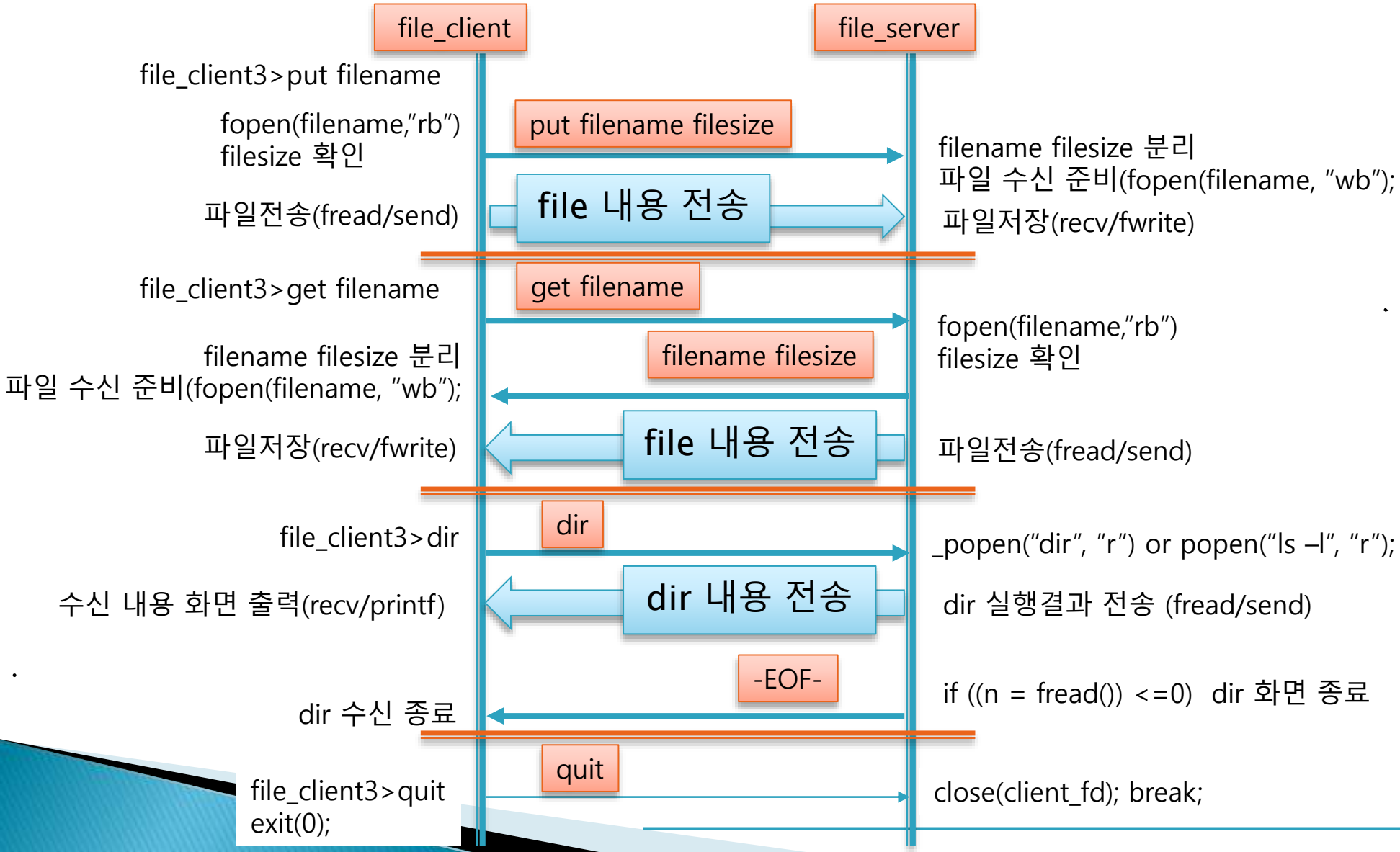
네트워크프로그래밍-6주
화상강의 자료

정인환교수

6주 화상강의 내용

- 중간고사 공지
  - 8주 화상강의 시간, Quiz 형식 (제한시간 50분)
  - 네트워크 기본, TCP/IP 이론
  - socket programming 간단한 손 코딩 및 이론
- 과제 Review
  - file_client/server 1, 2, 3, 4
  - Q&A FAQ 정리
  - 오류처리 Protocol 추가 필요성
    - get filename 에서 filename 이 Sever에 없는 경우 ??
- 6주 강의 요약
  - 채팅 client/server

