

정보통신실험3

# 5장. 프로그램 수정

# 오류 상황 및 해결 방안

- ▶ VoteClientTCP.c에서 수신된 응답에 먼저 송신한 요청 메시지가 남아있어 득표수 확인이 안되는 이상 현상 발생
  - ▶ 방법 1
    - ▶ getNextMsg()를 한번 더 호출하여 남아있는 메시지를 제거한 후 응답 확인 가능
    - ▶ 근본적인 해결책이 될 수 없음
  - ▶ Stream I/O로 Wrap 하는 과정에서 시스템의 오동작이 발생하는 것으로 추정됨
  - ▶ 방법 2
    - ▶ VoteClientTCP.c와 VoteServerTCP.c에서 Stream I/O로 Wrap 하는 과정을 제거하고, DelimFramer.c와 LengthFramer.c에서 파일이 아닌 소켓에 바로 읽고 쓰는 방식으로 수정
- ▶ Makefile을 이용하여 개발을 용이하게 할 필요가 있음

# Makefile

```
Common_files = addressUtility.o dieWithMessage.o
LIBS = -lsocket -lnsl
CFLAGS = -xc99
```

```
vs_tcp_td : vsTcp.o tcpServerUtility.o voteEncodingText.o delimiterFramer.o $(Common_files)
cc -o vs_tcp_td $(CFLAGS) $(LIBS) vsTcp.o tcpServerUtility.o voteEncodingText.o delimiterFramer.o $(Common_files)
```

```
vc_tcp_td : vcTcp.o tcpClientUtility.o voteEncodingText.o delimiterFramer.o $(Common_files)
cc -o vc_tcp_td $(CFLAGS) $(LIBS) vcTcp.o tcpClientUtility.o voteEncodingText.o delimiterFramer.o $(Common_files)
```

```
vs_tcp_tl : vsTcp.o tcpServerUtility.o voteEncodingText.o lengthFramer.o $(Common_files)
cc -o vs_tcp_tl $(CFLAGS) $(LIBS) vsTcp.o tcpServerUtility.o voteEncodingText.o lengthFramer.o $(Common_files)
```

```
vc_tcp_tl : vcTcp.o tcpClientUtility.o voteEncodingText.o lengthFramer.o $(Common_files)
cc -o vc_tcp_tl $(CFLAGS) $(LIBS) vcTcp.o tcpClientUtility.o voteEncodingText.o lengthFramer.o $(Common_files)
```

```
vs_tcp_bd : vsTcp.o tcpServerUtility.o voteEncodingBin.o delimiterFramer.o $(Common_files)
cc -o vs_tcp_bd $(CFLAGS) $(LIBS) vsTcp.o tcpServerUtility.o voteEncodingBin.o delimiterFramer.o $(Common_files)
```

```
vc_tcp_bd : vcTcp.o tcpClientUtility.o voteEncodingBin.o delimiterFramer.o $(Common_files)
cc -o vc_tcp_bd $(CFLAGS) $(LIBS) vcTcp.o tcpClientUtility.o voteEncodingBin.o delimiterFramer.o $(Common_files)
```

```
vs_tcp_bl : vsTcp.o tcpServerUtility.o voteEncodingBin.o lengthFramer.o $(Common_files)
cc -o vs_tcp_bl $(CFLAGS) $(LIBS) vsTcp.o tcpServerUtility.o voteEncodingBin.o lengthFramer.o $(Common_files)
```

```
vc_tcp_bl : vcTcp.o tcpClientUtility.o voteEncodingBin.o lengthFramer.o $(Common_files)
cc -o vc_tcp_bl $(CFLAGS) $(LIBS) vcTcp.o tcpClientUtility.o voteEncodingBin.o lengthFramer.o $(Common_files)
```

```
clean :
rm *.o
rm vs_tcp_* vc_tcp_*
```

```
all : vs_tcp_td vc_tcp_td vs_tcp_tl vc_tcp_tl vs_tcp_bd vc_tcp_bd vs_tcp_bl vc_tcp_bl
```

vs\_tcp\_td

server text delimiter

vc\_tcp\_bl

client binary length

# 방법 1 (임시방안) vcTcp.c (1)

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4 #include <stdint.h>
5 #include <unistd.h>
6 #include <errno.h>
7 #include <sys/socket.h>
8 #include <netinet/in.h>
9 #include <arpa/inet.h>
10 #include <netdb.h>
11 #include "Practical.h"
12 #include "VoteProtocol.h"
13 #include "Framer.h"
14 #include "VoteEncoding.h"
15
16 int main(int argc, char *argv[])
17 {
18     if (argc < 4 || argc > 5) // Test for correct # of args
19         DieWithUserMessage("Parameter(s)", "<Server Address/Name> <Server Port/Service> <Candidate> [I]");
20
21     char *server = argv[1]; // First arg: server address/name
22     char *service = argv[2]; // Second arg: server port/service
23
24     int candi = atoi(argv[3]);
25     if (candi < 0 || candi > MAX_CANDIDATE)
26         DieWithUserMessage("Candidate # not valid", argv[3]);
27
28     bool inq = (argc > 4) && (strcmp(argv[4], "I") == 0);
29
```

# 방법 1 (임시방안) vcTcp.c (2)

