네트워크프로그래밍-5주
소켓프로그래밍-파일전송

정인환교수

파일 전송

- 파일전송 프로그램 ftp
- 파일 전송 1단계 : echo_client1 키보드 대신 file 에서 입력
  - Windows / Linux Source 관리 방법
- 2 단계 : filename 전송
- 3 단계 : filename + filesize 전송
- 과제
  - ftp 프로그램 따라하기 (put, get, quit, dir) 기능 구현

파일전송 프로그램 ftp

- ftp 명령어
  - ftp [ip]
  - get filename, put filename, mget *.c, mput *.c, dir, ls, ..
- Linux ftp 설치
  - sudo apt-get install vsftpd
- Linux ftp 설정 변경
  - sudo vi /etc/vsftpd.conf
    - local_enable YES
    - write_enable YES
- ftp 시작
  - sudo /etc/init.d/vsftpd

ftp 사용 예

명령 프롬프트

```
C:₩Users₩Daddy>cd ₩ftp-test

C:₩ftp-test>dir
 C 드라이브의 볼륨에는 이름이 없습니다.
 볼륨 일련 번호: 7ABC-9B16

 C:₩ftp-test 디렉터리

2020-09-26  오전 10:06    <DIR>          .
2020-09-26  오전 10:06    <DIR>          ..
2020-09-26  오전 10:06                30 windows-data.txt
               1개 파일                  30 바이트
               2개 디렉터리  153,006,182,400 바이트 남음

C:₩ftp-test>ftp 192.168.126.130
192.168.126.130에 연결되었습니다.
220 (vsFTPd 3.0.3)
200 Always in UTF8 mode.
사용자(192.168.126.130:(none)): user
331 Please specify the password.
암호:
230 Login successful.
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxrwxr-x    2 1000     1000         4096 Sep 26 10:07 ftp-test
drwxrwxr-x    5 1000     1000         4096 Sep 25 08:59 netprog
drwxr-xr-x    3 1000     1000         4096 Aug 24 22:44 snap
drwxr-xr-x    2 1000     1000         4096 Aug 13 09:12 공개
drwxr-xr-x    3 1000     1000         4096 Sep 22 17:58 다운로드
drwxr-xr-x    2 1000     1000         4096 Aug 13 09:12 문서
drwxr-xr-x    2 1000     1000         4096 Aug 13 09:12 바탕화면
drwxr-xr-x    2 1000     1000         4096 Aug 13 09:12 비디오
drwxr-xr-x    2 1000     1000         4096 Sep 17 00:23 사진
drwxr-xr-x    2 1000     1000         4096 Aug 13 09:12 음악
```

명령 프롬프트

```
drwxr-xr-x    2 1000     1000         4096 Aug 13 09:12 템플릿
226 Directory send OK.
ftp: 0.03초 25.03KB/초
ftp> cd ftp-test
250 Directory successfully changed.
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-rw-r--    1 1000     1000           28 Sep 26 10:07 linux-data.txt
226 Directory send OK.
ftp: 0.00초 25.00KB/초
ftp> get linux-data.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for linux-data.txt (28 bytes).
226 Transfer complete.
ftp: 0.00초 28000.00KB/초
ftp> !dir
 C 드라이브의 볼륨에는 이름이 없습니다.
 볼륨 일련 번호: 7ABC-9B16

 C:₩ftp-test 디렉터리

2020-09-26  오전 10:08    <DIR>          .
2020-09-26  오전 10:08    <DIR>          ..
2020-09-26  오전 10:08                28 linux-data.txt
2020-09-26  오전 10:06                30 windows-data.txt
               2개 파일                  58 바이트
               2개 디렉터리  153,005,723,648 바이트 남음
ftp> put windows-data.txt
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
ftp: 0.00초 30.00KB/초
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
```

명령 프롬프트

```
-rw-rw-r--    1 1000     1000           28 Sep 26 10:07 linux-data.txt
-rw-------    1 1000     1000           30 Sep 26 10:08 windows-data.txt
226 Directory send OK.
ftp: 0.01초 21.29KB/초
ftp> !del *.txt
ftp> !dir
 C 드라이브의 볼륨에는 이름이 없습니다.
 볼륨 일련 번호: 7ABC-9B16

 C:₩ftp-test 디렉터리

2020-09-26  오전 10:09    <DIR>          .
2020-09-26  오전 10:09    <DIR>          ..
               0개 파일                   0 바이트
               2개 디렉터리  153,005,719,552 바이트 남음
ftp> mget *.txt
200 Switching to ASCII mode.
mget linux-data.txt? y
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for linux-data.txt (28 bytes).
226 Transfer complete.
ftp: 0.00초 28.00KB/초
mget windows-data.txt? y
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for windows-data.txt (30 bytes).
226 Transfer complete.
ftp: 0.00초 30000.00KB/초
ftp> !dir
 C 드라이브의 볼륨에는 이름이 없습니다.
 볼륨 일련 번호: 7ABC-9B16

 C:₩ftp-test 디렉터리

2020-09-26  오전 10:09    <DIR>          .
2020-09-26  오전 10:09    <DIR>          ..
2020-09-26  오전 10:09                28 linux-data.txt
```

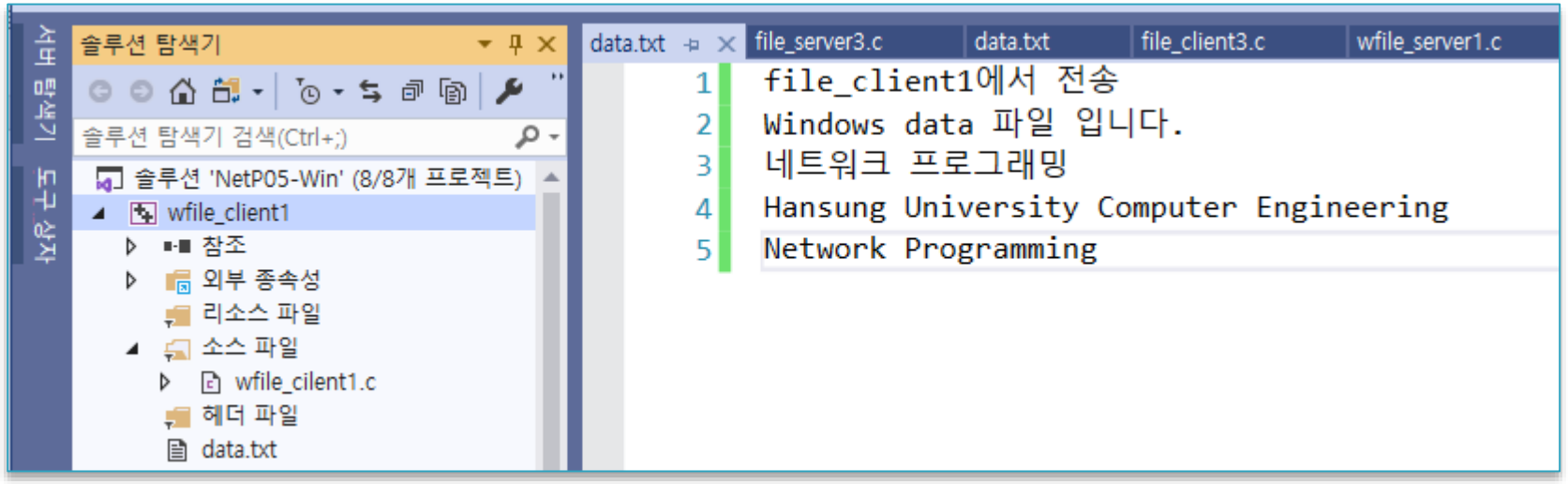
명령 프롬프트

```
 C:₩ftp-test 디렉터리

2020-09-26  오전 10:09    <DIR>          .
2020-09-26  오전 10:09    <DIR>          ..
2020-09-26  오전 10:09                28 linux-data.txt
2020-09-26  오전 10:09                30 windows-data.txt
               2개 파일                  58 바이트
               2개 디렉터리  153,005,719,552 바이트 남음
ftp> del *.txt
550 Delete operation failed.
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
226 Directory send OK.
ftp> mput *.txt
mput linux-data.txt? y
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
ftp: 0.00초 28000.00KB/초
mput windows-data.txt? y
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
ftp: 0.00초 30000.00KB/초
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-------    1 1000     1000           28 Sep 26 10:09 linux-data.txt
-rw-------    1 1000     1000           30 Sep 26 10:09 windows-data.txt
226 Directory send OK.
ftp: 0.01초 14.90KB/초
ftp> quit
221 Goodbye.
```