file client/server 프로토콜



Q&A FAQ 정리

1. 송/수신 데이타 출력 화면이 겹치거나 찌꺼기가 보인다.
- 원인 : 바로 앞에 얽었던 data가 남아 있는 현상.
- 해결방법 : client/server의 모든 source 코드의 fgets(), fread(), recv() 함수 앞에 반드시 memset(buf, 0, BUF_LEN) 로 clear 해 주면 됩니다.

```
memset(buf, 0, BUF_LEN); fgets(buf, BUF_LEN, stdin);
memset(buf, 0, BUF_LEN); fread(buf, 1, BUF_LEN, fp);
memset(buf, 0, BUF_LEN); recv(s, buf, BUF_LEN, 0) or recv(client_fd, buf, BUF_LEN, 0);
```

2. data.txt 를 수신했는데 파일에 저장이 안된다.
- 원인 : while () 로 수신 끝나고 fclose(fp)가 안되면 data.txt 가 내용이 수정이 되지 않습니다.

```
while () {
}
fclose(fp)
```

3. dir 결과가 화면이 깨지거나 겹쳐서 나오거나 client 가 동작을 먼춘다.
- 원인 : server가 send() 를 두번 해도 client는 recv() 한번으로 수신하는 현상.

server가 dir의 결과를 client에게 보낼 때 n = fread() 의 결과가 n < 128 이면 내용을 보내고 그 다음 n<=0 으로 whil()을 끝내고 server는 "-EOF-" 를 보내고 전송을 종료 하지만 client 는 recv(s, buf, 128, 0) 로 128 을 기다리다가. server가 보낸 dir 의 결과 끝에 -EOF- 가 붙어서 읽히고 client는 -EOF- 를 감지하지 못하고 무한대기로 기다리게 됩니다.
- 해결방법 : dir 의 결과를 보낼 때 무조건 128 로 고정시켜서 보낸다.

```
// Server 코드
fp = popen()
while () {
  memset(buf, 0, 128);
  n = fread(buf, 1, 128, fp);
  if (n<=0) break;
  send(client_fd, buf, 128, 0);
}
strcpy(buf, "-EOF-");
send(client_fd, buf, 128, 0)
// Client dir 처리 코드
while (1) {
  memset(buf, 0, 128)
  recv(s, buf, 128, 0);
  if (strncmp(buf, "-EOF-", 5) == 0) break;
  printf(buf); // buf를 그대로 화면에 출력해도 된다.
}
```

1. Linux dir 오류

– 원인 : fp = popen("ls –l", "rb") 에서 Linux는 "rb"가 안되서 fp error 처리를 안하기 때문 ==> "r" 로 변경.

fp = popen("ls –l", "rb") ==> popen("ls –l", "r")

2. client 또는 server 가 get/put/dir 하면 끝나지 않고 무한 루프로 빠진다.

put/get : while (readsum < filesize) 로 정확히 제어가 되는지 확인

dir : while() 안에서 send(client_fd, buf, 128, 0) 로 고정 크기로 전송하도록 하고, 끝나면 반드시 "–EOF–"도 고정크기로 전송할 것.

4. scanf, gets, .. 등등 secure 문제 오류..
프로젝트 속성 > 구성 속성 > C/C++ > SDL 검사 --> 아니오 로 변경