```
30 // Create a connected TCP socket
31 int sock = SetupTCPClientSocket(server, service);
32 if (sock < 0)
33     DieWithUserMessage("SetupTCPClientSocket() failed", "unable to connect");
34
35 FILE *str = fdopen(sock, "r+"); // Wrap for stream I/O
36 if (str == NULL)
37     DieWithSystemMessage("fdopen() failed");
38
39 // Set up info for a request
40 VoteInfo vi;
41 memset(&vi, 0, sizeof(vi));
42
43 vi.isInquiry = inq;
44 vi.candidate = candi;
45
46 // Encode for transmission
47 uint8_t outbuf[MAX_WIRE_SIZE];
48 size_t reqSize = Encode(&vi, outbuf, MAX_WIRE_SIZE);
49
50 // Print info
51 printf("Sending %d-byte %s for candidate %d...\n", reqSize,
52        (inq ? "inquiry" : "vote"), candi);
53
54 // Frame and send
55 if (PutMsg(outbuf, reqSize, str) < 0)
56     DieWithSystemMessage("PutMsg() failed");
57
```

파일 스트림으로  
Wrap 하는 과정

# 방법 1 (임시방안) vcTcp.c (3)

```
58 // Receive and print response
59 uint8_t inbuf[MAX_WIRE_SIZE];
60 memset(&vi, 0, sizeof(vi));
61 size_t respSize = GetNextMsg(str, inbuf, MAX_WIRE_SIZE); // Get the message
62 respSize = GetNextMsa(str, inbuf, MAX_WIRE_SIZE); // by kau
63
64 if (Decode(inbuf, respSize, &vi)) { // Parse it
65     printf("Received: %d\n", respSize);
66     if (vi.isResponse)
67         printf(" Response to ");
68     if (vi.isInquiry)
69         printf("inquiry ");
70     else
71         printf("vote ");
72
73     printf("for candidate %d\n", vi.candidate);
74     if (vi.isResponse)
75         printf(" count = %llu\n", vi.count);
76
77 }
78
79 // Close up
80 fclose(str);
81
82 exit(0);
83 }
```

GetNextMsg를 한번 더  
호출하여 이상 메시지 제거

# 방법 2 - 수정된 vcTcp.c (1)

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4 #include <stdint.h>
5 #include <unistd.h>
6 #include <errno.h>
7 #include <sys/socket.h>
8 #include <netinet/in.h>
9 #include <arpa/inet.h>
10 #include <netdb.h>
11 #include "Practical.h"
12 #include "VoteProtocol.h"
13 #include "Framer.h"
14 #include "VoteEncoding.h"
15
16 int main(int argc, char *argv[])
17 {
18     if (argc < 4 || argc > 5) // Test for correct # of args
19         DieWithUserMessage("Parameter(s)", "<Server Address/Name> <Server Port/Service> <Candidate> [I]");
20
21     char *server = argv[1]; // First arg: server address/name
22     char *service = argv[2]; // Second arg: server port/service
23
24     int candi = atoi(argv[3]);
25     if (candi < 0 || candi > MAX_CANDIDATE)
26         DieWithUserMessage("Candidate # not valid", argv[3]);
27
28     bool inq = (argc > 4) && (strcmp(argv[4], "I") == 0);
29
```

# 방법 2 - 수정된 vcTcp.c (2)

```
30 // Create a connected TCP socket
31 int sock = SetupTCPClientSocket(server, service);
32 if (sock < 0)
33     DieWithUserMessage("SetupTCPClientSocket() failed", "unable to connect");
34
35 // Set up info for a request
36 VoteInfo vi;
37 memset(&vi, 0, sizeof(vi));
38
39 vi.isInquiry = inq;
40 vi.candidate = candi;
41
42 // Encode for transmission
43 uint8_t outbuf[MAX_WIRE_SIZE] = {0}; // by kgu
44 size_t reqSize = Encode(&vi, outbuf, MAX_WIRE_SIZE);
45
46 // Print info
47 printf("Sending %d-byte %s for candidate %d...\n", reqSize,
48        (inq ? "inquiry" : "vote"), candi);
49
50 // Frame and send
51 if (PutMsg(outbuf, reqSize, sock) < 0)
52     DieWithSystemMessage("PutMsg() failed");
53
54 // Receive and print response
55 uint8_t inbuf[MAX_WIRE_SIZE] = {0}; // by kgu
56 memset(&vi, 0, sizeof(vi));
57 size_t respSize = GetNextMsg(sock, inbuf, MAX_WIRE_SIZE); // Get the message
58
```

