NFS & FTP

Content

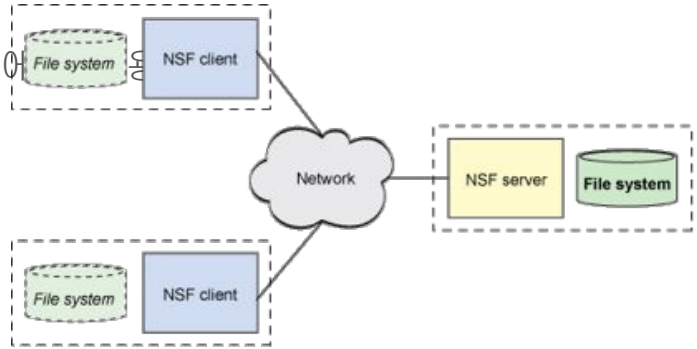
1. NFS
2. NFS 실습
3. FTP
4. Secure FTP
5. FTP 실습

NFS

NFS

NFS: Network File System

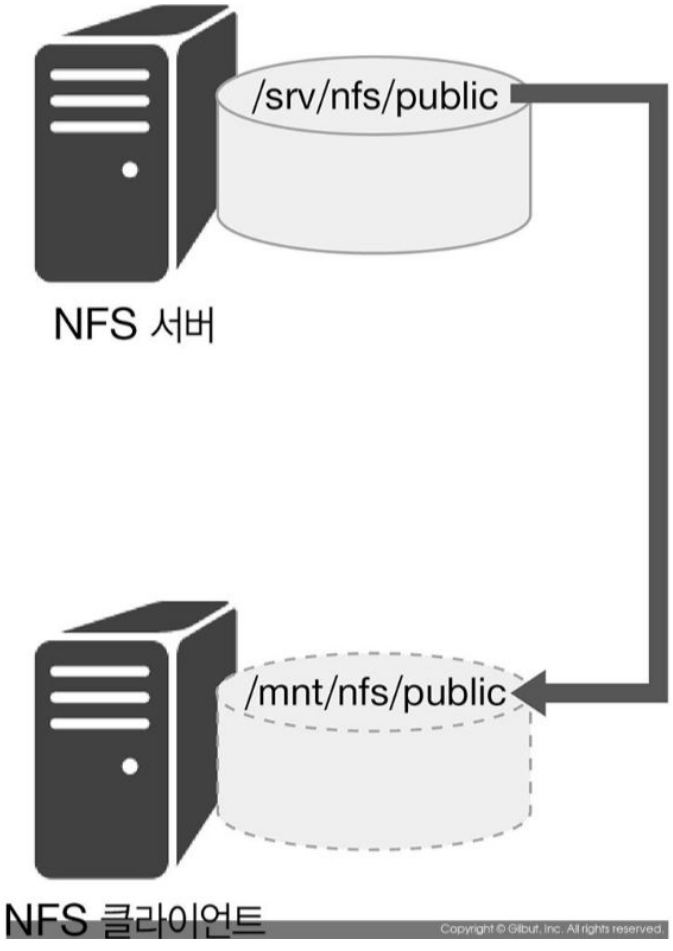
1. 공유된 원격 호스트의 파일을 로컬에서 사용할 수 있도록 개발된 파일 시스템
   a. 하드웨어, 운영체제, 네트워크 구조가 달라도 파일을 공유할 수 있도록 고안됨
   b. 스토리지가 고가였던 시절 디스크가 없는 시스템을 지원하기 위해 개발되었으나, 현재는 파일 공유, 파일 서버 용도로 사용
   c. 분산 서버 시스템에서 유용하게 사용할 수 있음
2. 서버/클라이언트 모델을 이용
   a. 클라이언트에서 네트워크를 통해 파일에 접근
   b. 서버에서 공유한 파일, 디렉토리를 클라이언트가 마운트



NFS

NFS: Network File Sys

1. 공유된 원격 호스트]도록 개발된 파일 시스템
2. 서버/클라이언트 모
   a. 클라이언트에서 네
   b. 서버에서 공유한 피 서 사용



NFS

NFS: Network File System

1. 공유된 원격 호스트의 파일을 로컬에서 사용할 수 있도록 개발된 파일 시스템
2. 서버/클라이언트 모델을 이용
   a. 클라이언트에서 네트워크를 통해 파일에 접근
   b. 서버에서 공유한 파일, 디렉토리를 클라이언트가 마운트에서 사용
3. RPC(Remote Procedure Call)를 이용
   a. TCP/IP 통신 사용
   b. FTP ← 뒤에서 배울거임