5. strtok() 문제
sscanf()와 strncmp()로 해결

4주 과제 put 처리 file_client4.c

```
// put filename
if (strncmp(cmdstr, "put",3) == 0) { // put filename
    sscanf(cmdstr, "%s %s", cmd, filename);
    if ((fp = fopen(filename, "rb")) == NULL) {
        printf("file open error\n");
        continue;
    }
    // 파일이 없는 경우를 check 한다.
    // 파일이 있는 경우에만 보낸다.
    fseek(fp, 0, 2);
    filesize = ftell(fp);
    rewind(fp);
    //printf("filesize = %d\n", filesize);
    sprintf(buf, "%s %s %d", "put", filename, filesize);
    printf("Sending %s %d bytes.\n", filename, filesize);
    if (send(s, buf, BUF_LEN, 0) <= 0) { // put filename filesize를 보낸다.
        printf("send error\n");
        exit(0);
    }
```

```
    // file을 모두 읽어서 보낸다.
    readsum = 0;
    if (filesize < BUF_LEN)
        nread = filesize;
    else
        nread = BUF_LEN;
    while (readsum < filesize) {
        // n = fgets(buf, BUF_LEN, fp);
        memset(buf, 0, BUF_LEN + 1);
        n = fread(buf, 1, nread, fp); // 파일을 읽어서
        if (n <= 0) // End of file ??
            break;
        //printf("Sending %d bytes : %s\n", n, buf);
        if (send(s, buf, n, 0) <= 0) { // 읽은 bytes 만큼만 네트워크로 보낸다.
            printf("send error\n");
            break;
        }
        readsum += n;
        if ((nread = (filesize - readsum)) > BUF_LEN) // 남아있는 data 만 read
            nread = BUF_LEN;
    }
    fclose(fp);
    printf("File %s %d bytes transferred.\n", filename, filesize);
    continue;
```

4주 과제 put 처리 file_server4.c

```c
if (strncmp(buf, "put",3) == 0) { // receive file from client
    sscanf(buf, "%s %s %d", cmdstr, filename, &filesize);
    printf("Receving %s %d bytes.\n", filename, filesize);
    if ((fp = fopen(filename, "wb"))== NULL) {
        printf("file open error\n");
        exit(0);
    }
    readsum = 0;
    if (filesize < BUF_LEN)
        nread = filesize;
    else
        nread = BUF_LEN;
    memset(buf, 0, BUF_LEN);
```

```c
        while (readsum < filesize) {
            memset(buf, 0, BUF_LEN);
            n = recv(client_fd, buf, nread, 0);
            if (n <= 0) { // end of file 이면 자동 종료
                printf("\nend of file\n");
                break;
            }
            //printf("read data = %d bytes : %s", n, buf);
            if (fwrite(buf, 1, n, fp) <= 0) {
                printf("fwrite error\n");
                break;
            }
            readsum += n;
            if ((nread = (filesize - readsum)) > BUF_LEN)
                nread = BUF_LEN;
            //printf("nread = %d\n", nread);
        }
        fclose(fp);
        printf("\nFile %s %d bytes received.\n", filename, filesize);
        continue;
    }
```

4주 과제 get 처리 – file_client4.c

```
// get filename
if (strncmp(cmdstr, "get",3) == 0) { // get filename
    if (send(s, cmdstr, BUF_LEN, 0) <= 0) { // 명령어 전체를 전송
        printf("send error\n");
        exit(0);
    }

    // filename 과 filesize 를 다시 받는다.
    // file_server3 의 수신 부분과 동일
    if (recv(s, buf, BUF_LEN, 0) <= 0) {
        printf("filename recv error\n");
        exit(0);
    }
    sscanf(buf, "%s %d", filename, &filesize);
    printf("Receiving %s %d bytes.\n", filename, filesize);
    if ((fp = fopen(filename, "wb")) == NULL) {
        printf("file open error\n");
        exit(0);
    }
    readsum = 0;
    if (filesize < BUF_LEN)
        nread = filesize;
    else
        nread = BUF_LEN;
```

```
    while (readsum < filesize) {
        memset(buf, 0, BUF_LEN + 1);
        n = recv(s, buf, nread, 0);
        if (n <= 0) { // end of file 이면 자동 종료
            printf("\nend of file\n");
            break;
        }
        //printf("read data = %d bytes : %s", n, buf);
        if (fwrite(buf, n, 1, fp) <= 0) {
            printf("fwrite error\n");
            break;
        }
        readsum += n;
        if ((nread = (filesize - readsum)) > BUF_LEN)
            nread = BUF_LEN;
        //printf("nread = %d\n", nread);
    }
    fclose(fp);
    printf("\nFile %s %d bytes received.\n", filename, filesize);
    continue;
}
```