소켓 스트림화 과정 삭제

버퍼 초기화

PutMsg() 함수 세번째 인자가 socket\_fd로 수정됨

버퍼 초기화

GetNextMsg() 함수 첫 번째 인자가 socket\_fd로 수정됨



## 방법 2 - 수정된 vcTcp.c (3)

```
59     if (Decode(inbuf, respSize, &vi)) { // Parse it
60         printf("Received: %d\n", respSize);
61         if (vi.isResponse)
62             printf(" Response to ");
63         if (vi.isInquiry)
64             printf("inquiry ");
65         else
66             printf("vote ");
67
68         printf("for candidate %d\n", vi.candidate);
69         if (vi.isResponse)
70             printf(" count = %llu\n", vi.count);
71
72     }
73
74     // Close up
75     close(sock);
76     exit(0);
77 }
```

# 방법 2 - 수정된 vsTcp.c (1)

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdbool.h>
4 #include <stdint.h>
5 #include <unistd.h>
6 #include <fcntl.h>
7 #include <errno.h>
8 #include <sys/socket.h>
9 #include <arpa/inet.h>
10 #include "Practical.h"
11 #include "VoteProtocol.h"
12 #include "VoteEncoding.h"
13 #include "Framer.h"
14
15 static uint64_t counts[MAX_CANDIDATE + 1];
16
17 int main(int argc, char *argv[])
18 {
19
20     if (argc != 2) // Test for correct number of arguments
21         DieWithUserMessage("Parameter(s)", "<Server Port/Service>");
22
23     int servSock = SetupTCPServerSocket(argv[1]);
24     // servSock is now ready to use to accept connections
25
26     for (;;) { // Run forever
27
28         // Wait for a client to connect
29         int clntSock = AcceptTCPConnection(servSock);
```

# 방법 2 - 수정된 vsTcp.c (2)

```
30
31 // Receive messages until connection closes
32 int mSize;
33 uint8_t inBuf[MAX_WIRE_SIZE] = {0}; // by kgu
34 VoteInfo v;
35 while ((mSize = GetNextMsg(clntSock, inBuf, MAX_WIRE_SIZE)) > 0) {
36     memset(&v, 0, sizeof(v)); // Clear vote information
37     printf("Received message (%d bytes)\n", mSize);
38
39     if (Decode(inBuf, mSize, &v)) { // Parse to get VoteInfo
40         if (!v.isResponse) { // Ignore non-requests
41             v.isResponse = true;
42             if (v.candidate >= 0 && v.candidate <= MAX_CANDIDATE) {
43                 if (!v.isInquiry)
44                     counts[v.candidate] += 1;
45                 v.count = counts[v.candidate];
46             } // Ignore invalid candidates
47         }
48         uint8_t outBuf[MAX_WIRE_SIZE] = {0}; // by kgu
49         mSize = Encode(&v, outBuf, MAX_WIRE_SIZE);
50         if (PutMsg(outBuf, mSize, clntSock) < 0) {
51             fputs("Error framing/outputting message\n", stderr);
52             break;
53         } else {
54             printf("Processed %s for candidate %d; current count is %llu.\n",
55                 (v.isInquiry ? "inquiry" : "vote"), v.candidate, v.count);
56         }
57     }
58 }
```

버퍼 초기화

GetNextMsg() 함수 첫 번째 인자가 socket\_fd로 수정됨

버퍼 초기화

PutMsg() 함수 세 번째 인자가 socket\_fd로 수정됨

## 방법 2 - 수정된 vsTcp.c (3)

```
58         else {
59             fputs("Parse error, closing connection.\n", stderr);
60             break;
61         }
62     }
63     puts("Client finished");
64     close(clntSock);
65 } // Each client
66 // NOT REACHED
```

# 방법 2 - 수정된 delimiterFramer.c (1)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <stdint.h>
4 #include <unistd.h>
5 #include "Practical.h"
6
7 static const char DELIMITER = '\n';
8
9 /* Read up to bufSize bytes or until delimiter, copying into the given
10 * buffer as we go.
11 * Encountering EOF after some data but before delimiter results in failure.
12 * (That is: EOF is not a valid delimiter.)
13 * Returns the number of bytes placed in buf (delimiter NOT transferred).
14 * If buffer fills without encountering delimiter, negative count is returned.
15 * If stream ends before first byte, -1 is returned.
16 * Precondition: buf has room for at least bufSize bytes.
17 */
18 int GetNextMsg(int sockfd, uint8_t *buf, size_t bufSize)
19 {
20     int count = 0;
21     uint8_t nextChar;
22
23     while (count < bufSize) {
24         read(sockfd, &nextChar, 1);
25         if (nextChar == EOF) {
26             if (count > 0)
27                 DieWithUserMessage("GetNextMsg()", "Stream ended prematurely");
```

