TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 - 최대성
c3d4s19@naver.com

```
2018.04.05. - 031일차 수업
```

기본 TCP/IP통신 헤더파일 이용하여 나누기

<init_sock.h 파일>

```
#ifndef __INIT_SOCK_H__
#define __INIT_SOCK_H__
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <arpa/inet.h>
#include <sys/socket.h>
typedef struct sockaddr_in si;
typedef struct sockaddr* sap;
socklen_t clnt_addr_size = sizeof(si);
# define SERV 0
# define CLNT 1
void err_handler(char* msg);
void init_sock_addr(int type, char** argv, struct
sockaddr_in addr, int* sock);
#endif
```

<init_sock.c 파일>

```
#include "init_sock.h"

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

void init_sock_addr(int type, char** argv, si
addr, int* sock){

    *sock = socket(PF_INET, SOCK_STREAM, 0);
    if(*sock == -1)
        err_handler("socket() error");

    memset(&addr, 0, sizeof(addr));
```

```
addr.sin family = AF INET;
    if(type == SERV){
        addr.sin_addr.s_addr = htonl(INADDR_ANY);
        addr.sin port = htons(atoi(argv[1]));
        if(bind(*sock, (sap)&addr, sizeof(addr))
== -1)
                err_handler("bind() error");
        if(listen(*sock, 5) == -1)
                err handler("listen() error");
    }
   else if(type == CLNT){
        addr.sin_addr.s_addr = inet_addr(argv[1]);
        addr.sin_port = htons(atoi(argv[2]));
        if(connect(*sock, (sap)&addr,
sizeof(addr)) == -1)
            err_handler("connect() error");
   }
```

<basic serv.c 파일>

```
#include "init_sock.h"
int main(int argc, char** argv){
    int serv_sock;
        int clnt_sock;
        si serv_addr;
        si clnt_addr;
        if(argc != 2){
                 printf("use: %s <port>₩n",
argv[0]);
                 exit(1);
    init_sock_addr(SERV, argv, serv_addr,
&serv_sock);
        cInt_sock = accept(serv_sock, (struct
sockaddr *)&cInt_addr, &cInt_addr_size);
        if(cInt\_sock == -1)
                 err_handler("accept() error");
    char buf[] = "ee4e36e48e510";
        write(cInt_sock, buf, strlen(buf));
        close(clnt_sock);
        close(serv sock);
```

<basic clnt.c 파일>

```
#include "init_sock.h"
int main(int argc, char** argv){
    int clnt sock;
    int str_len;
    si serv_addr;
    char msg[32];
    if(argc != 3){
        printf("use: %s <IP> <port>\mun argv[0]);
        exit(1);
    init_sock_addr(CLNT, argv, serv_addr,
&cInt_sock);
    str_len = read(cInt_sock, msg, sizeof(msg) -
1);
    if(str_len == -1)
        err_handler("read() error!");
    printf("msg from serv: %s\n", msg);
    close(clnt_sock);
    return 0;
```

리눅스 커널 내부구조 책 16 - 22page 과제

비유1

OS(운영체제)

운영체제는 컴퓨터에 있는 CPU, 메모리, 디스크 등의 자원들을 관리하는데

필요시 fork()와 exec()을 이용하여 프로세스를 생성하고 메모리에 할당/해제하여

관리해준다.

비유2

Context Switching

한정된 CPU 자원을 다수의 프로세스들에게 나누기 위해

우선순위 Queue에 프로세스들을 대기시켜놓고 정해진 짧은 시간 동안 번갈아가며

CPU 자원을 사용하도록 하며 프로세스마다 task_struct 구조체를 만들어

진행 사항들을 모두 기록하고 불러오면서 Multi-Tasking을 가능하게 한다.

비유3

Signal

시그널을 받은 프로세스는 기본적으로 종료되며 시그 널 핸들러를 이용하여

특정 시그널을 받았을 경우 특정 함수가 실행되도록 할 수 있다.

비유4

TCP 3-way Handshake

신뢰성 있는 데이터 전송을 위해 TCP/IP 프로토콜을 이용해서

3번의 요청/응답 후에 연결하여 통신을 하는 방식이다.

비유5

super block

파일 시스템의 전체적인 정보를 정리하여 저장한다.

메모리 공간 부족시 디스크와 메모리 사이 Swap을 돕기도 한다.