NFS

RPC: Remote Procedure Call

1. 한글말로 “원격 프로시저 호출”
2. 별도의 원격 제어를 위한 코딩 없이 다른 주소 공간에서 함수나 프로시저를 실행할 수 있게하는 프로세스 간 통신 기술 (by. wikipedia)

NFS

RPC: Remote Procedure Call

1. 한글말로 “원격 프로
2. 별도의 원격 제어를
   실행할 수 있게하는

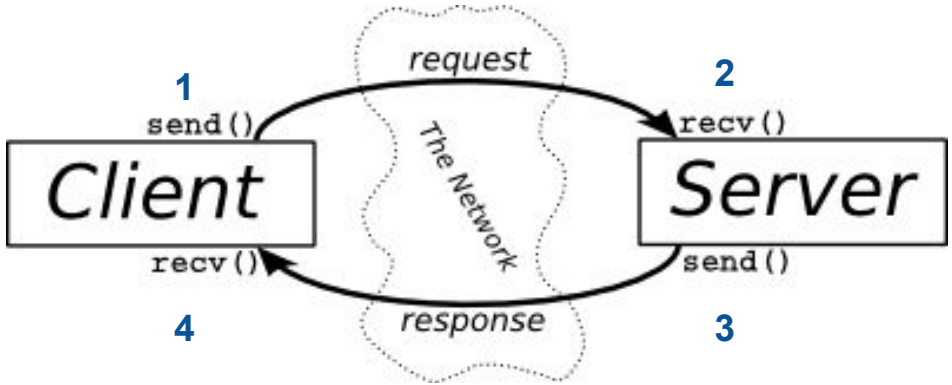
서 함수나 프로시저를

edia)

NFS

Client-Server 패턴 (RPC 이전에…)

1. Client가 Server에게 요청후 기다리면
2. Server에서 Client의 요청을 받아 명령을 실행하고
3. Server가 Client에게 결과값을 반환해주고
4. 기다리던 Client는 결과값은 받아서 작업 진행



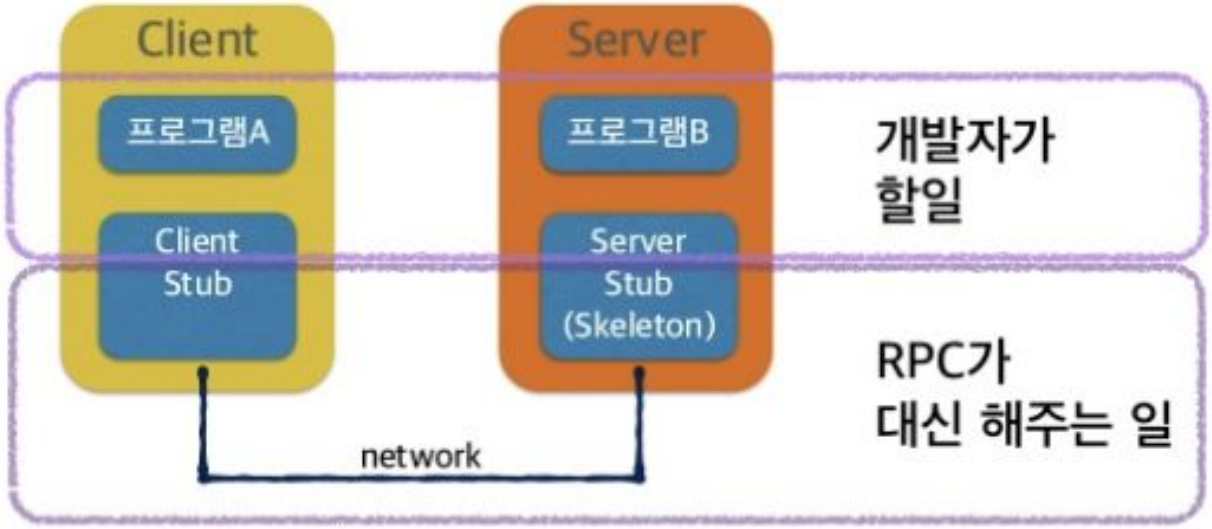
NFS

Client-Server 패턴

1. Socket으로 구현할 수 있겠네요! -> 가능합니다!
2. 문제는
   a. 네트워크는 항상 빠르고 장애가 없어야 됨
   b. Server는 Client가 요청하면 언제든 즉시 응답해줘야 함
   c. Client도 언제든 Server응답을 바로 받아 반응해야함
3. 2번을 위해서 발생가능한 예외상황들을 모두 예측해서 대비(구현)해주어야만 합니다 :(

