

# Final project #1 번

2014146019 석상우

## 1. Server.c 코드분석

▶ 예제와 동일한 부분을 빼고, 무한루프에서 fork()하는 부분부터의 코드

```
72     while(1)
73     {
74         new_fd = accept(sockfd, (struct sockaddr *)&their_addr, &sin_size);
75         send(new_fd, INIT_MSG, strlen(INIT_MSG) + 1, 0);
76         if(new_fd < 0)
77         {
78             printf("\nServer: accept failed!\n");
79             exit(0);
80         }
81         else
82         {
83             printf("accept() is OK...\n");
84         }
85         pid = fork();
86
87         if(pid == 0)
88         {
89             recv_byte = recv(new_fd, id, sizeof(id), 0);
90             recv_byte = recv(new_fd, pw, sizeof(pw), 0);
91             char success1[100] = "Log-in success!! [User1] xxxx\n";
92             char success2[100] = "Log-in success!! [User2] xxxx\n";
93             char fail[100] = "Log in fail : Incorrect password...\n";
94
95             printf("\n-----\nUser Information\n");
96             printf("ID: %s, PW: %s\n", id, pw);
97             printf("-----");
98             if((strcmp(id, USER1_ID) == 0) && (strcmp(pw, USER1_PW) == 0))
99             {
100                 printf("%s", success1);
101                 send(new_fd, success1, strlen(success1) + 1, 0);
102             }
103             else if((strcmp(id, USER2_ID) == 0) && (strcmp(pw, USER2_PW) == 0))
104             {
105                 printf("%s", success2);
106                 send(new_fd, success2, strlen(success2) + 1, 0);
107             }
108             else
109             {
110                 printf("%s", fail);
111                 send(new_fd, fail, strlen(fail) + 1, 0);
112             }
113             close(new_fd);
114             exit(0);
115         }
116         else if(pid > 0)
117         {
118             close(new_fd);
119         }
120     }
121     close(sockfd);
122     return 0;
123 }
```

74 ~ 75 줄 : client 와 통신할 socket 을 생성하고 초기화면을 client 로 send 한다

85 줄 : 부모 process 를 fork 를 한 정수값을 pid 로 받는다.

87 줄~ : 자식프로세스가 하는 일

89 ~ 90 줄 : client 에서 입력한 id 와 password 를 receive 한다.

96 ~ 98 줄 : 받아온 id 와 password 를 보여주기 위한 인터페이스를 보여준다.

98 ~ 112 줄 : id 와 password 가 미리 지정해 둔 매크로 문자열과 같다면 로그인 이 됐다는 정답화면을 출력하고 다르다면 로그인 실패라는 실패 화면을 출력한다. + [정답 / 실패] 에 해당하는 문자열을 client 로 send 한 후 client 측에서 받은 문자열을 출력시킨다.

[Strcmp(a,b)가 0 이라면 a 와 b 문자열은 같다.]

113 줄 : 소켓을 close 하고 나간다.

114 줄 : pid > 0 [ 부모프로세스는 아무일도 하지 않고 socket 을 close 하고 다시 while 루프로 가서 새로운 client 를 기다린다.]

## 2. Client.c 코드분석(client1 과 2 는 동일코드)

```
43     rcv_byte = recv(sockfd, buf, sizeof(buf), 0);
44     printf("%s\n", buf);
45
46     printf("ID: ");
47     scanf("%s", id);
48     send(sockfd, id, strlen(id)+1, 0);
49
50     printf("PW: ");
51     scanf("%s", pw);
52     send(sockfd, pw, strlen(pw)+1, 0);
53
54     rcv_byte = recv(sockfd, correct, sizeof(correct), 0);
55     printf("%s", correct);
56
57     return 0;
58 }
```

43 ~ 44 줄 : server 에서 보낸 초기화면에 대한 문자열을 받고 출력하는 코드

46 ~ 52 줄 : client 에서 id 와 password 를 입력하고 server 로 id , pw 문자열을 send 한다.

54 ~ 55 줄 : server 측에서 정답/오답에 대한 문자열을 받고 출력한다.

## 3. 실행 화면

### 1) client 초기화면 + [id / password 입력] (정답)

```
Client  socket() sockfd is Ok...
Client  connect() is Ok...

-----

Hello! I'm p2p sharing Server...
Please, LOG IN !

-----

ID: user1
PW: passwd1
Log in success!! [User1] >>>
```

## 2) 정답일 경우 [Server]

```
listen() is OK...
accept() is OK...

=====
User Information
ID: user1, PW: passwd
=====Log-in success!! [User1] ****
```

## 3) client 초기화면 + [id / password 입력] (오답)

```
=====
Hello! I'm p2p chatting Server...
Please, LOG IN !
=====

ID: asfd
PW: asfd
Log-in fail : Incorrect password...
```

## 4) 오답일 경우의 [server]

```
=====
accept() is OK...

=====
User Information
ID: user2, PW: sangwoo
=====Log-in fail: Incorrect password...
```

결과 : server 와 client 두개를 이용해 client 는 접속을 시도하고, server 가 accept 한다면 client 쪽에서 id 와 password 를 입력하고 입력한 정보를 server 쪽으로 넘겨서 정해져 있는 데이터와 일치하는지 판별하고 그에대한 결과를 출력하는 프로그램.