File 전송 1단계 : Client 파일 읽어서 echo 요청

- Windows echo_server1 그대로 이름만 변경
  - file_server1 [port]
- Windows echo_client1.c 수정
  - file_client1 [ip] [port]
  - 실행하면 file명을 입력
  - Keyboard 입력 대신 File 입력으로 수정

file_client1 실행 폴더에 data.txt 추가



file_client1에서 전송
Windows data 파일 입니다.
네트워크 프로그래밍
Hansung University Computer Engineering
Network Programming

wfile_client1/wfile_server1 실행 화면

C:₩Windows₩system32₩cmd.exe

```
file_server1 waiting connection..
server_fd = 252
Server : waiting connection request.
Client connected from 127.0.0.1:3944
client_fd = 256
Received len=22 : file_client1에서 전송
Sending len=22 : file_client1에서 전송
Received len=26 : Windows data 파일 입니다.
Sending len=26 : Windows data 파일 입니다.
Received len=20 : 네트워크 프로그래밍
Sending len=20 : 네트워크 프로그래밍
Received len=40 : Hansung University Computer Engineering
Sending len=40 : Hansung University Computer Engineering
Received len=19 : Network ProgrammingSending len=19 : Network Programmingrecv er
ror
```

file_server1

C:₩Windows₩system32₩cmd.exe

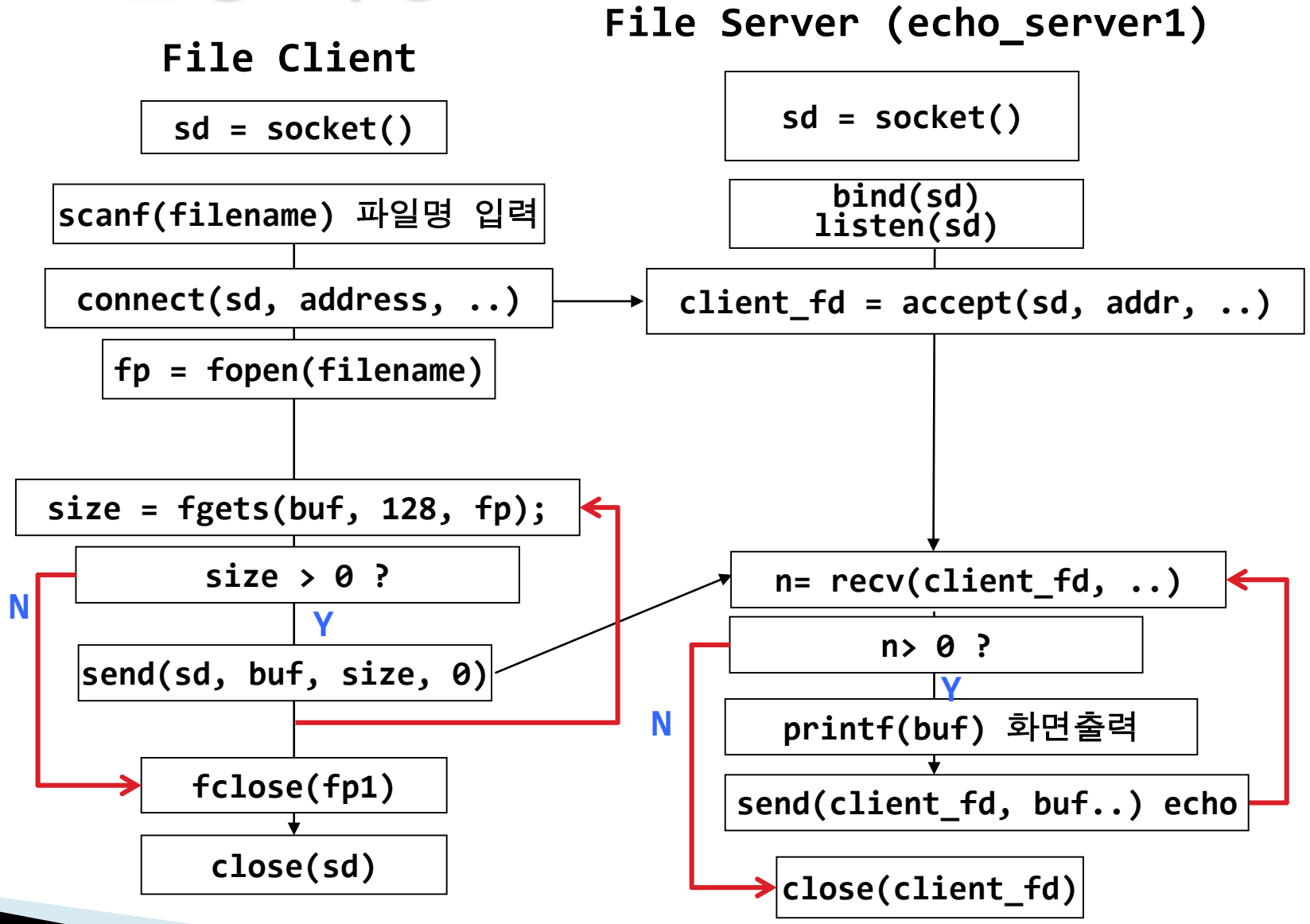
```
Enter file name : data.txt
Connecting 127.0.0.1 30000
Sending len=22 : file_client1에서 전송
Received len=22 : file_client1에서 전송
Sending len=26 : Windows data 파일 입니다.
Received len=26 : Windows data 파일 입니다.
Sending len=20 : 네트워크 프로그래밍
Received len=20 : 네트워크 프로그래밍
Sending len=40 : Hansung University Computer Engineering
Received len=40 : Hansung University Computer Engineering
Sending len=19 : Network ProgrammingReceived len=19 : Network ProgrammingEnd of
file
계속하려면 아무 키나 누르십시오 . . .
```

file_client1

file_client1\data.txt

file_client1에서 전송
Windows data 파일 입니다.
네트워크 프로그래밍
Hansung University
Computer Engineering
Network Programming

file_client1 전송 과정



file_client1.c

```c
// 파일명 입력
FILE* fp;
char filename[BUF_LEN] = "data.txt"; // data file 예
printf("Enter file name : ");
scanf("%s", filename);
getchar(); // Enter key 처리.
if ((fp = fopen(filename, "r")) == NULL) {
    printf("Can't open file %s\n", filename);
    exit(0);
}

/* 연결요청 */
printf("Connecting %s %s\n", ip_addr, port_no);

if (connect(s, (struct sockaddr*)&server_addr,
    sizeof(server_addr)) < 0) {
    printf("can't connect.\n");
    exit(0);
}
```

```c
while (1) {
    /* 파일에서 입력을 받음 */
    //printf("Input string : ");
    //if (fgets(buf, BUF_LEN, stdin)) { // gets(buf);
    if (fgets(buf, BUF_LEN, fp)) {
        len_out = strlen(buf);
        buf[BUF_LEN] = '\0';
    }
    else {
        //printf("fgets error\n");
        printf("\End of file\n");
        exit(0);
    }
    /* echo 서버로 메시지 송신 */
    printf("Sending len=%d : %s", len_out, buf);
    if (send(s, buf, len_out, 0) < 0) {
        printf("send error\n");
        exit(0);
    }
    if (strcmp(buf, "exit\n") == 0)
        break;
    if ((n = recv(s, buf, BUF_LEN, 0)) < 0) {
        printf("recv error\n");
        exit(0);
    }
    buf[n] = '\0'; // 문자열 끝에 NULL 추가
    printf("Received len=%d : %s", n, buf);
}
closesocket(s);
```