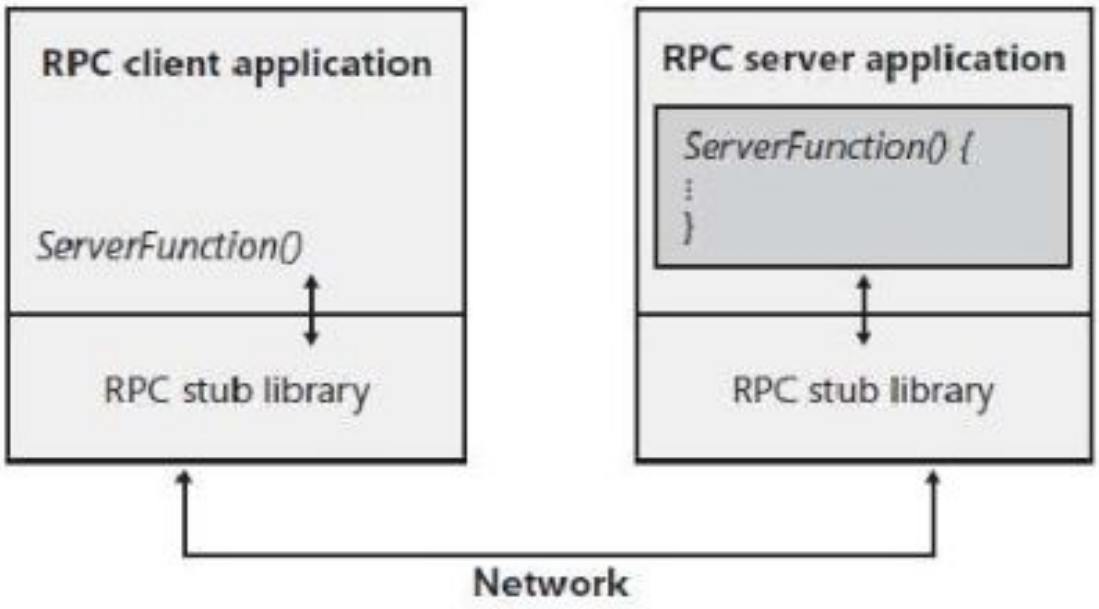
NFS

RPC: Remote Procedure Call 로 돌아와서....



