

10주차_FTP(1)

데이터 네트워크연구실
이현호

lee075@cs-cnu.org

Assignment

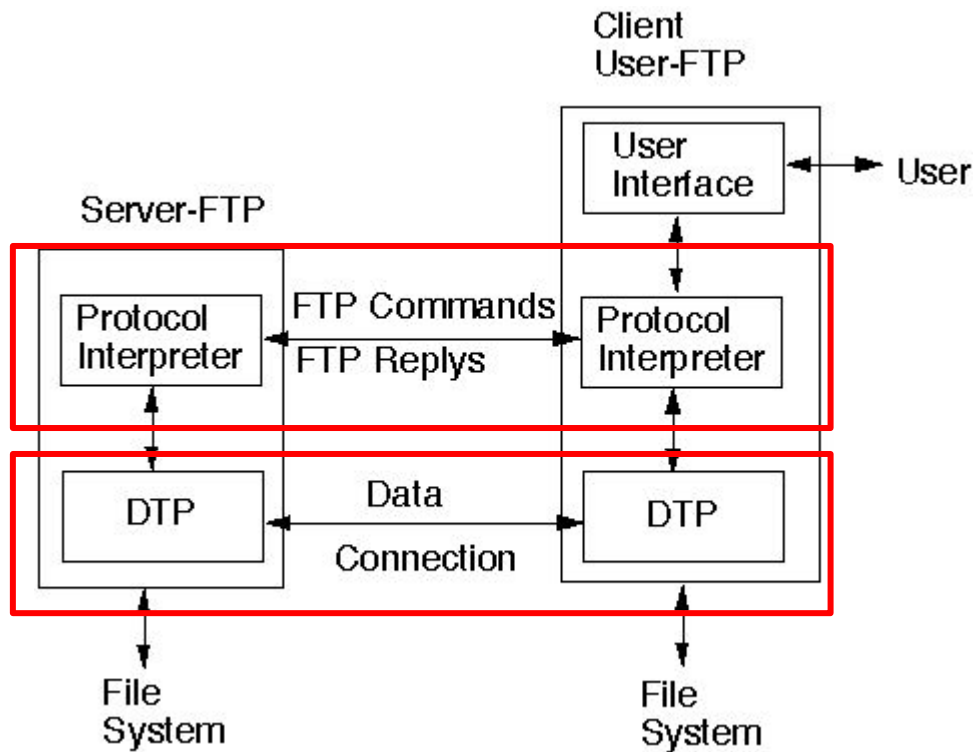
- FTP 프로그램 제작
- 구성 설명

FTP protocol

- HTTP와는 달리 연결의 종류는 2가지가 있다:
 - **명령 연결**: 먼저 제어 포트인 서버 21번 포트로 사용자 인증, 명령을 위한 연결이 만들어지고, 여기를 통해 클라이언트에서 지시하는 명령어가 전달된다.
 - **데이터 전송용 연결**: 실제의 파일 전송은 필요할 때 새로운 연결이 만들어진다.
 - **능동 모드**(포토 모드): 서버가 자신의 데이터 포트인 20번 포트에서부터 클라이언트가 지정한 지점으로의 데이터 연결을 만든다. 클라이언트가 지정하는 포트는 주로 1023 보다 큰 번호가 매겨진 포트이다.
 - **수동 모드**: 클라이언트가 서버가 지정한 서버 포트에 연결할 수 있게 한다. 이 때에는 보통 양쪽 포트 모두 1023 보다 큰 포트를 사용한다.

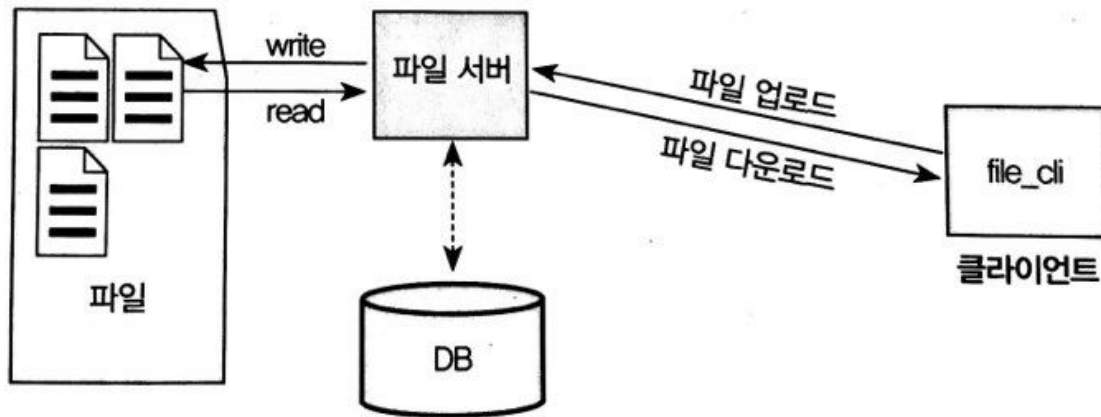
FTP protocol

- 명령을 하기 위한 연결 생성
- 파일을 전송하기 위한 다른 연결 생성



Structure

- 서버와 클라이언트 제작
- 서버의 기능
- 파일 업로드 / 다운로드
- 정보 저장



Server

Server

- 클라이언트의 요청에 따른 동작
- 클라이언트의 요청
 - 파일 업로드
 - 파일 다운로드
 - 파일 리스트
 - 도움말 출력

Process

1. 클라이언트가 접속하여 보내는 명령 메시지를 받음

```
memset(&addr, 0x00, sizeof(addr));  
addr.sin_family = AF_INET;  
addr.sin_addr.s_addr = htonl(INADDR_ANY);  
addr.sin_port = htons(PORTNUM);
```


Process

2. 클라이언트 접속 후 새로운 프로세스로 명령 처리

```
clilen = sizeof(addr);  
cli_sockfd = accept(sockfd, NULL, &clilen);  
if(cli_sockfd < 0)  
    exit(0);  
pid = fork();  
if(pid == 0)  
{  
    process(cli_sockfd);  
}
```

Process

3. 입력받은 명령을 처리

- 업로드 : 1
- 다운로드 : 2
- 리스트 조회 : 3

```
int process(int sockfd)
{
    char buff[MAXLINE];
    struct Cquery query;
    while(1)
    {
        if(recv() <= 0)
        {
            return -1;
        }
        query.command = ntohl(query.command);
        switch(query.command)
        {
            case (Q_UPLOAD):
                file_upload();
                break;
            case (Q_DOWNLOAD):
                file_download();
                break;
            case (Q_LIST):
                file_list();
                break;
        }
        break;
    }
    return 1;
}
```

Result

- 예시 결과 화면

```
crawler@DNLAB:~/test$ ./client -i 168.188.125.218 -u icmp.c
crawler@DNLAB:~/test$ ./client -i 168.188.125.218 -l
    name = icmp.c
    ip = 168.188.129.166
up_date = 2017/11/21 13:20:45
    count = 0
    size = 2100
=====
End!
```