TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 – 문한나

mhn97@naver.com

예제 1)

<file_server.c>

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

	fd = open("file_server.c",O_RDONLY); //파일디스크립터를 얻어온다
	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);

	for(;;){

		read_cnt = read(fd,buf,BUF_SIZE); //파일을 읽어서 바이트 수를  read_cnt 에 넣는다

		if(read_cnt < BUF_SIZE){ //파일의 크기만큼 계속 읽는다

			write(clnt_sock,buf,read_cnt);
			break;
		}
		write(clnt_sock,buf,BUF_SIZE);
	}

	shutdown(clnt_sock,SHUT_WR); //전송이 끝나면 닫는다
	read(clnt_sock,buf,BUF_SIZE); //클라이언트에서 보내온 데이터를 읽는다
	printf("msg from client: %s\n",buf); //그 데이터를 출력한다

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

	return 0;

}
```

<file_client.c>

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

```
		exit(1);
	}

	fd = open("receive.txt",O_CREAT | O_WRONLY);  //새로운 파일을 생성하고 그 파일에 대한 디스크립터
를 얻는다
	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 = htonl(INADDR_ANY);
	serv_addr.sin_port = htons(atoi(argv[2]));

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

	while((read_cnt = read(sock, buf, BUF_SIZE)) != 0) //서버가 보낸 데이터의 크기만큼 읽는다
		write(fd, buf, read_cnt); //읽은 데이터를 새로 생성한 파일에 적는다

	puts("Received File Dtat");
	write(sock, "Than you",10); //서버에게 전송 할 메시지
	close(fd);
	close(sock);

	return 0;

}
```

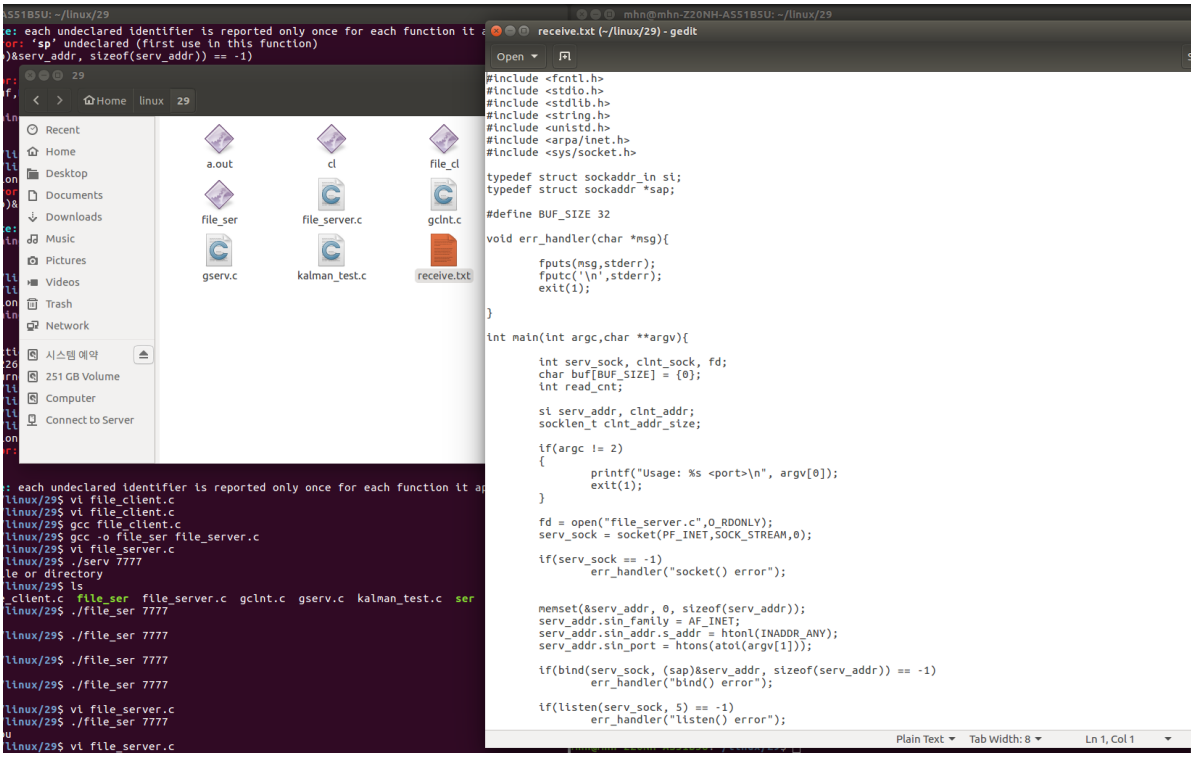
<file_server.c>

```
mhn@mhn-Z20NH-AS51B5U:~/linux/29$ ./file_ser 7777
msg from client: Than you
```

<file_client.c>

```
mhn@mhn-Z20NH-AS51B5U:~/linux/29$ ./file_cl 127.0.0.1 7777
Conneted--------
Received File Dtat
```

<receive.txt> 서버에서 보낸 데이터가 새로 생성된 파일에 담겨있음



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 2)

```
#include <unistd.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <string.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); // 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 (별칭 없을수도)

	printf("Address Type: %s\n",(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;
}
```

```
mhn@mhn-Z20NH-AS51B5U:~/linux/29$ ./a.out naver.com
Official Name: naver.com
Address Type: AF_INET
IP Addr 1: 210.89.160.88
IP Addr 2: 210.89.164.90
IP Addr 3: 125.209.222.141
IP Addr 4: 125.209.222.142
mhn@mhn-Z20NH-AS51B5U:~/linux/29$ vi gethostbyname.c
mhn@mhn-Z20NH-AS51B5U:~/linux/29$ ./a.out google.com
Official Name: google.com
Address Type: AF_INET
IP Addr 1: 216.58.220.206
```

getgostbyname() - 도메인 이름으로 hostent 정보를 구함

```
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.  */
};
```

헤더 netdb.h
형태 struct hostent *gethostbyname(const char *name);
인수 호스트 이름이거나 표준 점 표기법의 IPv4 주소, 콜론(그리고 점 표기법도 가능)표기법의 IPv6
반환 성공 → hostent 구조체
에러 → h_errno 변수에 에러 넘버 대입