Windows/Linux Source 하나의 파일로 관리

- C 언어의 #ifdef / #else / #endif 사용
- Windows 의 경우 WIN32 가 기본 정의되어 있다.
  - #ifdef WIN32
    - // Windows Code
  - #else
    - // Linux Code
  - #endif

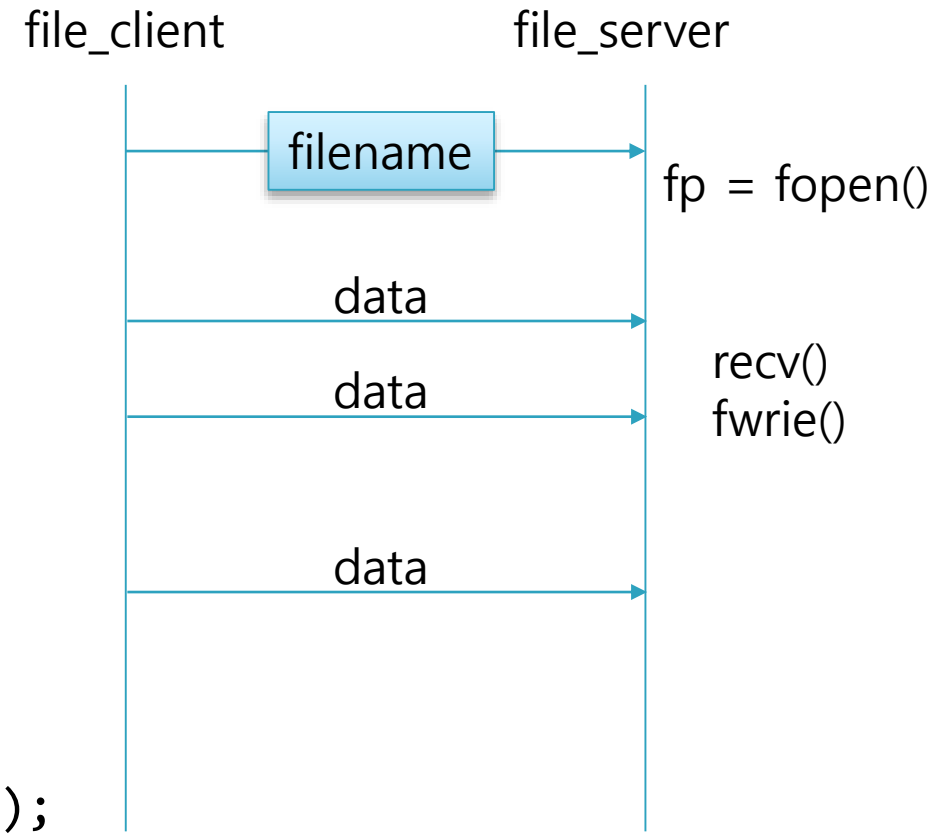
```
// header file 설정
#ifdef WIN32
#include <winsock.h>
#include <signal.h>
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#else
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#endif
```

```
#ifdef WIN32
WSADATA wsadata;
int main_socket;
void exit_callback(int sig)
{
 closesocket(main_socket);
 WSACleanup();
 exit(0);
}
void init_winsock()
{
 WORD sversion;
 u_long iMode = 1;
// winsock 사용을 위해 필수적임
 signal(SIGINT, exit_callback);
 sversion = MAKEWORD(1, 1);
 WSAStartup(sversion,
&wsadata);
}
#endif
```

```
// 처음 시작할 때
#ifdef WIN32
  printf("Windows : ");
  init_winsock();
#else // Linux
  printf("Linux : ");
#endif

if ((s = socket(PF_INET,
SOCK_STREAM, 0)) < 0) {
     printf("can't create
     socket\n");
     exit(0);
}
#ifdef WIN32
     main_socket = s;
#endif
// 끝나기 전에
#ifdef WIN32
     closesocket(s);
#else
     close(s);
#endif
```

2단계: filename 송신/저장

- Client
  - connect() 다음
    - send(sd, filename, 128, 0); // 파일명
  - fgets() 대신 n = fread(buf, 1, 128, fp);
  - send(s, buf, n, 0);
- Server 파일명 수신 및 file open
  - recv(client_fd, filename, 128, 0);
  - fp = fopen(filename, “wb”);
- Server 수신된 파일 내용 저장
  - n = recv(client_fd, buf, 128, 0);
  - printf() 대신 fwrite(buf, 1, n, fp);
- 문제점 ?
  - send / recv 사용 횟수를 줄여야 함
    - n = fread(buf, 1, 2048); send(s, buf, n, 0);
  - End of file check가 어려울 수 있다.
    - file size를 미리 통보하는 방법



file_client2 실행 폴더에 data.txt 추가

file_clinet2에서 전송
Windows data 파일 입니다.
네트워크 프로그래밍
Hansung University Computer Engineering
Network Programming

And so, my fellow Americans:
ask not what your country can do for you--ask what you can do for your country.
My fellow citizens of the world: ask not what America will do for you,
but what together we can do for the freedom of man.
Finally, whether you are citizens of America or citizens of the world,
ask of us here the same high standards of strength and sacrifice which we ask of you.
With a good conscience our only sure reward, with history the final judge of our deeds,
let us go forth to lead the land we love, asking His blessing and His help,
but knowing that here on earth God's work must truly be our own.