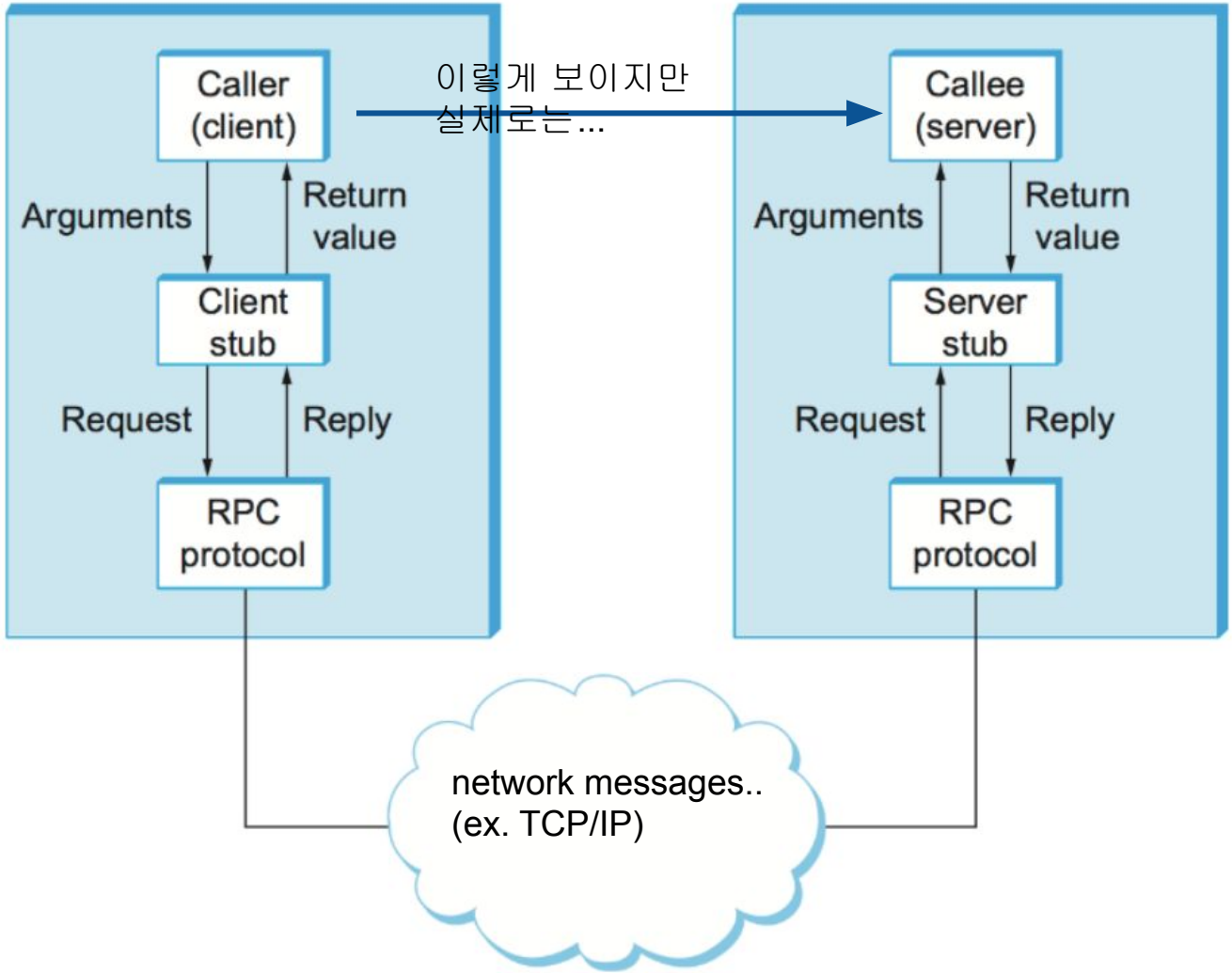
NFS

RPC: Remote Procedure Call 로 돌아와서....