GetNextMsg() 함수 첫 번째 인자가 socket\_fd로 수정됨

read() 함수 이용 첫 번째 인자가 sockfd이고, 1바이트씩 읽어들이

## 방법 2 - 수정된 delimiterFramer.c (2)

```
28         else
29             return -1;
30     }
31     if (nextChar == DELIMITER)
32         break;
33     buf[count++] = nextChar;
34 }
35
36 if (nextChar != DELIMITER) { // Out of space: count==bufSize
37     return -count;
38 } else { // Found delimiter
39     fprintf(stderr, "GetNextMsg: %d bytes : %s\n", count, buf);
40     return count;
41 }
42 }
43
```

## 방법 2 - 수정된 delimiterFramer.c (3)

```
44 /* Write the given message to the output stream, followed by
45 * the delimiter. Return number of bytes written, or -1 on failure.
46 */
47
48 int PutMsg(uint8_t buf[], size_t msgSize, int sockfd)
49 {
50
51     // Check for delimiter in message
52     int i;
53     for (i = 0; i < msgSize; i++)
54         if (buf[i] == DELIMITER)
55             return -1;
56     if (write(sockfd, buf, msgSize) != msgSize)
57         return -1;
58
59     uint8_t last_ch = (uint8_t) DELIMITER;
60     write(sockfd, &last_ch, 1);
61
62     return msgSize;
63 }
```

PutMsg ( ) 함수 세번째 인자가 sockfd로 수정됨

write ( ) 함수 이용 첫번째 인자가 sockfd이고, 주소와 크기만 전달됨

write ( ) 함수 이용 소켓에 DELIMITER 삽입

# 방법 2 - 수정된 lengthFramer.c (1)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <stdint.h>
4 #include <unistd.h>
5 #include <netinet/in.h>
6 #include "Practical.h"
7
8 /* Read 2-byte length and place in big-endian order.
9  * Then read the indicated number of bytes.
10  * If the input buffer is too small for the data, truncate to fit and
11  * return the negation of the *indicated* length. Thus a negative return
12  * other than -1 indicates that the message was truncated.
13  * (Ambiguity is possible only if the caller passes an empty buffer.)
14  * Input stream is always left empty.
15  */
16 int GetNextMsg(int sockfd, uint8_t *buf, size_t bufSize)
17 {
18
19     uint16_t mSize = 0;
20     uint16_t extra = 0;
21
22     if (read(sockfd, &mSize, sizeof(uint16_t)) != sizeof(uint16_t))
23         return -1;
```

GetNextMsg( ) 함수 첫 번째 인자가 sockfd로 수정됨

read( ) 함수 이용 첫 번째 인자가 sockfd이고, 길이를 읽어 들임



# 방법 2 - 수정된 lengthFramer.c (2)

```
24
25     mSize = ntohs(mSize);
26
27     if (mSize > bufSize) {
28         extra = mSize - bufSize;
29         mSize = bufSize; // Truncate
30     }
31     if (read(sockfd, buf, sizeof(uint8_t)* mSize) != mSize) {
32         fprintf(stderr, "Framing error: expected %d, read less\n", mSize);
33         return -1;
34     }
35
36     if (extra > 0) { // Message was truncated
37         uint8_t waste[BUFSIZE];
38         read(sockfd, waste, sizeof(uint8_t) * extra); // Try to flush the channel
39         return -(mSize + extra); // Negation of indicated size
40     } else
41         return mSize;
42 }
```

read() 함수 이용  
첫번째 인자가 sockfd이고,  
먼저 읽은 길이만큼 를 읽어 들임

# 방법 2 - 수정된 lengthFramer.c (3)

```
43
44 /* Write the given message to the output stream, followed by
45 * the delimiter. Precondition: buf[] is at least msgSize.
46 * Returns -1 on any error.
47 */
48 int PutMsg(uint8_t buf[], size_t msgSize, int sockfd)
49 {
50     if (msgSize > UINT16_MAX)
51         return -1;
52
53     uint16_t payloadSize = htons(msgSize);
54     if ((write(sockfd, &payloadSize, sizeof(uint16_t)) != sizeof(uint16_t)) ||
55         (write(sockfd, buf, sizeof(uint8_t)* msgSize) != sizeof(uint8_t) * msgSize))
56         return -1;
57     return msgSize;
58 }
```

PutMsg() 함수 세 번째 인자가 sockfd로 수정됨

write() 함수 이용 첫 번째 인자가 sockfd이고, 주소와 크기만 전달됨