System Programming

시스템 프로그래밍

(화5, 목6)

Assignment #3-2

Advanced Echo Server

김 태 석 교수님

컴퓨터정보공학부

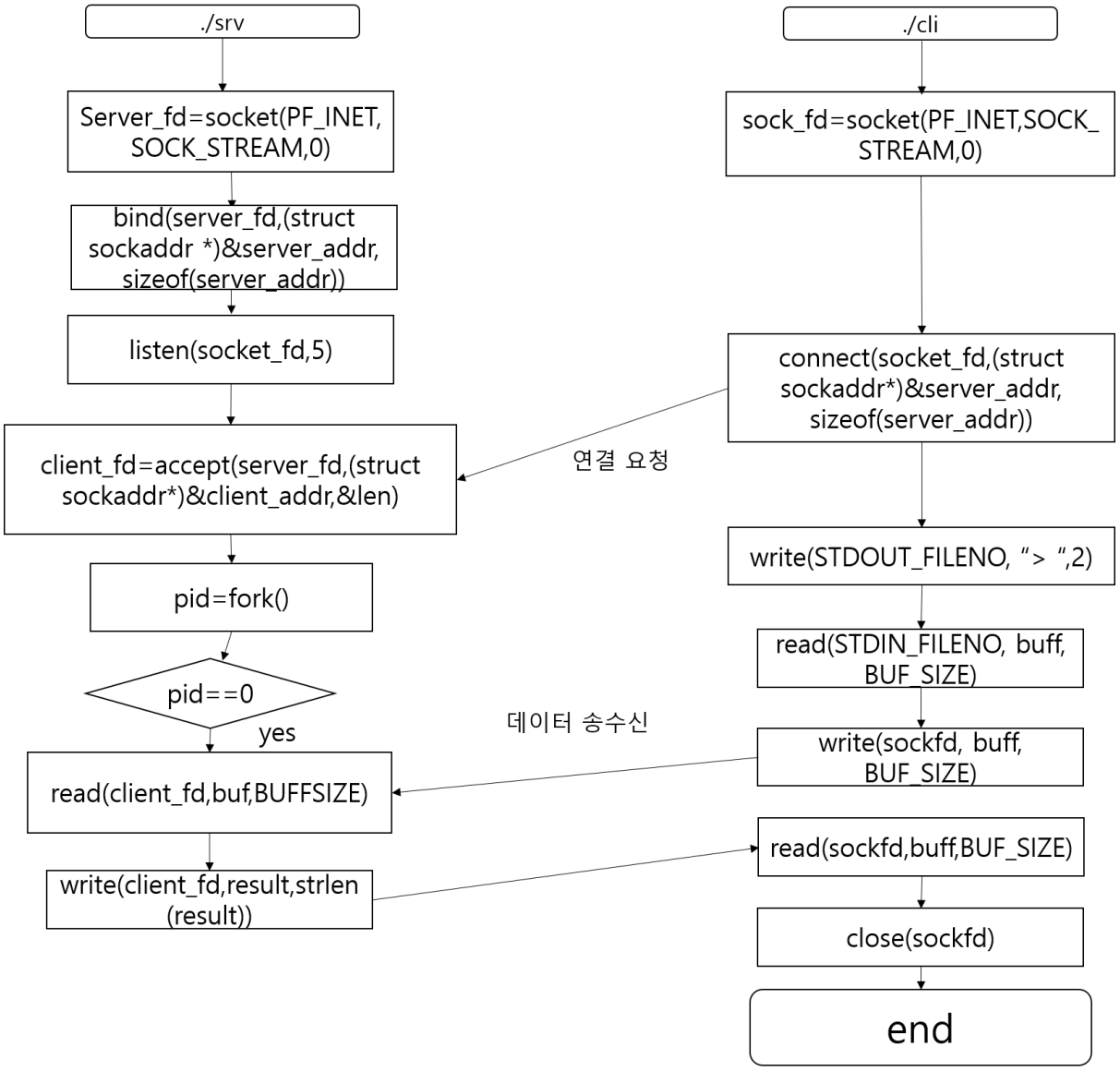
2017202037

오 민 혁

<Introduction>

이번 과제는 Server와 Client를 만들고, Socket programming을 이용하여 Client를 Server에 연결한다. 그 후 fork()를 이용하여 자식 프로세스를 생성하고 이렇게 생성된 자식 프로세스에서 client에서 입력되는 message들을 받고 받은 메세지들을 다시 client로 보내서 client에서 입력 받은 후 출력시키는 프로그램을 구현하는 것이다. 그리고 client에서 QUIT 명령어가 들어오게 되면 Server에서 자식 프로세스를 종료 시킨 후 client도 종료된다.