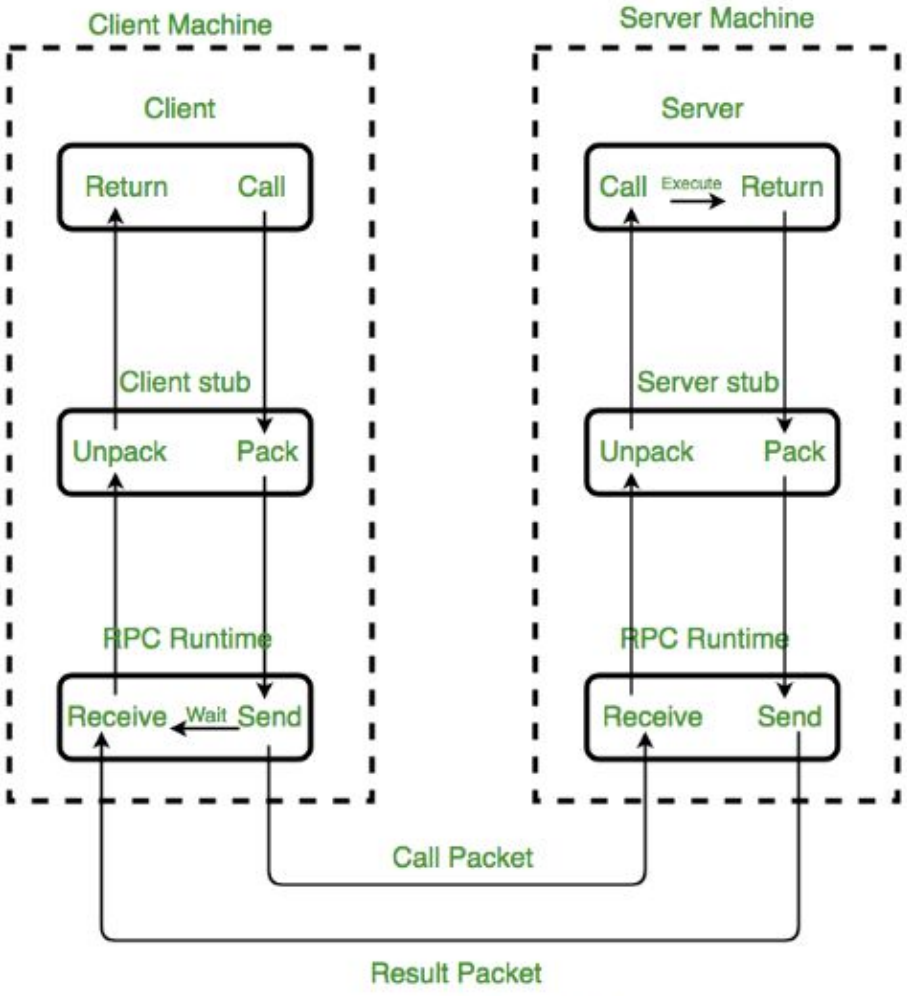
NFS

RPC: Re


Caller
(client)
Arguments
Return
value
Client
stub
Request
Reply
RPC
protocol
이렇게 보이지만
실제로는...
Callee
(server)
Arguments
Return
value
Server
stub
Request
Reply
RPC
protocol
network messages..
(ex. TCP/IP)

NFS

RPC: Remote Proc



Implementation of RPC mechanism

NFS

RPC: Remote Procedure Call 로 돌아와서....

1. network 통신과 관련된 작업은 RPC에서 알아서 해줌
   a. Remote interaction에 대한 세부 구현은 신경 쓸 필요가 없어짐
2. 한 프로그램이 다른 메모리 공간의 procedure를 실행하는 것
3. 주로 네트워크를 통해 다른 컴퓨터에 내가 원하는 명령을 보낼 때 쓰입니다
   a. ONC RPC (Open Network Computing RPC)
4. Remote를 local처럼 쓸 수 있도록 도와주는 미들웨어라고 보면 됩니다

NFS

RPC: Remote Procedure Call

1. 한글말로 “원격 프로시저 호출”
2. 별도의 원격 제어를 위한 코딩 없이 다른 주소 공간에서 함수나 프로시저를 실행할 수 있게하는 프로세스 간 통신 기술 (by. wikipedia)

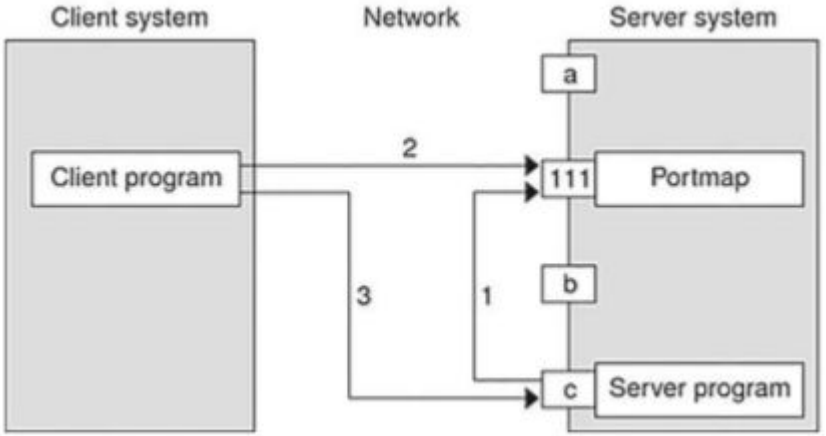
이해가 되시나요???? :)

NFS

ONC RPC (Open Network Computing RPC)

1. 네트워크를 통한 RPC
2. NFS는 ONC RPC에서 '파일' 정보를 왔다갔다 할 수 있는 시스템입니다
3. Portmap를 통해 RPC 서비스에 접근
    a. 포트요청이 있을 때 새로운 포트를 할당해주고, client-server mapping을 해줌
    b. portmapper라는 프로그램을 이용해서 네트워크 포트를 할당 받음

