

TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정



2018.04.04

30 일차

강사 - Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 - 신민철

akrn33@naver.com

```
load_test_clnt.c//구조체를 넘기는 클라이언트 코드
#include "common.h"
#include <signal.h>
#include <setjmp.h>
```

```
jmp_buf env;
int tmp_sock;
```

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

```
void read_proc(int sock, d *buf)
{
    for(;;)
    {
        int len = read(sock, buf, BUF_SIZE);

        if(!len)
            return;

        printf("msg from serv: %d, %f\n", buf->data, buf-
>fdata);
    }
}
```

```
void quit_proc(int signo)
{
    printf("Exited!\n");
    shutdown(tmp_sock, SHUT_WR);
    longjmp(env, 1);
}
```

```

void write_proc(int sock, d *buf)
{
    char msg[32] = {0};

    tmp_sock = sock;
    signal(SIGINT, quit_proc);

    for(;;)
    {
#ifdef DEBUG
        fgets(msg, BUF_SIZE, stdin);
#endif
        buf->data = 3;
        buf->fdata = 7.7;

        write(sock, buf, sizeof(d));
    }
}

int main(int argc, char **argv)
{
    pid_t pid;
    int i, sock;
    si serv_addr;
    d struct_data;
    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, (sp)&serv_addr, sizeof(serv_addr)) == -1)
    err_handler("connect() error");
else
    puts("Connected!\n");

pid = fork();

if(!pid)
{
    int ret;

    if((ret = setjmp(env)) == 0)
        ;
    else if(ret > 0)
        goto end;

    write_proc(sock, (d *)&struct_data);
}
else
    read_proc(sock, (d *)&struct_data);

end:
close(sock);

return 0;
}

```

```

load_test_serv.c//채팅비율을 볼 수 있는 서버코드
#include "common.h"
#include "load_test.h"

#include <signal.h>
#include <sys/wait.h>

typedef struct sockaddr_in    si;
typedef struct sockaddr *     sp;

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

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

int main(int argc, char **argv)
{
    int serv_sock, clnt_sock, len, state;
    char buf[BUF_SIZE] = {0};
    si serv_addr, clnt_addr;
    struct sigaction act;
    socklen_t addr_size;
    d struct_data;
    pid_t pid;

```

```

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

act.sa_handler = read_cproc;
sigemptyset(&act.sa_mask);
act.sa_flags = 0;
state = sigaction(SIGCHLD, &act, 0);

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, (sp)&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, (sp)&clnt_addr,
&addr_size);

    if(clnt_sock == -1)
        continue;

```

```

else
    puts("New Client Connected!\n");

pid = fork();

if(pid == -1)
{
    close(clnt_sock);
    continue;
}

if(!pid)
{
    int cnt = 0;
    tv start, end;
    double runtime = 0.0;
    double load_ratio;

    close(serv_sock);

    for(;;)
    {
        gettimeofday(&start, NULL);

        len = read(clnt_sock, (d *)&struct_data,
BUF_SIZE);

        printf("struct.data = %d, struct.fdata = %f\n",
struct_data.data, struct_data.fdata);
        write(clnt_sock, (d *)&struct_data, len);

        gettimeofday(&end, NULL);

        runtime = get_runtime(start, end);
        cnt++;
    }
}

```

```

        load_ratio = cnt / runtime;
        printf("load_ratio = %lf\n", load_ratio);
    }
    #if 0
        while((len = read(clnt_sock, (d *)&struct_data,
BUF_SIZE)) != 0)
        {
            printf("struct.data = %d, struct.fdata = %f\n",
struct_data.data, struct_data.fdata);
            write(clnt_sock, (d *)&struct_data, len);
        }
    #endif

        close(clnt_sock);
        puts("Client Disconnected!\n");
        return 0;
    }
    else
        close(clnt_sock);
}
close(serv_sock);

return 0;
}

```