**유닉스 프로그래밍 기말 과제**

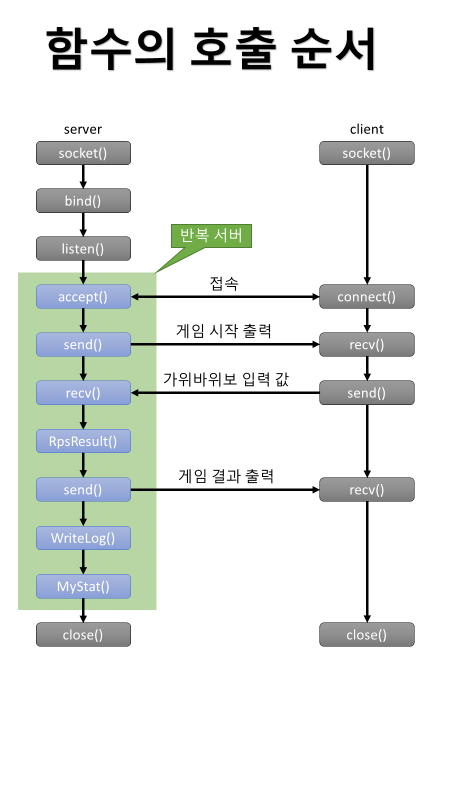
1. 프로그램 개요

* 개발 대상 : 가위바위보 소켓 프로그램
* 개발 목적 : 소켓 프로그래밍을 활용하여 게임을 만들고자 함
* 구현 환경 : UNIX, Putty (apple.smu.ac.kr / 8722)
* 개발 언어 : C
* 구현 디렉토리 : ~/project2

2. 프로그램 설명

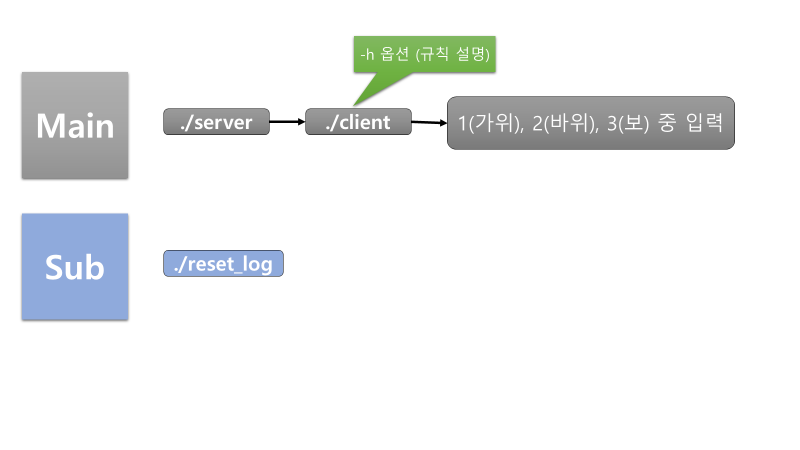
* 소켓 프로그래밍을 이용한 가위바위보 게임.
* 가위바위보 기능 뿐만 아니라 게임 및 전적 기록을 파일로 기록.
* -h 옵션을 통해 규칙 설명을 들을 수 있음
* rand 함수를 이용한 공정한 컴퓨터 가위바위보 값
* 클라이언트 프로그램에선 단순히 recv(), send() 함수만 이용해줌으로써, 코드 은닉 가능

3. 프로그램 구조에 대한 설명서



4. 프로그램 사용 설명서

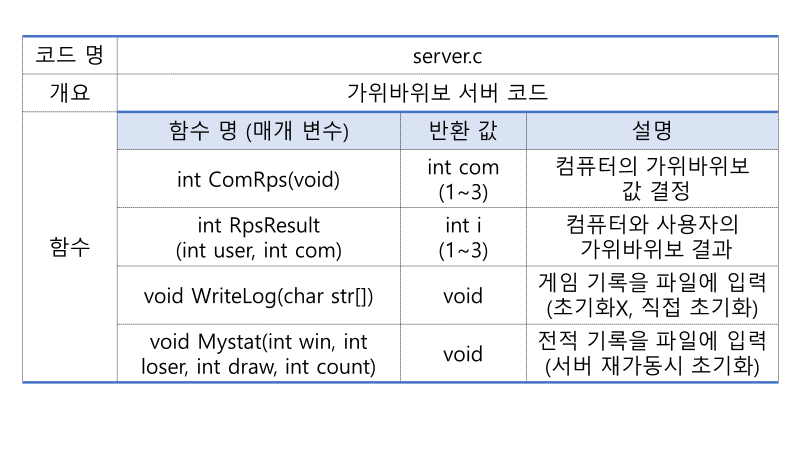
아래의 그림처럼 server 프로그램 실행 후 client 프로그램(-h 옵션 : 규칙 설명) 을 실행



5. 폴더 내 파일 설명



6. server.c 코드 함수 설명



7. 소스코드와 주석문

**server.c**

/\*\*

\* ----------------------------------------

\* @title : server.c

\* @author : 임현 (201511054@sangmyung.kr)