4주 과제 get 처리 – file_server.c

```
// send file from server to client
if (strncmp(buf, "get", 3) == 0) {
    sscanf(buf, "%s %s", cmdstr, filename);
    if ((fp=fopen(filename, "rb")) == NULL) {
        printf("File %s open error\n", filename);
        sprintf(buf, "%s -1", filename);
        if (send(client_fd, buf, BUF_LEN, 0) <= 0) {
            printf("filename send error\n");
        }
        continue;
    }
    fseek(fp, 0, 2);
    filesize = ftell(fp);
    rewind(fp);
    //printf("filesize = %d\n", filesize);

    // file name + file size를 다시 보내준다.
    sprintf(buf, "%s %d", filename, filesize);
    printf("Sending file %s %d bytes.\n", filename, filesize);
    if (send(client_fd, buf, BUF_LEN, 0) <= 0) { // 전송 단위는 BUF_LEN
        printf("filename send error\n");
        break;
    }
```

```
    // file을 모두 읽어서 보낸다.
    readsum = 0;
    if (filesize < BUF_LEN)
        nread = filesize;
    else
        nread = BUF_LEN;
    while (readsum < filesize) {
        // n = fgets(buf, BUF_LEN, fp);
        memset(buf, 0, BUF_LEN + 1);
        n = fread(buf, 1, nread, fp); // 파일을 읽어서
        if (n <= 0) // End of file ??
            break;
        //printf("Sending %d bytes : %s\n", n, buf);
        // 읽은 bytes 만큼만 네트워크로 보낸다.
        if (send(s, buf, n, 0) <= 0) {
            printf("send error\n");
            break;
        }
        readsum += n;
        // 남아있는 data 만 read
        if ((nread = (filesize - readsum)) > BUF_LEN)
            nread = BUF_LEN;
    }
    fclose(fp);
    printf("File %s %d bytes transferred.\n", filename, filesize);
    continue;
}
```

4주 과제 dir 처리 – file_client4.c

```
// dir
if (strncmp(cmdstr, "dir",3) == 0) { // Server directory 보기
    if (send(s, cmdstr, BUF_LEN, 0) <= 0) { // 명령어 전체를 전송
        printf("send error\n");
        exit(0);
    }
    while (1) {
        n = recv(s, buf, BUF_LEN, 0);
        if (n <= 0) // end of file 오류 메시지를 출력할 필요는 없다.
            break;
        //printf("read data = %d bytes\n", len);
        if (strncmp(buf, "-EOF-", 5) == 0)
            break;
        printf(buf);
    }
    continue;
}
```