file_client2/file_server2 실행화면

C:₩Windows₩system32₩cmd.exe

file_server2

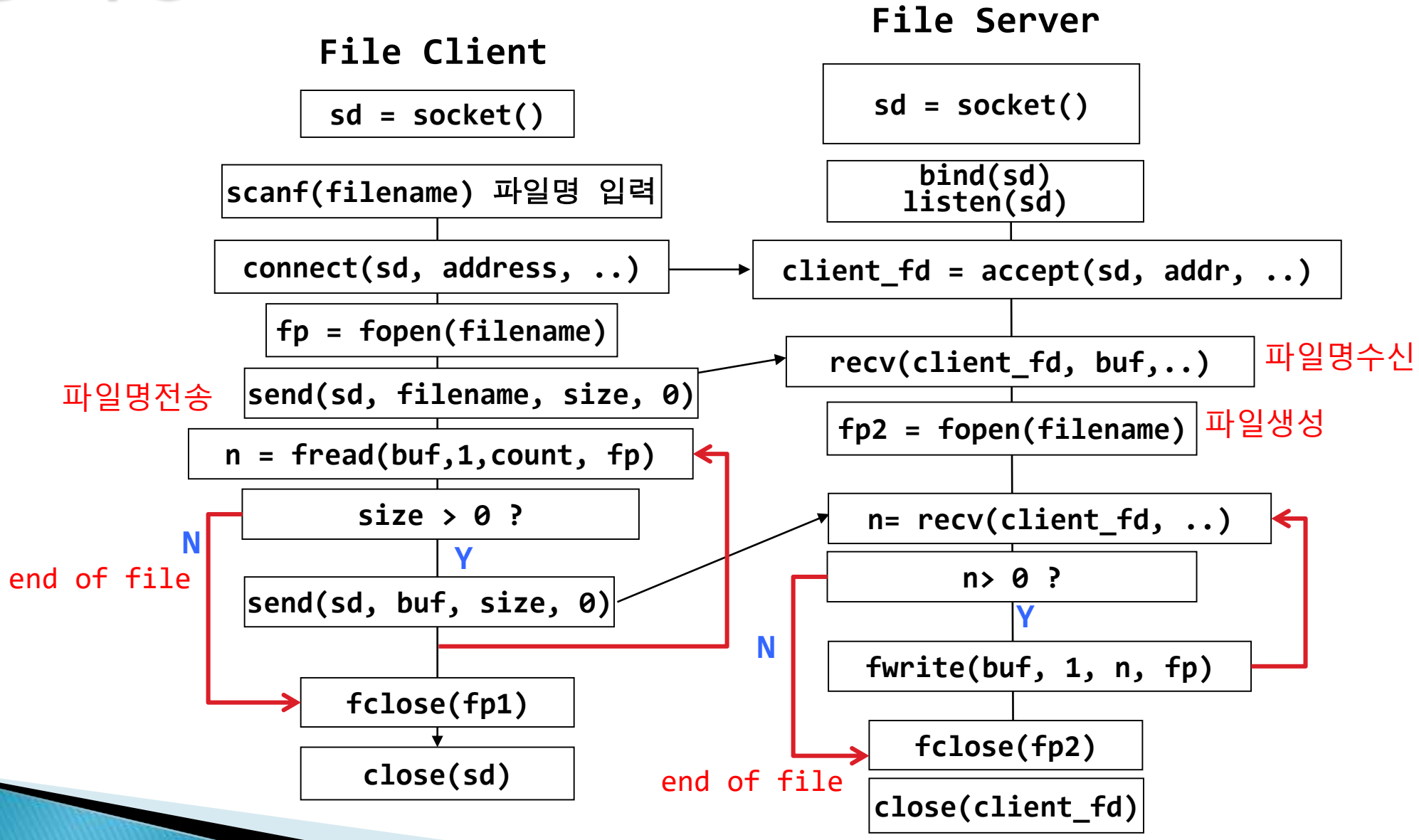
```
Server : waiting connection request.
Client connected from 127.0.0.1:11569
client_fd = 268
Received filename : data.txt
read data = 128 bytes : file_clinet2에서 전송
Windows data 파일 입니다.
네트워크 프로그래밍
Hansung University Computer Engineering
Network Programming
read data = 128 bytes :
And so, my fellow Americans:
ask not what your country can do for you--ask what you
can do for your country.
My fellow citizenread data = 128 bytes : s of the world
: ask not what America will do for you,
but what together we can do for the freedom of man.
Finally, whether you read data = 128 bytes : are citize
ns of America or citizens of the world,
ask of us here the same high standards of strength and
sacrifice which we askread data = 128 bytes :  of you.
With a good conscience our only sure reward, with histo
ry the final judge of our deeds,
let us go forth to lead the laread data = 111 bytes : n
d we love, asking His blessing and His help,
but knowing that here on earth God's work must truly be
 our own.
recv error : end of file
```

C:₩Windows₩system32₩cmd.exe

file_client2

```
Enter file name : data.txt
Connecting 127.0.0.1 30000
Sending 128 bytes : file_clinet2에서 전송
Windows data 파일 입니다.
네트워크 프로그래밍
Hansung University Computer Engineering
Network Programming

Sending 128 bytes :
And so, my fellow Americans:
ask not what your country can do for you--ask what you can do for your country.
My fellow citizen
Sending 128 bytes : s of the world: ask not what America will do for you,
but what together we can do for the freedom of man.
Finally, whether you
Sending 128 bytes : are citizens of America or citizens of the world,
ask of us here the same high standards of strength and sacrifice which we ask
Sending 128 bytes :  of you.
With a good conscience our only sure reward, with history the final judge of our
 deeds,
let us go forth to lead the la
Sending 111 bytes : nd we love, asking His blessing and His help,
but knowing that here on earth God's work must truly be our own.
계속하려면 아무 키나 누르십시오 . . .
```

전송 과정


File Client
sd = socket()
scanf(filename) 파일명 입력
connect(sd, address, ..)
fp = fopen(filename)
파일명전송
send(sd, filename, size, 0)
n = fread(buf,1,count, fp)
size > 0 ?
N
end of file
Y
send(sd, buf, size, 0)
fclose(fp1)
close(sd)
File Server
sd = socket()
bind(sd)
listen(sd)
client_fd = accept(sd, addr, ..)
recv(client_fd, buf,..)
파일명수신
fp2 = fopen(filename)
파일생성
n= recv(client_fd, ..)
n> 0 ?
N
Y
fwrite(buf, 1, n, fp)
fclose(fp2)
end of file
close(client_fd)

file_client2.c/file_server2.c

```c
if ((fp = fopen(filename, "rb")) == NULL) {
    printf("Can't open file %s\n", filename);
    exit(0);
}

/* 연결요청 */
printf("Connecting %s %s\n", ip_addr, port_no);

if (connect(s, (struct sockaddr*)&server_addr,
    sizeof(server_addr)) < 0) {
    printf("can't connect.\n");
    exit(0);
}

//fseek(fp, 0, 2);
//filesize = ftell(fp);
//rewind(fp);
//printf("filesize = %d\n", filesize);

// file name 을 보낸다.
if (send(s, filename, BUF_LEN, 0) <= 0) { // 전송 단위는 BUF_LEN
    printf("filename send error\n");
    exit(0);
}
// file을 모두 읽어서 보낸다.
while (1) {
    int n;
    // n = fgets(buf, BUF_LEN, fp);
    memset(buf, 0, BUF_LEN + 1);
    n = fread(buf, 1, BUF_LEN, 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;
    }
}
fclose(fp);
closesocket(s);
```

fopen(filename, "rb")

send()

fread()

send()

```c
printf("Client connected from %s:%d\n", inet_ntoa(client_a
printf("client_fd = %d\n", client_fd);

char filename[BUF_LEN];
FILE* fp;
if (recv(client_fd, filename, BUF_LEN, 0) <= 0) {
    printf("filename recv error\n");
    exit(0);
}
printf("Received filename : %s\n", filename);
if ((fp = fopen(filename, "wb")) == NULL) {
    printf("file open error\n");
    exit(0);
}
while (1) {
    int i, len;
    memset(buf, 0, BUF_LEN+1);
    msg_size = recv(client_fd, buf, BUF_LEN, 0);
    if (msg_size <= 0) { // end of file 이면 자동 종료
        printf("\nrecv error : end of file\n");
        break;
    }
    printf("read data = %d bytes : %s", msg_size, buf);
    if (fwrite(buf, msg_size, 1, fp) <= 0) {
        printf("fwrite error\n");
        break;
    }
}
fclose(fp);
closesocket(client_fd); // close(client_fd);
```

recv()