\* @since : 2017 - 06 - 08

\* @brief : 인터넷 소켓(서버) 가위바위보 게임

\* ----------------------------------------

\*/

#include <sys/types.h>

#include <sys/socket.h>

#include <sys/stat.h> // FileIO

#include <unistd.h>

#include <arpa/inet.h>

#include <stdlib.h>

#include <string.h>

#include <stdio.h>

#include <time.h> // rand

#include <fcntl.h> // FileIO

// 201511054 컴퓨터과학과 임현 개인포트 번호

#define PORTNUM 50013

// 가위, 바위, 보

static const char \*RPS[] = { "가위", "바위", "보" };

// 승리, 패배, 무승부

static const char \*RESULT[] = { "승리", "패배", "무승부" };

/\*\*

\* @title : int ComRps(void)

\* @brief : 컴퓨터의 가위바위보 값 결정

\*/

int ComRps(void);

/\*\*

\* @title : int RpsResult(int user, int com);

\* @brief : 컴퓨터와 사용자의 가위바위보 결과

\*/

int RpsResult(int user, int com);

/\*\*

\* @title : void WriteLog(char str[]);

\* @brief : 게임 기록을 파일에 입력하는 함수 (초기화 X, 직접 초기화)

\*/

void WriteLog(char str[]);

/\*\*

\* @title : void MyStat(int win, int lose, int draw, int count);

\* @brief : 전적 기록을 파일에 입력하는 함수 (서버 재가동시 초기화)

\*/

void MyStat(int win, int lose, int draw, int count);

int main(void) {

char buf[256];

struct sockaddr\_in sin, cli;

int sd, ns, clientlen = sizeof(cli);

int com, user; // 컴퓨터 및 사용자의 가위바위보 값, [가위(1), 바위(2), 보(3)]

int win = 0; // 승리 횟수

int lose = 0; // 패배 횟수

int draw = 0; // 무승부 횟수

int count = 0; // 게임 횟수

int i; // 승리(1), 패배(2), 무승부(3)

// 소켓 파일 기술자 생성

if ((sd = socket(AF\_INET, SOCK\_STREAM, 0)) == -1) {

perror("socket");

exit(1);

}

// 소켓 경로명 지정

memset((char \*)&sin, '\0', sizeof(sin));

sin.sin\_family = AF\_INET;

sin.sin\_port = htons(PORTNUM); // 50013

sin.sin\_addr.s\_addr = inet\_addr("117.16.42.29"); // apple.smu.ac.kr

// 소켓 파일 기술자를 지정된 IP 주소/포트 번호와 결합(bind)

if (bind(sd, (struct sockaddr \*)&sin, sizeof(sin))) {

perror("bind");

exit(1);

}

// 클라이언트의 접속 요청 대기

if (listen(sd, 5)) {

perror("listen");

exit(1);

}

// 반복 서버

while (201511054) {

// 클라이언트의 접속 허용

if ((ns = accept(sd, (struct sockaddr \*)&cli, &clientlen)) == -1) {

perror("accept");

exit(1);

}

// 가위바위보 게임 시작 출력

sprintf(buf, "\n===============가위바위보 게임===============\n\n1(가위), 2(>바위), 3(보) 중 하나를 입력하십시오.\n");

// 데이터 송신 (시작 화면 출력)

if (send(ns, buf, strlen(buf) + 1, 0) == -1) {

perror("send");

exit(1);

}

// 데이터 수신 (buf 값(가위, 바위, 보)을 가져옴)

if (recv(ns, buf, sizeof(buf), 0) == -1) {

perror("recv");

exit(1);

}

user = atoi(buf); // 사용자 가위바위보 값 결정 (아스키 -> 인티져)

com = ComRps(); // 컴퓨터 가위바위보 값 결정 (rand함수)

// 가위바위보 결과

i = RpsResult(user, com);

// 가위바위보 결과 출력

sprintf(buf, "\n사용자 : %s vs 컴퓨터 : %s \n%s 하셨습니다!\n", RPS[user - 1], RPS[com - 1], RESULT[i - 1]);

// 데이터 송신 (가위바위보 결과 값을 보냄)

if (send(ns, buf, strlen(buf) + 1, 0) == -1) {

perror("send");

exit(1);

}

// 로그 기록 함수

WriteLog(buf);

// 전, 승, 무, 패 증가

switch (i) {

case 1: win++; break;

case 2: lose++; break;

case 3: draw++; break;

}

count++;

// 전적 기록 함수

MyStat(win, lose, draw, count);

}

// 소켓 파일 기술자 종료

close(ns);

close(sd);

return 0;

}

// 컴퓨터의 가위바위보 값 결정

int ComRps(void) {

int com; // 컴퓨터의 가위바위보 값

srand((unsigned)time(NULL));

com = rand() % 3 + 1; // 1 ~ 3 까지의 랜덤 변수

return com;

}

// 컴퓨터와 사용자의 가위바위보 결과

int RpsResult(int user, int com) {

int i;

if (user == com) // 무승부 (같을 경우)

i = 3;

else { // 무승부가 아닐 경우

switch (user) { // 사용자의 값에 따라

case 1: i = (com == 3 ? 1 : 2); break;

case 2: i = (com == 1 ? 1 : 2); break;

case 3: i = (com == 2 ? 1 : 2); break;