<Flow Chart>



<Source Code>

<srv.c>

```
////////////////////////////////////////////////////////////////////////
// File Name : cli.c //
// Date : 2020/05/27 //
// Os : Ubuntu 16.04.5 LTS 64bits //
// Author : Oh Min Hyeok //
// Student ID : 2017202037 //
// ------------------------------------------------------------ //
// Title : System Programming Assignment #3-2 ( socket programming ) //
// Description : ... //
////////////////////////////////////////////////////////////////////////
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <sys/wait.h>
#include <signal.h>
#include <string.h>
```

```
#define BUF_SIZE 256

void client_info(struct sockaddr_in cliaddr){ // print client info
        printf("==========Client Info==========₩n");
        printf("client IP: %s₩n₩n",inet_ntoa(cliaddr.sin_addr));
        printf("client port: %d₩n",cliaddr.sin_port);
        printf("================================₩n");
}
void sh_chld(int signum){
        printf("Status of Child process was changed.₩n");
        wait(NULL);
}
void sh_alrm(int signum){
        printf("Child Process(PID : %d) will be terminated.₩n", getpid());
        exit(1);
}

int main(int argc, char **argv)
{
        char buff[BUF_SIZE];
        int n;
        struct sockaddr_in server_addr, client_addr;
```

```
int server_fd, client_fd;
int len;
int port;

signal(SIGCHLD, (void*)sh_chld); // Applying signal handler SIGCHLD
signal(SIGALRM, (void*)sh_alrm); // Applying signal handler SIGALRM

server_fd = socket(PF_INET, SOCK_STREAM, 0); // open socket

memset(&server_addr, 0, sizeof(server_addr)); // initialize
server_addr.sin_family=AF_INET;
server_addr.sin_addr.s_addr=htonl(INADDR_ANY);
server_addr.sin_port=htons(atoi(argv[1]));
// binding server
if(bind(server_fd, (struct sockaddr *)&server_addr, sizeof(server_addr))<0){
        printf("Server: Can't bind local address.₩n"); // exception handling
        return 0;
}
listen(server_fd, 5); // listening server_fd signal

while(1)
{
        pid_t pid;
```

```
		len = sizeof(client_addr);
		client_fd = accept(server_fd, (struct sockaddr*)&client_addr, &len);
		if(client_fd<0){ // exception handling
			printf("Server: accept failed.₩n");
			return 0;
		}
		client_info(client_addr); // printf client info

		pid=fork(); // making child process
		if(pid==0){
			printf("Child Process ID : %d₩n",getpid()); // printf child process information
			while(read(client_fd,buff,BUF_SIZE)>0){ // read buff
				if(strcmp(buff,"QUIT₩n")==0)
					alarm(1); // call sh_arlm()
				else{
					write(client_fd,buff,BUF_SIZE); // write at client_fd
					memset(buff,NULL,BUF_SIZE); // initialize
				}
			}
		}
		close(client_fd); // close
```

```
        }


        return 0;
}
```

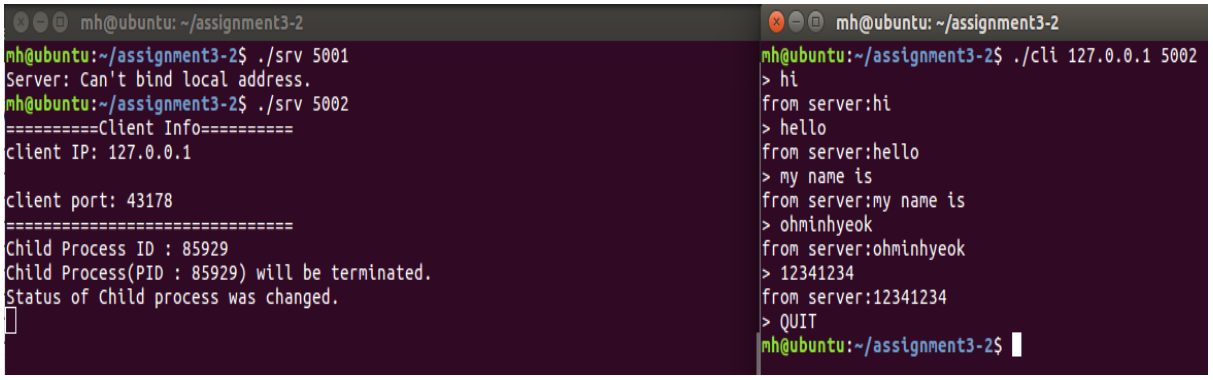
<cli.c>

```
////////////////////////////////////////////////////////////////////
// File Name : cli.c //
// Date : 2020/05/27 //
// Os : Ubuntu 16.04.5 LTS 64bits //
// Author : Oh Min Hyeok //
// Student ID : 2017202037 //
// ------------------------------------------------------------ //
// Title : System Programming Assignment #3-2 ( socket programming ) //
// Description : ... //
////////////////////////////////////////////////////////////////////
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
```

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

#define BUF_SIZE 256

int main(int argc, char **argv)
{
	char buff[BUF_SIZE];
	int n;
	int sockfd;
	struct sockaddr_in serv_addr;

	sockfd = socket(AF_INET, SOCK_STREAM, 0); // socket open

	memset(&serv_addr, 0, sizeof(serv_addr)); // initialize
	serv_addr.sin_family=AF_INET;
	serv_addr.sin_addr.s_addr=inet_addr(argv[1]);
	serv_addr.sin_port=htons(atoi(argv[2]));
	connect(sockfd,(struct sockaddr*)&serv_addr, sizeof(serv_addr)); // connecting with server
	while(1){
		write(STDOUT_FILENO, "> ", 2);
		read(STDIN_FILENO, buff, BUF_SIZE);
```

```
            if(write(sockfd, buff, BUF_SIZE) > 0){
                if(read(sockfd, buff, BUF_SIZE)>0){
                    printf("from server:%s", buff);
                    memset(buff,NULL,BUF_SIZE);
                }
                else
                    break;

            }else
                break;
        }
        close(sockfd);
        return 0;
}
```

<Result Screen>


mh@ubuntu: ~/assignment3-2
mh@ubuntu:~/assignment3-2$ ./srv 5001
Server: Can't bind local address.
mh@ubuntu:~/assignment3-2$ ./srv 5002
==========Client Info==========
client IP: 127.0.0.1
client port: 43178
==============================
Child Process ID : 85929
Child Process(PID : 85929) will be terminated.
Status of Child process was changed.
mh@ubuntu: ~/assignment3-2
mh@ubuntu:~/assignment3-2$ ./cli 127.0.0.1 5002
> hi
from server:hi
> hello
from server:hello
> my name is
from server:my name is
> ohminhyeok
from server:ohminhyeok
> 12341234
from server:12341234
> QUIT
mh@ubuntu:~/assignment3-2$