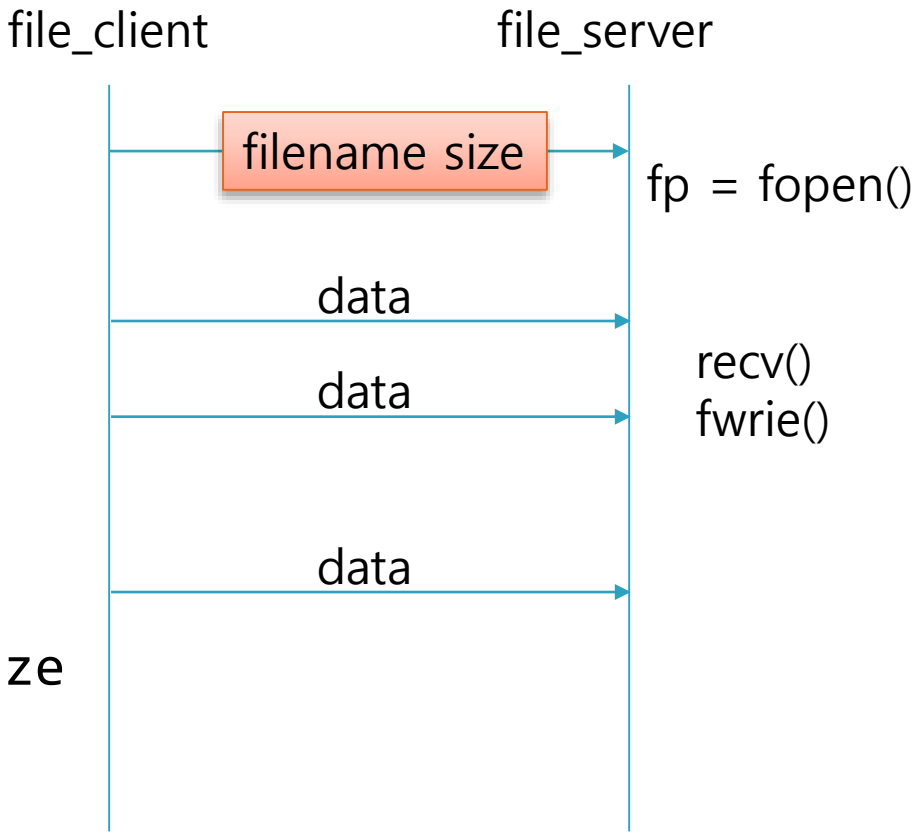
fopen(filename, "wb")

recv()

fwrite()

3단계: file size 적용하기

- file size 알기
  - fseek(fp, 0, 2);
  - filesize = ftell(fp); rewind(fp);
- Client
  - filename 과 filesize 같이 전송
  - sprintf(buf,“%s %d”, filename, filesize);
  - send(s, buf, 128, 0);
  - while (readsum < filesize)
    - n = fread(buf,nread,fp); readsum += n;
    - send(s, buf, n, 0);
- Server
  - recv(client_fd, buf, 128, 0); // filename size
  - sscanf(buf, “%s %d”, filename, &filesize);
  - while (readsum < filesize)
    - n = recv();
    - fwrite(buf, 1, n, fp);
    - readsum += n;



file_client3/file_server3 실행화면

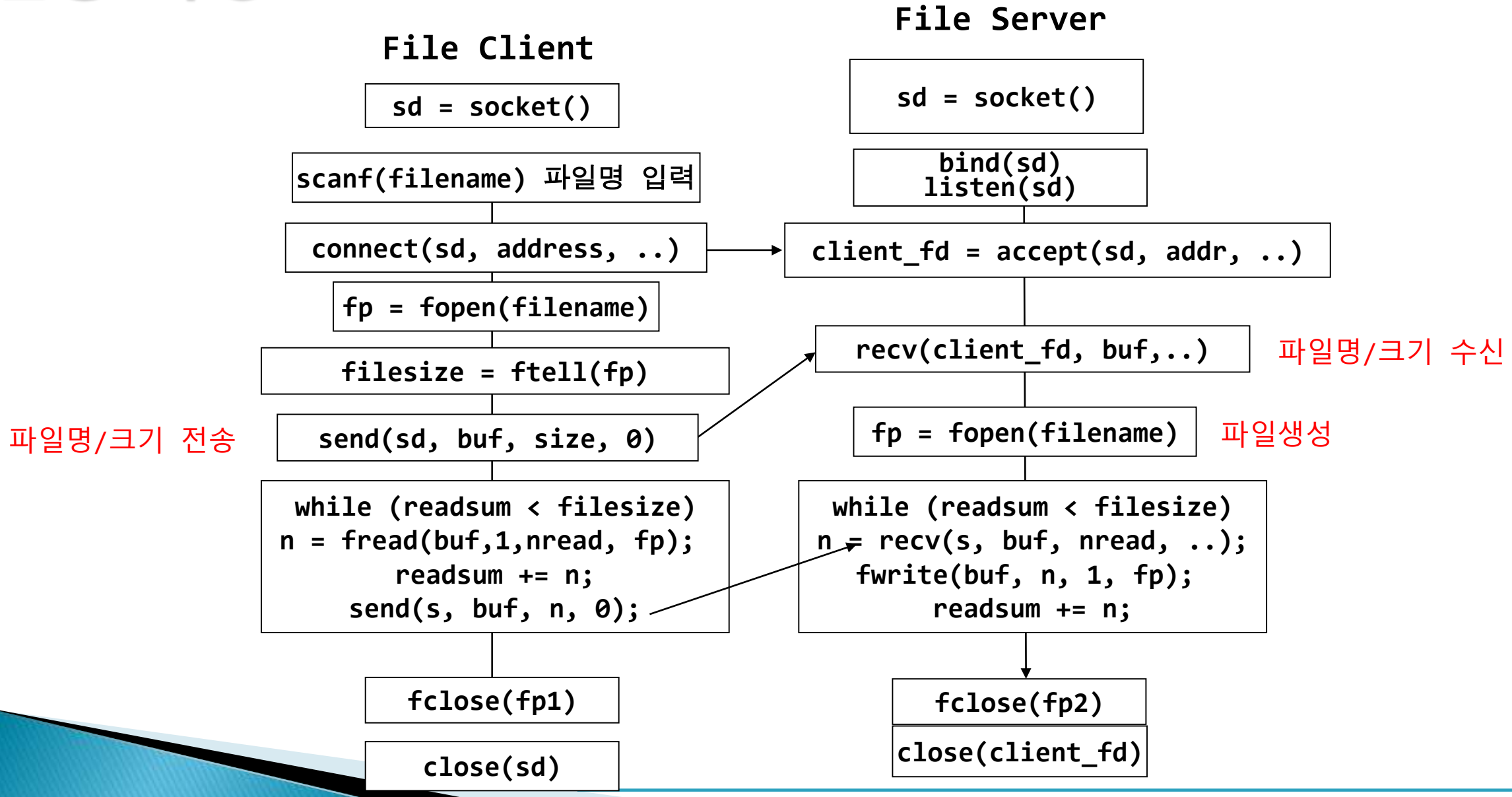
file_server3

```
C:₩Windows₩system32₩cmd.exe
Windows : file_server3 waiting connection..
server_fd = 260
Server : waiting connection request.
Client connected from 127.0.0.1:6755
client_fd = 216
Received filename : data.txt size = 765
read data = 128 bytes : file_clinet30에서 전송
Windows data 파일 입니다.
네트워크 프로그래밍
Hansung University Computer Engineering
Network Programmread data = 128 bytes : ing

And so, my fellow Americans:
ask not what your country can do for you--ask what y
ou can do for your country.
My fellowread data = 128 bytes :  citizens of the wo
rld: ask not what America will do for you,
but what together we can do for the freedom of man.
Finally, whread data = 128 bytes : ether you are cit
izens of America or citizens of the world,
ask of us here the same high standards of strength a
nd sacrifice wread data = 128 bytes : hich we ask of
 you.
With a good conscience our only sure reward, with hi
story the final judge of our deeds,
let us go forth tread data = 125 bytes : o lead the
land we love, asking His blessing and His help,
but knowing that here on earth God's work must truly
 be our own.h t
File data.txt 765 receive completed.
```

file_client3

```
C:₩Windows₩system32₩cmd.exe
Enter file name : data.txt
Connecting 127.0.0.1 30000
filesize = 765
Sending 128 bytes : file_clinet30에서 전송
Windows data 파일 입니다.
네트워크 프로그래밍
Hansung University Computer Engineering
Network Programm
Sending 128 bytes : ing

And so, my fellow Americans:
ask not what your country can do for you--ask what you can
 do for your country.
My fellow
Sending 128 bytes :  citizens of the world: ask not what A
merica will do for you,
but what together we can do for the freedom of man.
Finally, wh
Sending 128 bytes : ether you are citizens of America or c
itizens of the world,
ask of us here the same high standards of strength and sac
rifice w
Sending 128 bytes : hich we ask of you.
With a good conscience our only sure reward, with history
the final judge of our deeds,
let us go forth t
Sending 125 bytes : o lead the land we love, asking His bl
essing and His help,
but knowing that here on earth God's work must truly be ou
r own.
계속하려면 아무 키나 누르십시오 . . .
```