NFS

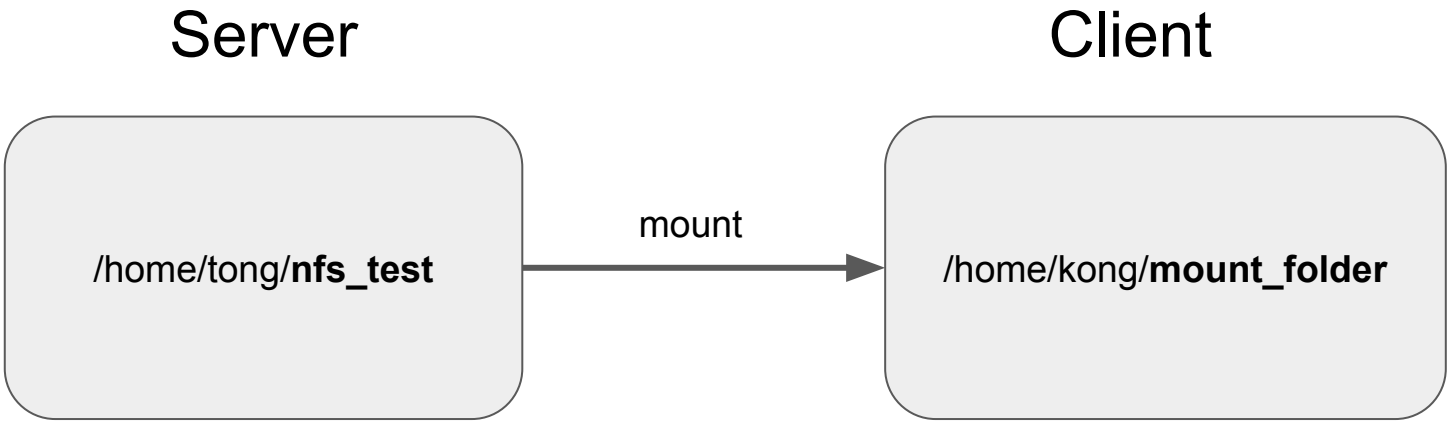


Portmap (portmapper)

1. RPC 연결에 관여하는 데몬
   a. nfs server와 nfs client를 이어주는 다리역할
2. 111번 포트 사용
3. **nfs 서비스를 요구하는 접속이 들어오면 포트번호 바꾸어 접속을 연결시켜줌**
   a. 동적 포트 할당
4. 5.x까지는 portmap 패키지 별도 설치, 6.x 이후 rpcbind 패키지 사용

NFS 실습

NFS 실습


Server
Client
/home/tong/**nfs_test**
mount
/home/kong/**mount_folder**

NFS 실습

NFS 서버

$ sudo apt-get install nfs-common nfs-kernel-server portmap

// nfs에 사용한 디렉토리 만들고 권한주기
$ mkdir nfs_test
$ chmod 777 nfs_test

NFS 실습

NFS 서버

// nfs 환경설정 : **/etc/exports**

$ vim /etc/exports

// [공유할 디렉토리] [공유할 client](options)

>> (ex) /home/ubuntu/nfs_test [client ip](rw,sync,no_subtree_check)

>> [client ip]에 *사용가능 -> 모든 client ip로부터의 파일 접근을 허용하겠습니다!

NFS 실습

NFS 서버 : NFS 옵션 설정하기

옵션	뜻
ro(Read Only)	읽기 전용
rw(Read and Write)	읽기+쓰기
sync	Client가 파일 변경시 즉시 동기화
no_subtree_check	하부 구조 검사를 사용하지 않음(전송률 높아짐)

* 하부구조검사: 파일 시스템의 전체 디렉토리가 아닌 하부 디렉토리가 내보내진 경우, 내보낸 하부 디렉토리에 요청된 파일이 존재하는지 검사하는 과정

NFS 실습

NFS 서버

// NFS 서비스 시작하기

$ sudo service nfs-kernel-server restart

$ sudo service portmap restart

NFS 실습

NFS 서버

// RPC 작동 확인

$ sudo rpcinfo -p // -p option : 간단버전으로 port와 연결된 서비스들 목록 보기 가능

NFS 실습

NFS 클라이언트

// NFS install

$ sudo apt-get install nfs-common

// mount할 폴더 만들기

$ mkdir mount_foler

// mount 하기

$ sudo mount -t nfs [server ip]:/home/ubuntu/nfs_test /home/ubuntu/mount_folder

NFS 실습

NFS 클라이언트

// umount 하기 : 마운트를 해제

$ sudo unmount nfs

// server에서 공유하고 있는 (마운트 된) 디렉토리 목록 확인하기

$ sudo showmount -e [server ip]