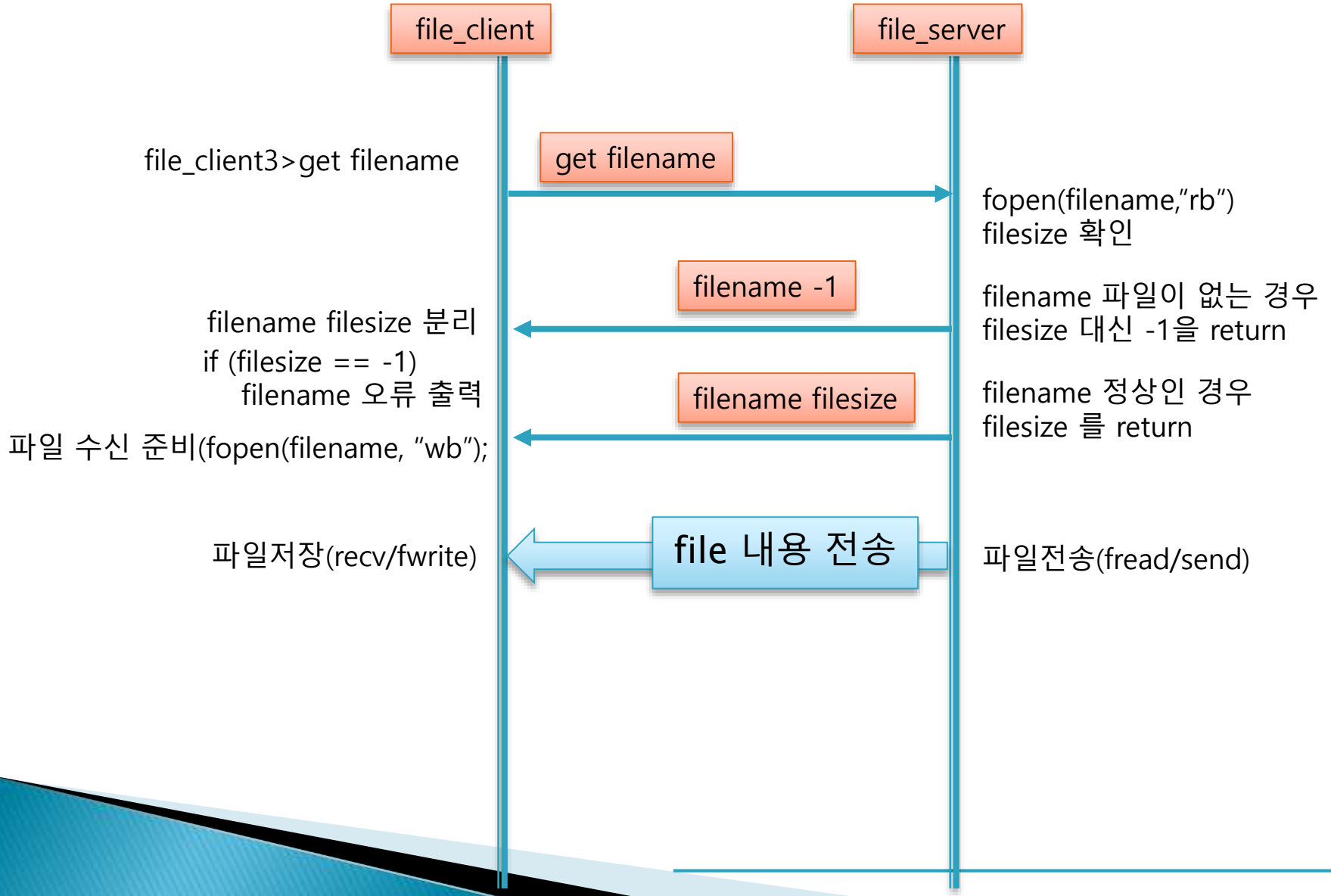
4주 과제 dir 처리 – file_server.c

```
        if (strncmp(buf, "dir", 3) == 0) { // list file name
#ifdef WIN32
            fp = _popen("dir", "r");
#else
            fp = popen("ls -l", "r");
#endif
            printf("Sending directory listing.\n");
            while (1) {
                int n;
                memset(buf, 0, BUF_LEN);
                n = fread(buf, 1, BUF_LEN, fp); // 파일을 읽어서
                if (n <= 0) // End of file
                    break;
                //fwrite(buf, 1, n, stdout);
                if (send(client_fd, buf, BUF_LEN, 0) <= 0) { // 네트워크로 보낸다.
                    printf("send error\n");
                    break;
                }
            }
            //printf("dir completed\n");
            strcpy(buf, "-EOF-"); // 끝 표시를 보낸다.
            send(client_fd, buf, BUF_LEN, 0);
#ifdef WIN32
            _pclose(fp);
#else
            pclose(fp);
#endif
            continue;
        }
```

4주 과제 ldir, !cmd 처리 – file_client4.c

```
        if (strncmp(cmdstr, "ldir",4) == 0) {
#ifdef WIN32
            system("dir");
#else
            system("ls -l");
 #endif
            continue;
        }
        if (strncmp(cmdstr,"!",1)==0) { // local system command
            system(cmdstr+1);
            continue;
        }
        if (strncmp(cmdstr, "quit", 4) == 0) {
            sprintf(buf, "quit");
            if (send(s, buf, BUF_LEN, 0) <= 0) {
                printf("send error\n");
                exit(0);
            }
            break;
        }
        // 메뉴에 없는 명령어 입력
        printf("Unknown command %s\n", cmdstr);
```