전송 과정


File Client
File Server
sd = socket()
sd = socket()
scanf(filename) 파일명 입력
bind(sd)
listen(sd)
connect(sd, address, ..)
client_fd = accept(sd, addr, ..)
fp = fopen(filename)
recv(client_fd, buf,..)
파일명/크기 수신
filesize = ftell(fp)
파일명/크기 전송
send(sd, buf, size, 0)
fp = fopen(filename)
파일생성
while (readsum < filesize)
n = fread(buf,1,nread, fp);
readsum += n;
send(s, buf, n, 0);
while (readsum < filesize)
n = recv(s, buf, nread, ..);
fwrite(buf, n, 1, fp);
readsum += n;
fclose(fp1)
fclose(fp2)
close(sd)
close(client_fd)

file_client3.c file_server3.c

```c
int filesize;
int readsum = 0, nread;
fseek(fp, 0, 2);
filesize = ftell(fp);
rewind(fp);
printf("filesize = %d\n", filesize);

// file name + file size를 보낸다.
sprintf(buf, "%s %d", filename, filesize);
if (send(s, buf, BUF_LEN, 0) <= 0) { // 전송 단위는 BUF_LEN
    printf("filename send error\n");
    exit(0);
}
// file을 모두 읽어서 보낸다.
readsum = 0;
if (filesize < BUF_LEN)
    nread = filesize;
else
    nread = BUF_LEN;
while (readsum < filesize) {
    int n;
    // 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);
closesocket(s);
```

fopen(filename, "rb")

send()

fread()

send()

```c
char filename[BUF_LEN];
FILE* fp;
int filesize, readsum = 0, nread = 0, n;
if (recv(client_fd, buf, BUF_LEN, 0) <= 0) {
    printf("filename recv error\n");
    exit(0);
}
sscanf(buf, "%s %d", filename, &filesize);
printf("Received filename : %s size = %d\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 + 1);
while (readsum < filesize) {
    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, 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);
closesocket(client_fd); // close(client_fd);
```

recv()

fopen(filename, "wb")

recv()

fwrite()

5주 과제1: 강의내용 복습

- 1단계 : file_client1/server1
  - keyboard 대신 파일 읽어서 echo
- 2단계 : file_client2/server2
  - filename 전송
  - recv() 오류로 end of file 처리
- 3단계 : file_client3/server3
  - filename + filesize 전송
  - 정확하게 readsum < filesize 일때까지 송/수신

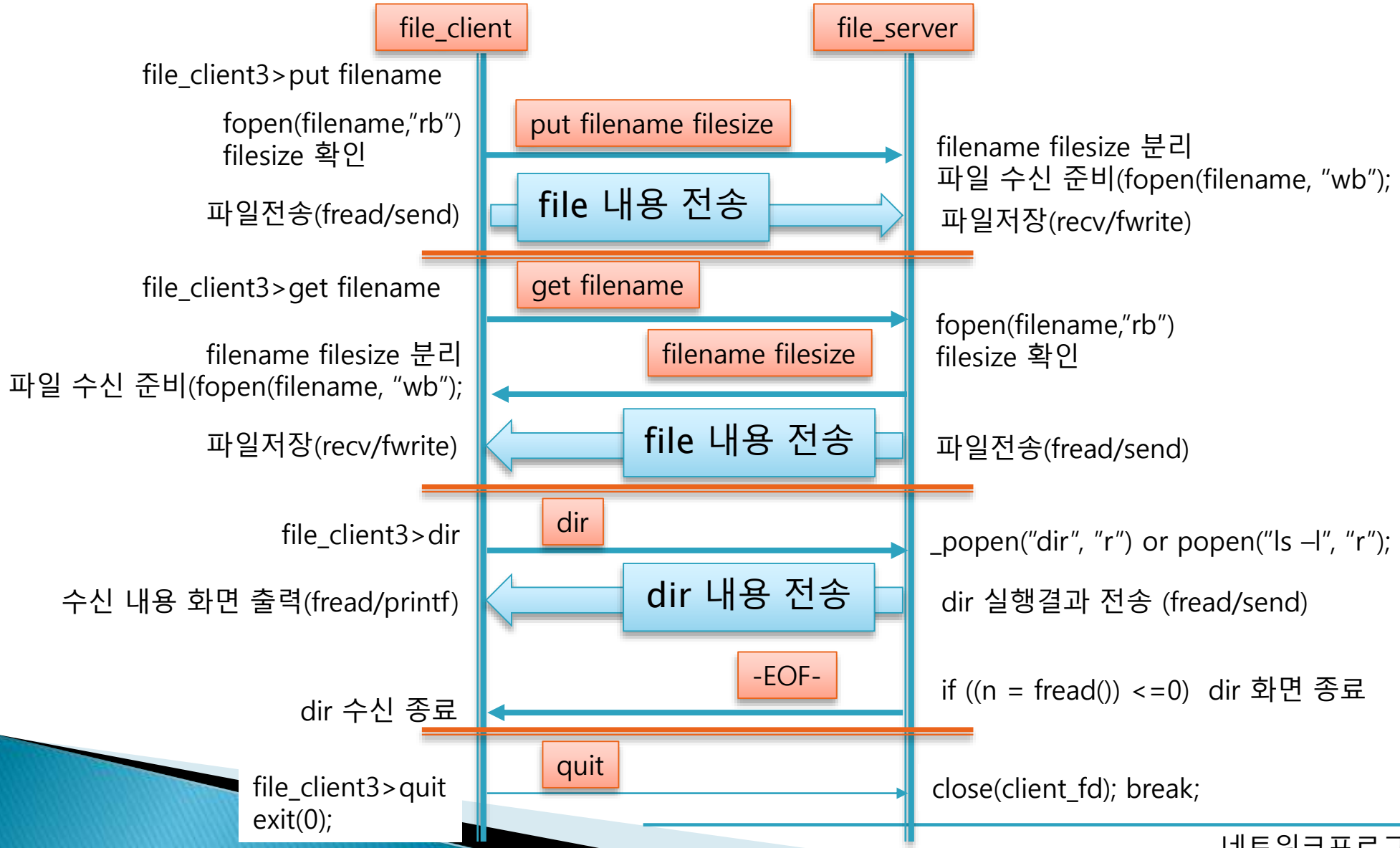
5주 과제2: 양방향 파일 전송

- 현재 : Windows → Windows/Linux 한방향 파일 전송
- 과제 : Windows ←→ Windows/Linux 양방향 전송
- 명령어 종류
  - get, put, quit, dir, ldir, !cmd
- 명령어 실행 예
  - file_client4>put filename
    - client → server 파일 전송
  - file_client4>dir
    - server 파일 목록 보기 (Windows dir / Linux ls –l)
  - file_client4>get filename
    - client ← server 파일 가져오기
  - file_client4>quit
    - client 접속 종료
    - server는 다음 client 서비스