FTP

FTP

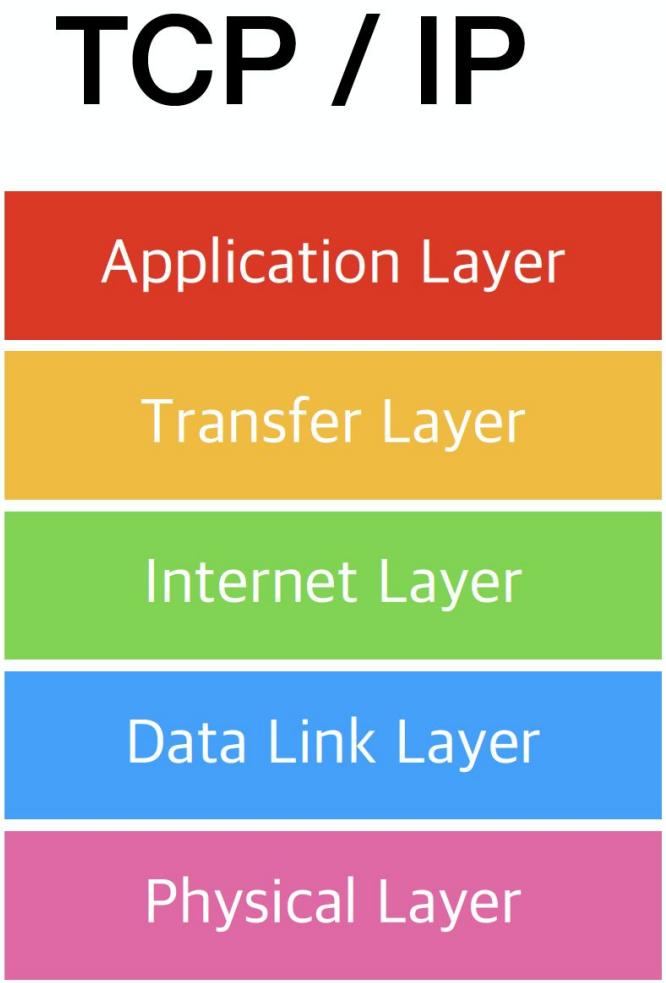
FTP: File Transfer Protocol

1. “파일 전송"을 위한 프로토콜!

FTP

FTP: File Transfer Protocol

FTP는 어느 layer일까?



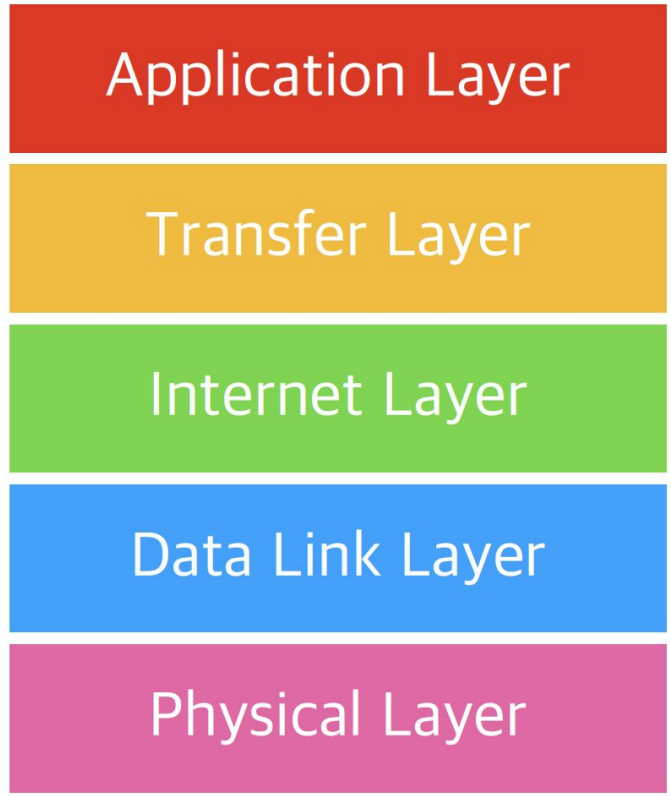
FTP

FTP: File Transfer Protocol

FTP, HTTP, DNS...

“FTP는 TCP를 이용하는

application layer의 프로토콜입니다.”