file client/server 프로토콜 – get 오류 처리 추가



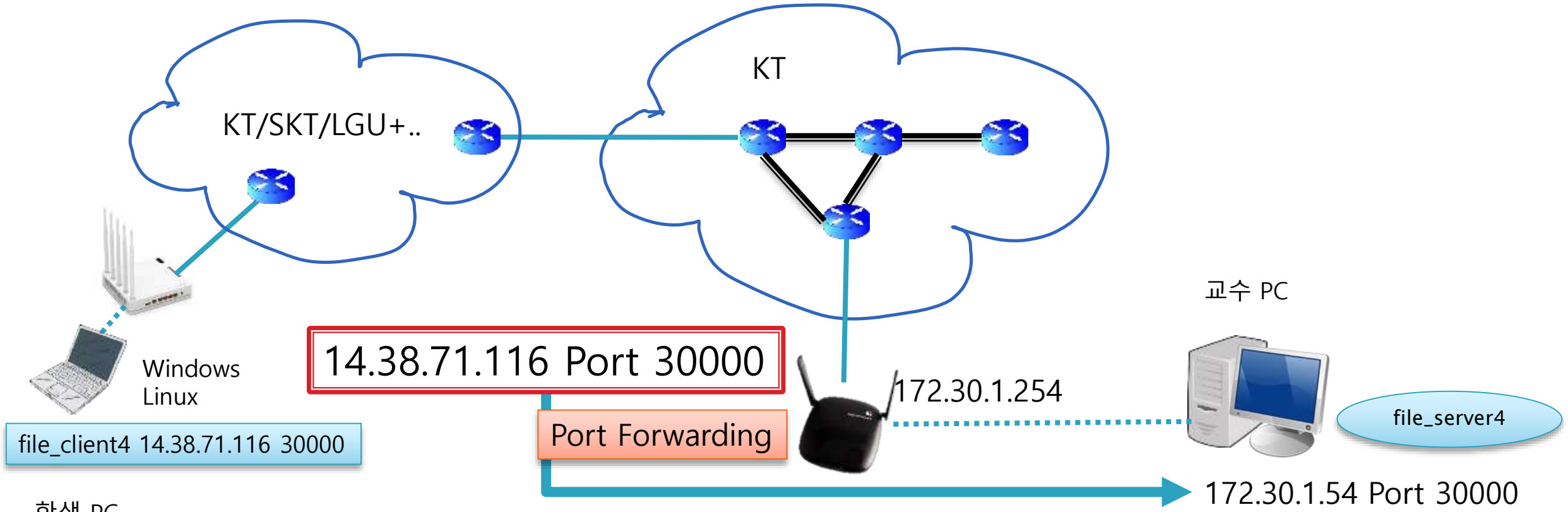
get 오류 처리 결과 file_client4.c/file_server4.c

```c
// get filename
if (strncmp(cmdstr, "get",3) == 0) { // get filename
    if (send(s, cmdstr, BUF_LEN, 0) <= 0) { // 명령어 전체를 전
        printf("send error\n");
        exit(0);
    }

    // filename 과 filesize 를 다시 받는다.
    // file_server3 의 수신 부분과 동일
    if (recv(s, buf, BUF_LEN, 0) <= 0) {
        printf("filename recv error\n");
        exit(0);
    }
    sscanf(buf, "%s %d", filename, &filesize);
    // Server에서 filename을 open 하지 못하면 (파일이 없는 경우)
    // filesize = -1 로 return 한다.
    if (filesize == -1) {
        printf("Server file %s open error.\n", filename);
        continue;
    }
    printf("Receiving %s %d bytes.\n", filename, filesize);
```

```c
// send file from server to client
if (strncmp(buf, "get", 3) == 0) {
    sscanf(buf, "%s %s", cmdstr, filename);
    if ((fp=fopen(filename, "rb")) == NULL) {
        printf("File %s open error\n", filename);
        sprintf(buf, "%s -1", filename);
        if (send(client_fd, buf, BUF_LEN, 0) <= 0) {
            printf("filename send error\n");
        }
        continue;
    }
```

학생 PC file_client <WiFi> <ISP> 교수 Windows file_server



6주 강의 요약

- 채팅
  - chat_client1/server1 : 기본 chatting
  - chat_client2/server2 : username 단순 사용, exit
  - chat_client3/server3 : 채팅 기능 추가하기
    - /list : username 목록보기, /exit
- 과제
  - chat_client4/server4
    - 입장/퇴장 알림
    - /to 귓속말
    - /sleep /wakeup
- 도전과제
  - chat_client5/server5
    - /with, /end 1:1 대화모드
  - chat_client6/server6
    - /filesend 파일 전송