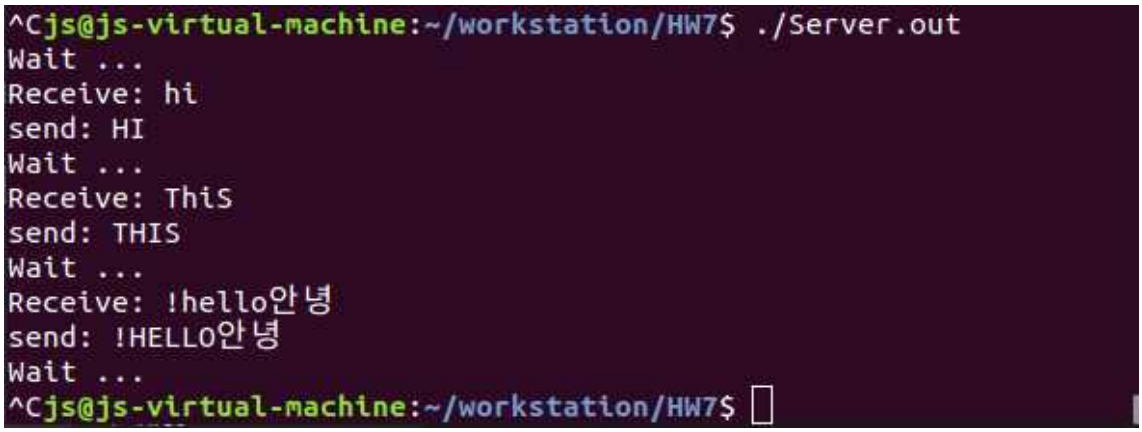
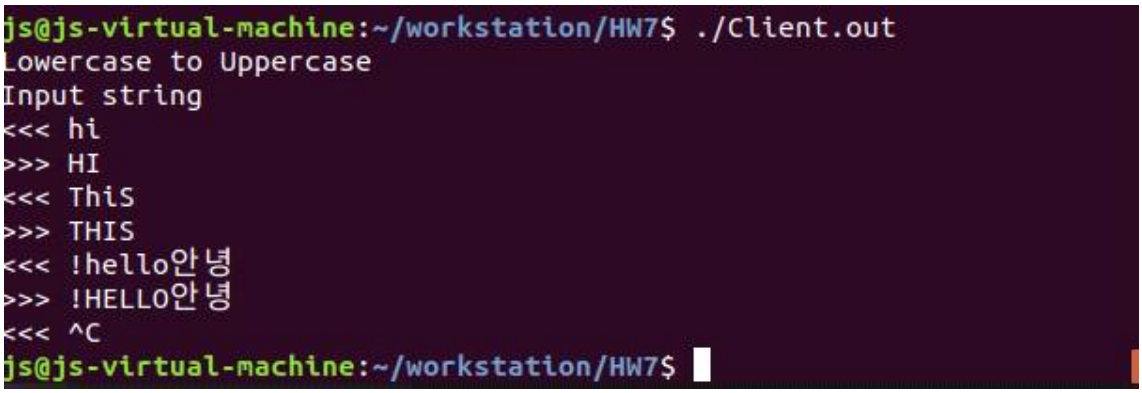
HW7

2019203102 유지성

실행 결과



Server.out



Client.out

MyMsg.h

```
4   #define MSG_TYPE_IN 1
5   #define MSG_TYPE_OUT 2
6   #define MSG_SIZE_IN (sizeof(MsgIO) - sizeof(long))
7   #define MSG_SIZE_OUT (sizeof(MsgIO) - sizeof(long))
8
9   #define MAX_BUFF 512
10
11  struct __MsgIO {
12      long mtype;
13      char str[MAX_BUFF];
14  };
15  typedef struct   MsgIO MsgIO;
```

메시지 유형을 저장할 mtype, 문자열을 저장할 str을 구조체에 선언

Server.c

```
18      MsgIO msgIn;
19      MsgIO msgOut;
20
21      mykey = ftok("mymsgkey", 1);            // 키생성
22      msqid = msgget(mykey, IPC_CREAT | 0600); // 메시지큐생성
```

msgsnd와 msgrcv에 사용될 msgIn과 msgOut 선언
키와 메시지큐 생성

```
26      while (1) {
27          puts("Wait ...");
28
29          memset(&msgIn, 0x00, sizeof(MsgIO));
30          msgrcv(msqid, &msgIn, MSG_SIZE_IN, MSG_TYPE_IN, 0);
31          printf("Receive: %s\n", msgIn.str);
32
33          memset(&msgOut, 0x00, sizeof(MsgIO));
34          msgOut.mtype = MSG_TYPE_OUT;
35          strcpy(msgOut.str, toUpper(msgIn.str));
36          msgsnd(msqid, &msgOut, MSG_SIZE_OUT, 0);
37          printf("send: %s\n", msgOut.str);
38
39          fflush(stdout);
40      }
```

Client로부터 msgrcv로 메시지를 받아와 msgIn에 저장

msgOut에 메시지 타입과 uppercase로 바꾼 문자열을 저장 후 msgsnd로 Client에 전송

```
44  void signalHandler(int signum) {
45      if (signum == SIGINT) {
46          msgctl(msqid, IPC_RMID, NULL);
47          exit(0);
48      }
49  }
```

SIGINT 시그널이 들어올 경우 msgctl의 IPC_RMID로 메시지 큐 제거

```
51  char *toUpper(char str[]) {
52      for (int i = 0; str[i] != '\0'; i++) {
53          if (str[i] >= 'a' && str[i] <= 'z')
54              str[i] = str[i] - 32;
55      }
56      return str;
57  }
```

소문자를 대문자로 바꿔줌

Client.c

```
12      key_t mykey = ftok("mymsgkey", 1);
13      int msqid = msgget(mykey, IPC_CREAT);
14
15      char string[100];
16
17      MsgIO msgIn;
18      MsgIO msgOut;
```

msgsnd와 msgrcv에 사용될 msgIn과 msgOut 선언
키와 메시지큐 생성

```
20      puts("Lowercase to Uppercase");
21      puts("Input string");
22      while (1) {
23          printf("<<< ");
24
25          scanf("%s", string);
26          fflush(stdout);
27          fflush(stdin);
28
29          memset(&msgIn, 0x00, sizeof(MsgIO));
30          msgIn.mtype = MSG_TYPE_IN;
31          strcpy(msgIn.str, string);
32          msgsnd(msqid, &msgIn, MSG_SIZE_IN, 0);
33
34          memset(&msgOut, 0x00, sizeof(MsgIO));
35          memset(&msgOut.str, 0x00, sizeof(msgOut.str));
36          msgrcv(msqid, &msgOut, MSG_SIZE_OUT, MSG_TYPE_OUT, 0);
37          printf(">>> %s\n", msgOut.str);
38      }
```

msgIn에 메시지 타입과 Server에 보낼 string을 저장
msgsnd로 msgIn을 Server로 전송
msgrcv로 Server로부터 msgOut을 받아옴

고찰

Client의 msgrcv에서 수신하고자 하는 메시지의 크기를 잘못 설정해서 찾아내는데 힘든 시간을 보냈다. MSG_SIZE_OUT로 설정해야 하는데 MSG_TYPE_OUT으로 설정해 버려서 문자열이 앞의 두 문자만 출력되는 버그가 있었다.
프로세스에서 다른 프로세스로 명령을 보내는 방법 중 하나를 알게 되었다.