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

#27

강사:Innova Lee(이 상훈)

학생: 김시윤

수업내용 복습

echo_server.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 1024

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


int main(int argc, char **argv)
{
	int i, str_len;
	int serv_sock, clnt_sock;

	char msg[BUF_SIZE];

	si serv_addr, clnt_addr;
	socklen_t clnt_addr_size;

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

	serv_sock = socket(PF_INET, SOCK_STREAM, 0);
//ipv4 TCP 통신

	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);//어느것이든 받겠다
	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)//접속허용클라이언트 5 개
		err_handler("listen() error");

	clnt_addr_size = sizeof(clnt_addr);

	for(i =0; i<5; i++)
	{
		clnt_sock = accept(serv_sock, (struct sockaddr *)&clnt_addr,
&clnt_addr_size);
//사용자가 들어오면 승인해준다. 클라이언트 전용소켓이 생김

		if(clnt_sock == -1)
			err_handler("accept() error");
		else
			printf("Connected Client %d\n", i + 1);

		while((str_len = read(clnt_sock, msg, BUF_SIZE)) !=0)
```

```
//클라이언트에서 읽은 메시지를 저장한다. read 가 block 이라 한 서버밖에 접근불가 nonblock 해줄라면 fork 해줘야함 (구현 못함 ㅜㅜ)
                        write(clnt_sock, msg, str_len);
//클라이언트에 서버의 메시지에 들어있는값을 적는다, 서버에 메시지에는 클라이언트에서 읽은값이 들어있으므로 클라이언트에서 입력한 문자열에 되돌아간다.
                close(clnt_sock);
        }
        close(serv_sock);

        return 0;
}
```

```
#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 1024

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

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

```
        char *m = "Input number(q to quit): ";

        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 .....");

        for(;;)
        {
                fputs("Input number(q to quit): " ,stdout);//write
                fgets(msg,BUF_SIZE, stdin);//read

                if(!strcmp(msg, "q\n") || !strcmp(msg, "Q\n"))
                        break;
                //q 가 입력되면 클라이언트가 포문을 나가 닫는다.

        write(sock, msg, strlen(msg));
        str_len = read(sock, msg, BUF_SIZE -1);


        if(str_len == -1)
```

```
                err_handler("read() error!");

                msg[str_len] = 0;

                printf("msg from serv: %s\n",msg);
        }
        close(sock);

        return 0;
}
```

op_server.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 1024
#define OPSZ 4
//operation size 4
void err_handler(char *msg)
{
        fputs(msg, stderr);
        fputc('\n',stderr);
        exit(1);
}

int calculate(int opnum, int *opnds, char op)
{
        int result = opnds[0], i;

        switch(op)
        {
                case '+':
                        for(i= 1; i < opnum; i++)
//opnds[0] 에 result 를 저장했기 때문에 i 는 1 부터
                                result += opnds[i];
                        break;
                case '-':
                        for(i= 1; i < opnum; i++)
                                result -= opnds[i];
                        break;
                case '*':
                        for(i= 1; i< opnum; i++)
                                result *= opnds[i];
                        break;
        }

        return result;
}

int main(int argc, char **argv)
{
        int serv_sock, clnt_sock;
        char opinfo[BUF_SIZE];

        int result , opnd_cnt, i;
        int recv_cnt , recv_len;

        si serv_addr , clnt_addr;
        socklen_t clnt_addr_size;

        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]));

	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);

	for(i =0; i < 5; i++)
	{
		opnd_cnt = 0;
		clnt_sock = accept(serv_sock, (sap)&clnt_addr,
&clnt_addr_size);
		read(clnt_sock, &opnd_cnt, 1);
	//입력하는 문자열이 숫자라서 1 바이트씩 인트형으로 받는게 가능한건지??..
		recv_len = 0;

		while((opnd_cnt * OPSZ +1) > recv_len)
//연산하는게 정수형이라 사이즈 4
//더블형이라면 사이즈 8 로 바꿔야함
		{
			recv_cnt = read(clnt_sock, &opinfo[recv_len],
BUF_SIZE -1);
//소켓에서 읽어서 렌에 채움 recv_cnt 읽은 바이트 수 (read 의 리턴 읽은 바이트 수)
			recv_len += recv_cnt;
//읽은 바이트 수를 len 에 넣음
```

```
//opinfo 우리가 입력한 숫자가 들어와있음
		}

			result = calculate(opnd_cnt, (int *)opinfo,
opinfo[recv_len -1]);
//calculate 계산하는 작언 3 을 입력하면 3 이 들어가있음 갯수 , 읽어온거 , 12 바이트 읽었
다고 하면 배열의 시작은 0~11 이라 -1 해
			write(clnt_sock, (char *)&result, sizeof(result));
//&result 결과값이 클라이언트로 넘어감
			close(clnt_sock);
		}
		close(serv_sock);

		return 0;
}
```

op_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;

#define BUF_SIZE 1024
#define RLT_SIZE 4
#define OPSZ 4

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

```
        exit(1);
}

int main(int argc, char **argv)
{
        int i,sock, result, opnd_cnt;
        char opmsg[BUF_SIZE] = {0};
        si serv_addr;

        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.....");

        fputs("Operand Cnt: ", stdout);//write 와 동일
        scanf("%d", &opnd_cnt);//read or scanf 와 동일

        opmsg[0] = (char)opnd_cnt;
        //opmsg0 번에 opnd_cnt 저장
        for(i = 0; i< opnd_cnt; i++)
        {
                printf("Operand %d: ", i+1);
                scanf("%d", (int *)&opmsg[i*OPSZ +1]);
        }

        fgetc(stdin);//1byte 만 읽음
        fputs("Operator: ", stdout);
        scanf("%c",&opmsg[opnd_cnt * OPSZ + 1]);//opsmg 에 모든게 다들어있음.
        write(sock, opmsg, opnd_cnt * OPSZ + 2);
        read(sock, &result, RLT_SIZE);//서버가 라이트한 값이 클라이언트에 넘어옴
계산은 서버가함

        printf("Operation result: %d\n", result);
        close(sock);

        return 0;
}
```