}

}

// i = 1(승리), i = 2(패배), i = 3(무승부)

return i;

}

// 게임 기록을 파일에 입력하는 함수

void WriteLog(char str[]) {

int fd;

int n;

// log.txt 파일에 입력

fd = open("log.txt", O\_WRONLY | O\_APPEND); // 쓰기 전용, 이어 쓰기

if (fd == -1) {

perror("Open");

exit(1);

}

// buf값(가위바위보 결과 값)을 파일에 저장

n = write(fd, str, strlen(str));

if (n != strlen(str)) {

perror("Write");

exit(2);

}

close(fd);

}

// 전적 기록을 파일에 입력하는 함수

void MyStat(int win, int lose, int draw, int count) {

int fd;

int n;

char str[256];

// 전, 승, 무, 패, 승률 출력

sprintf(str, "%d 전 \t %d 승 \t %d 무 \t %d 패 \n승률 : %1.f%%\n", count, win, draw, lose, (double)win / count \* 100);

// mystat.txt 파일에 입력

fd = open("mystat.txt", O\_WRONLY); // 쓰기 전용

if (fd == -1) {

perror("Open");

exit(1);

}

// 전적을 파일에 저장

n = write(fd, str, strlen(str));

if (n != strlen(str)) {

perror("Write");

exit(2);

}

close(fd);

}

**client.c**

/\*\*

\* ----------------------------------------

\* @title : client.c

\* @author : 임현 (201511054@sangmyung.kr)

\* @since : 2017 - 06 - 08

\* @brief : 인터넷 소켓(클라이언트) 가위바위보 게임

\* ----------------------------------------

\*/

#include <sys/types.h>

#include <sys/socket.h>

#include <unistd.h>

#include <arpa/inet.h>

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

// 201511054 컴퓨터과학과 임현 개인포트 번호

#define PORTNUM 50013

/\*\*

\* @title : void Rule(int argc, char \*argv[]);

\* @brief : 가위바위보 규칙을 설명해주는 함수 (-h 옵션)

\*/

void Rule(int argc, char \*argv[]);

int main(int argc, char \*argv[]) {

int sd;

char buf[256];

struct sockaddr\_in sin;

// 가위바위보 규칙 (옵션 -h 사용 시 출력)

Rule(argc, argv);

// 소켓 파일 기술자 생성

if ((sd = socket(AF\_INET, SOCK\_STREAM, 0)) == -1) {

perror("socket");

exit(1);

}

// 소켓 경로명 지정

memset((char \*)&sin, '\0', sizeof(sin));

sin.sin\_family = AF\_INET;

sin.sin\_port = htons(PORTNUM); // 50013

sin.sin\_addr.s\_addr = inet\_addr("117.16.42.29"); // apple.smu.ac.kr

// 클라이언트가 서버에 접속 요청

if (connect(sd, (struct sockaddr \*)&sin, sizeof(sin))) {

perror("connect");

exit(1);

}

// 데이터 수신 (가위바위보 게임 시작 출력을 받아옴)

if (recv(sd, buf, sizeof(buf), 0) == -1) {

perror("recv");

exit(1);

}

printf("%s\n", buf);

// buf에 입력

printf("입력 : ");

scanf("%s", &buf);

// 데이터 송신 (buf 값을 보냄)

if (send(sd, buf, strlen(buf) + 1, 0) == -1) {

perror("send");

exit(1);

}

// 데이터 수신 (가위바위보 결과 값을 받아옴)

if (recv(sd, buf, sizeof(buf), 0) == -1) {

perror("recv");

exit(1);

}

printf("%s\n", buf);

// 소켓 파일 기술자 종료

close(sd);

return 0;

}

// 가위바위보 규칙을 설명해주는 함수 (-h 옵션)

void Rule(int argc, char \*argv[]) {

int n;

// getopt 함수로 인자가 있는지 확인

while ((n = getopt(argc, argv, "h")) != -1) {

switch (n) { // 옵션별 기능 수행

case 'h': // -h 옵션

printf("\n");

printf(" ========가위바위보 규칙=========\n");

printf(" | 가위는 보자기를 자르고, |\n");

printf(" | 보자기는 바위를 감싸고, |\n");

printf(" | 바위는 가위로 자를 수 없다. |\n");

printf(" | 즉, 가위>보자기>바위>가위... |\n");

printf(" --------------------------------\n");

break;

}

}

}

**reset\_log.c**

/\*\*

\* ----------------------------------------

\* @title : reset\_log.c

\* @author : 임현 (201511054@sangmyung.kr)

\* @since : 2017-06-09

\* @brief : log.txt(게임 기록) 초기화 프로그램

\* ----------------------------------------

\*/

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

#include <stdlib.h>

#include <stdio.h>

int main(void) {

int fd;

// 파일 열기 (oflag = 모두 지우거나, 파일을 만듬), (접근권한 : 0644)

fd = open("log.txt", O\_TRUNC | O\_CREAT, 0644);

if (fd == -1) {

perror("Open");

exit(1);

}

close(fd);

}