj/linux/30$ gcc -o ser quiz_serv.c
mhn@mhn-900X3L:~/my_proj/linux/30$ ./ser 7777
himhn@mhn-900X3L:~/my_proj/linux/30$ 
```



```
mhn@mhn-900X3L:~/my_proj/linux/30$ ./cl 127.0.0.1 7777
Connected....
hellomhn@mhn-900X3L:~/my_proj/linux/30$ 
```

예제 2)

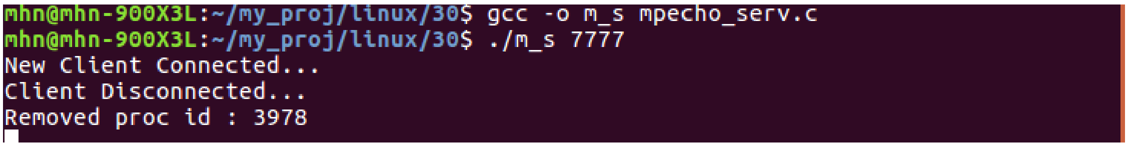
<mpehco_serv.c>

```
#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){ //자식프로세스죽으면 죽었다고 얘기해줌

        pid_t pid;
        int status;
        pid = waitpid(-1, &status, WNOHANG);
        printf("Removed proc id : %d\n",pid); //죽은 프로세스의  pid 를 뿌림

}

int main(int argc,char **argv){

        int serv_sock, clnt_sock;
        si serv_addr, clnt_addr;
        pid_t pid;
        struct sigaction act; //시그액션은 시그널 대체가능
        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;
```

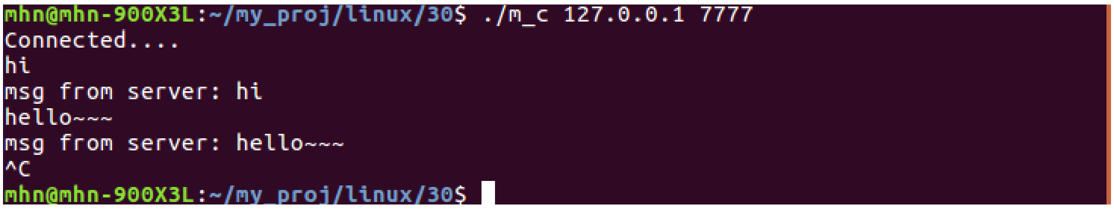
```
    sigemptyset(&act.sa_mask);
    act.sa_flags = 0;
    state = sigaction(SIGCHLD,&act,0); //구동시키렴

    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]));

    serv_sock = socket(PF_INET, SOCK_STREAM, 0);

    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(;;){ //여러사람이 동시에 접속하면 갱신이 바로 안됨

    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); //서버소켓 필요없음
        while((str_len = read(clnt_sock,buf,BUF_SIZE)) != 0) //리드는 블로킹이라서 안끝남
            write(clnt_sock,buf,str_len);

        close(clnt_sock);
        puts("Client Disconnected...");
        return 0;
    }else
        close(clnt_sock);

    }

    close(serv_sock);
    return 0;

}
```

<mpehco_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){ //서버에서 날아오는 메시지 뿌려줌

    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){ //키보드 입력한 것 서버로 보냄

    for(;;){

        fgets(buf,BUF_SIZE,stdin);

        if(!strcmp(buf,"q\n") || !strcmp(buf,"Q\n")){
            shutdown(sock,SHUT_WR); //전송이 끝나면 닫는다
            return;
        }
        write(sock,buf,strlen(buf));
    }
}

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(atol(argv[2]));

	if(connect(sock, (sap)&serv_addr, sizeof(serv_addr)) == -1)
		err_handler("connect() error");
	else
		puts("Connected....");

	pid = fork();

	if(pid == 0) //자식
		write_routine(sock,buf);
	else
		read_routine(sock,buf);

	close(sock);

	return 0;

}
```



```
mhn@mhn-900X3L:~/my_proj/linux/30$ gcc -o m_s mpecho_serv.c
mhn@mhn-900X3L:~/my_proj/linux/30$ ./m_s 7777
New Client Connected...
Client Disconnected...
Removed proc id : 3978
```



```
mhn@mhn-900X3L:~/my_proj/linux/30$ ./m_c 127.0.0.1 7777
Connected....
hi
msg from server: hi
hello~~~
msg from server: hello~~~
^C
mhn@mhn-900X3L:~/my_proj/linux/30$
```

int shutdown(int *sockfd*, int *how*);

헤더 - #include <sys/socket.h>

첫번째 인자
The **shutdown**() call causes all or part of a full-duplex connection on the socket associated with *sockfd* to be shut down.
두번째 인자
If *how* is **SHUT_RD**, further receptions will be disallowed. If *how* is **SHUT_WR**, further transmissions will be disallowed. If *how* is **SHUT_RDWR**, further receptions and transmissions will be disallowed.

반환값 - On success, zero is returned. On error, -1 is returned, and *errno* is set appropriately.

예제 3)

<gethostbyaddr.c>

```
#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("Usage: %s <port>\n", argv[0]);
                exit(1);

        }

        memset(&addr, 0, sizeof(addr));
        addr.sin_addr.s_addr = inet_addr(atoi(argv[1]));
        host = gethostbyaddr((char *)&addr.sin_addr,4,AF_INET);

        if(!host)
                err_handler("gethost error!");

        printf("Official Name: %s\n",host→h_name); //Official name of host.

        for(i = 0; host->h_aliases[i]; i++)
                printf("Aliaese %d: %s\n", i + 1, host->h_aliases[i]);  //Alias list.
        (host->h_addrtype == AF_INET ? "AF_INET" : "AF_INET6"); //Host address type.


        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]));
//List of addresses from name server.
        return 0;

}
```

gethostbyaddr() : 2 진수 IP 주소를 도메인 네임로 변환하는 함수

```
struct hostent *gethostbyaddr(const void *addr,
                              socklen_t len, int type);
```

헤더 : **`#include <sys/socket.h> /* for AF_INET */`**

반환값 : 해당 호스트의 정보를 가진 hostent 구조체를 리턴

The **gethostbyaddr**() function returns a structure of type *hostent* for the given host address *addr* of length *len* and address type *type*. Valid address types are **AF_INET** and **AF_INET6**. The host address argument is a pointer to a struct of a type depending on the address type, for example a *struct in_addr ** (probably obtained via a call to **inet_addr**(3)) for address type **AF_INET**.

※ 호스트의 정보를 얻기 위해서는 DNS 서버에서 Reverse DNS 가 수행되어야 한다.

※ 그런데 최근 보안문제로 대부분의 DNS 서버가 Reverse DNS 기능을 제공하지 않는다.

```
struct hostent
{
  char *h_name;                 /* Official name of host.  */
  char **h_aliases;             /* Alias list.  */
  int h_addrtype;               /* Host address type.  */
  int h_length;                 /* Length of address.  */
  char **h_addr_list;           /* List of addresses from name server.  */
#define h_addr  h_addr_list[0]  /* Address, for backward compatibility.  */
};
```