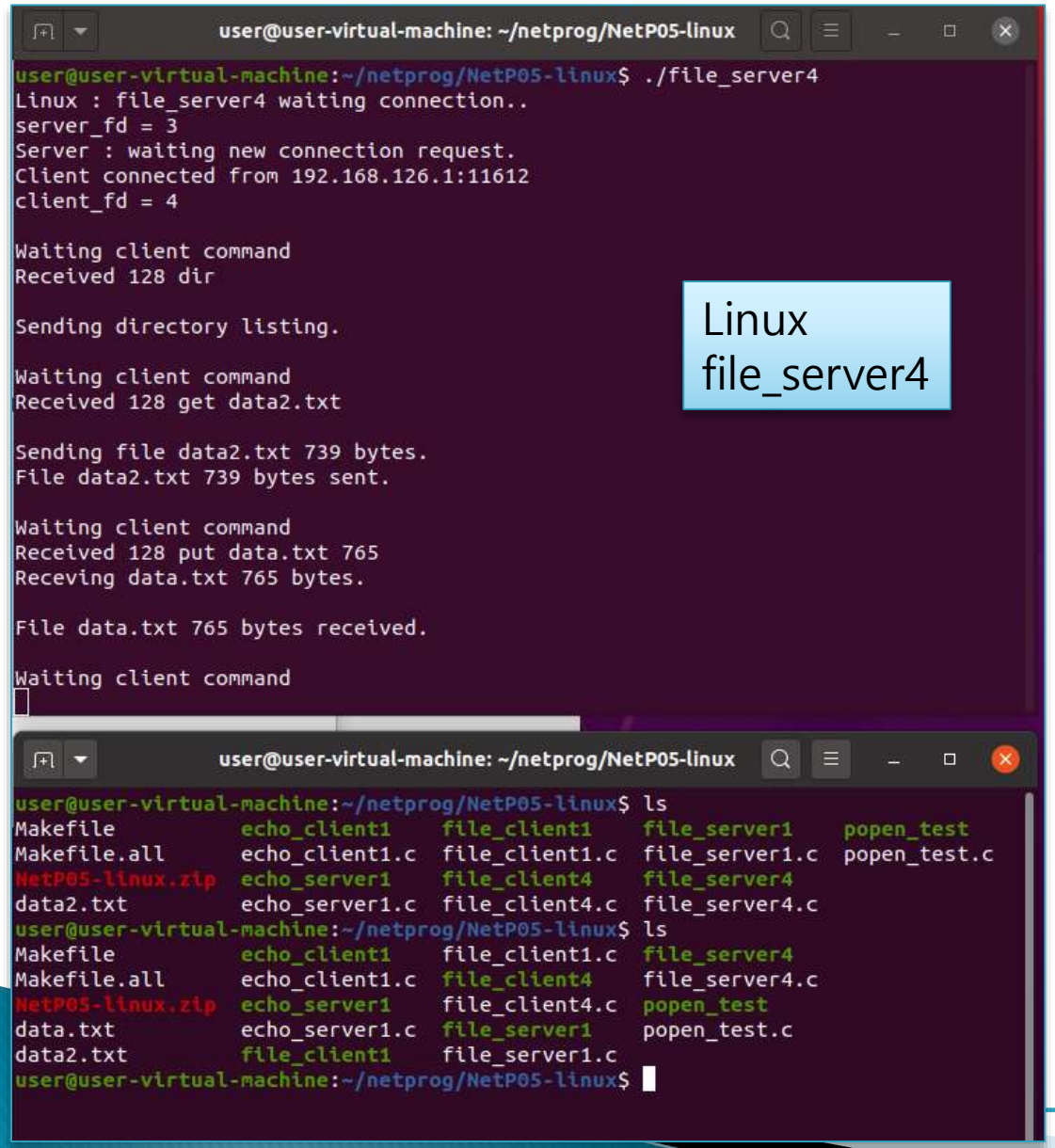
명령어 구현 방법

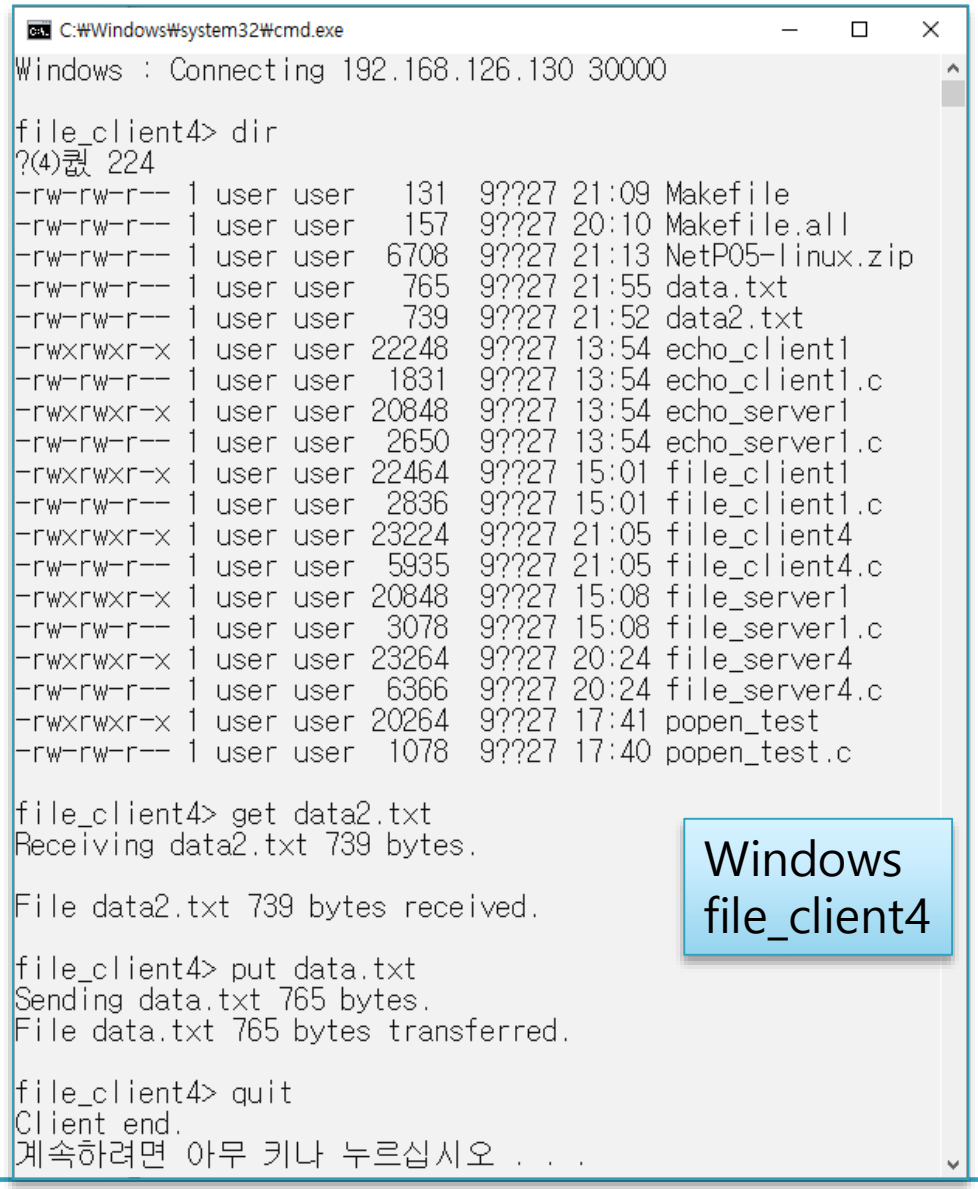
- put 명령어
  - file_client3 의 파일 전송 부분 그대로 이용
  - Client(fread/send) → Server(recv/fwrite)
- get 명령어 : put 명령어 반대로
  - client가 server에게 filename을 먼저 보낸다.
  - server가 client에게 filesize를 알려 온다.
  - 그 다음은 put 명령어 처리와 동일
  - Client(recv/fwrite) ← Server(fread/send)
  - file_client4
    - fopen(“rb”) 대신 fopen(“wb”)
    - fread() 대신 recv()
    - send() 대신 fwrite()
  - file_server4
    - fopen(“wb”) 대신 fopen(“rb”);
    - recv() 대신 fread()
    - fwrite() 대신 send()
- quit 명령어
  - echo_client4/server4 의 exit 와 같은 방법으로 구현

file client/server 프로토콜



file_client4/file_server4 실행 화면





dir 명령어

- Sever → Client 파일 전송과 같은 원리 이용
  - Server에서 특정 파일 대신
  - Windows/Linux 명령어를 실행하고 그 결과를 파일처럼 읽어서 보내준다
- file_server 에서 pipe 함수 이용
  - FILE *fp;
  - fp = fopen(filename, “rb”); 대신
  - fp = _popen(“dir”, “r”); // Linux 는 popen(“ls -l”, “r”);
  - while (1) {
    - n = fread(buf, 1, 128, fp); // recv();
    - if (n <= 0)
  - break;
    - send(client_fd, buf, 128, 0); // printf(buf);
  - }
  - strcpy(buf, “-EOF-”); // dir 명령어가 끝났다는 표시를 보내준다.
  - send(client_fd, buf, 128, 0); // strncmp(buf,”-EOF-”, 5)==0 ??
- popen_test.c 참고

popen_test.c

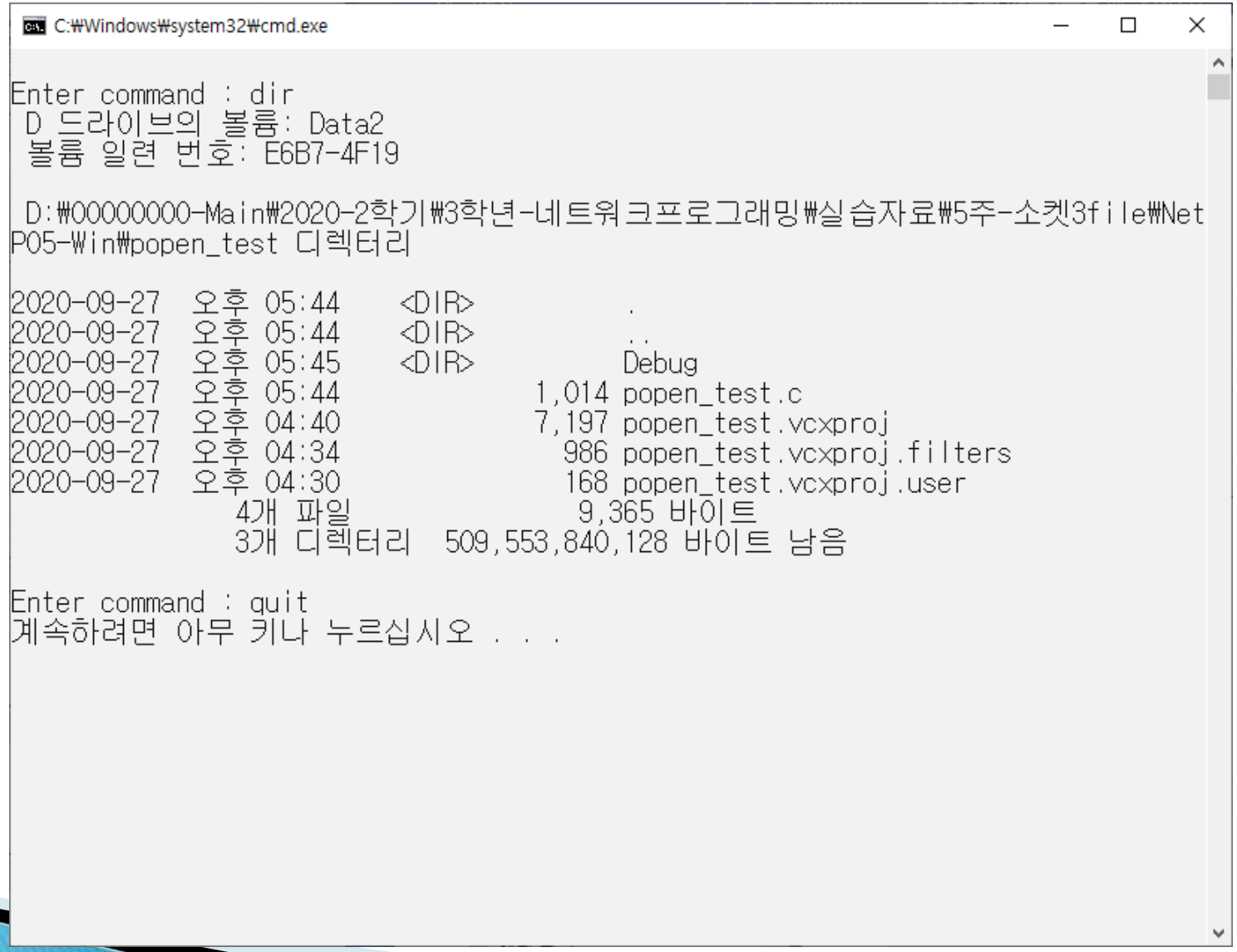
```c
// popen_test.c
// 명령어를 실행하고 결과를 파일로 읽는 예제
#include <stdio.h>
#include <stdlib.h>
void main()
{
    char str[129] = { 0 };
    FILE* fp;
    char buf[129] = { 0 };
    int n;

    while (1) {
        printf("\nEnter command : ");
        gets(str);
        if (strcmp(str, "quit") == 0)
            break;
        //system(str);
        // file_server4는 gets(str) 대신
        // recv(client_fd, buf, 128, 0); 을 사용하고
        // if (strncmp("dir", buf, 3)==0) 이면
        //  popen() 처리를 하고 결과를 보내면 된다.

#ifdef WIN32
        fp = _popen(str, "r");
#else // Linux 는 dir 대신 ls -l 사용
        if (strcmp("dir", str) == 0)
            strcpy(str, "ls -l");
        fp = popen(str, "r"); //
#endif
```

"r" 사용

```c
        while (1) {
            memset(buf, 0, 128);
            n = fread(buf, 1, 128, fp);
            if (n <= 0)
                break;
            //printf("n=%d\n", n);
            printf(buf); // 화면에 출력
            // file_server4는
            // 만약 ftp_client4에서 dir 명령어를 요청한다면
            // printf() 대신 send(client_fd, buf, 128, 0); 쓰면 된다.
            //
        }
#ifdef WIN32
        _pclose(fp);
#else
        pclose(fp);
#endif

    }
}
```

128로 고정

popen_test 실행 화면



ldir (local directory), !cmd (local 명령어) 구현

- file_client4>ldir
  - system("dir"); // Linux "ls -l"

- !cmd
  - local에서 cmd 실행한 결과 보여준다
  - 예:) file_client4>!netstat
    - cmd 창에서 netstat 실행 결과 보는 것과 같은 화면 보여준다.
    - system("netstat");
  - printf("file_clinet4>"); gets(cmdstr);
  - if (cmdstr[0]=='!')
    - system(cmdstr+1);
    - // !뒤의 명령어를 Local에서 실행시키고 화면에 보여준다.