FTP

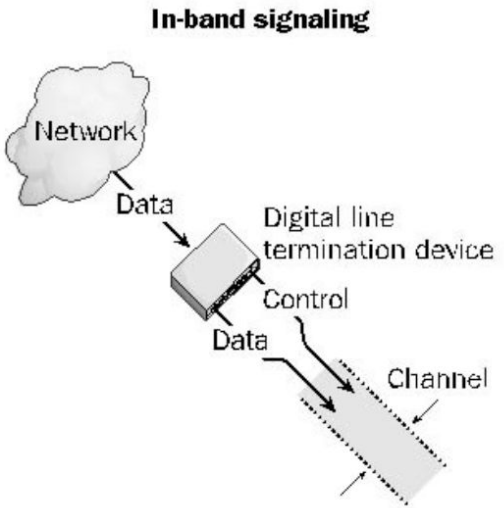
FTP vs. HTTP

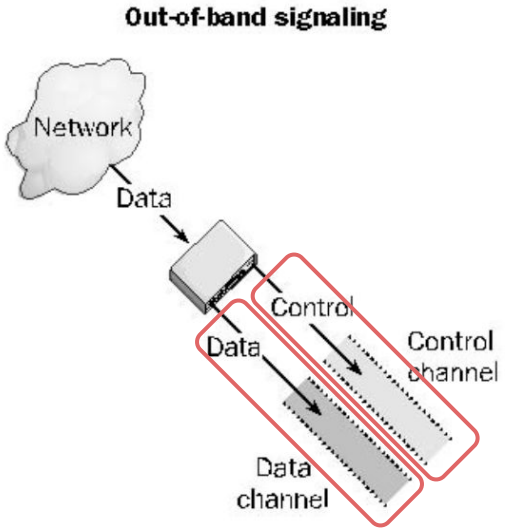
1. FTP가 HTTP보다 먼저 만들어졌음
2. HTTP에 비해 비용(overhead)가 적음
   a. 파일 전송만을 목적으로 하기 때문에 컨트롤에 필요한 데이터가 적다
   b. **Out-of-band** -> 데이터 전송시 아주 짧은 헤더, 혹은 헤더 없이도 전송 가능
3. 이어받기 기능 지원
   a. 전송 도중 인터넷 연결이 끊어져도, 전송되던 중간부터 다시 이어서 받을 수 있음
   b. REST [바이트 수] : 특정 바이트 수 지점부터 다시 파일 전송 시작 명령어

FTP

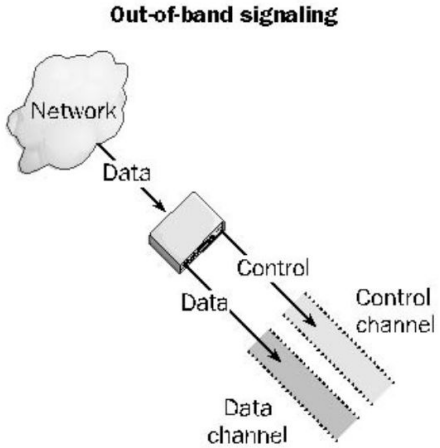
Out-of-band ?

1. 파일을 전송하는 data connection과 제어를 위한 control connection이 다른 것
   a. In-band : 동일 대역/채널/포트/연결 상에서 전송한다는 뜻





FTP



FTP connection

1. Control connection
   a. Client, Server사이에 서비스 요청 및 결과 통보를 알리기 위한 명령
   b. ex. 사용자 계정, 암호등의 정보, 파일전송 명령, 결과값 등등…
   c. ftp 연결 시작부터 끝까지 계속 연결되어 있음
   d. default : 21번 포트

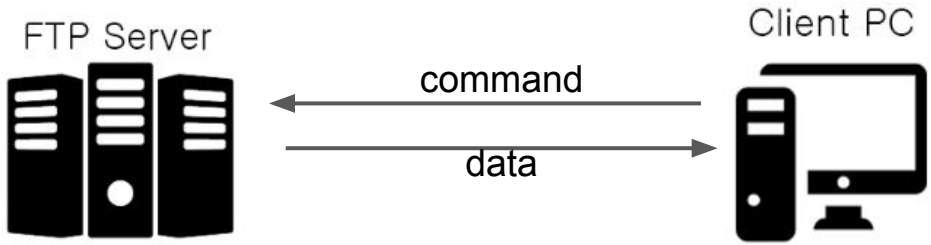
2. Data connection
   a. 실제 파일 데이터를 전송하기 위한 포트
   b. 데이터를 전송할 때마다 새로 연결됨 (하나의 파일 전송이 끝나면 data connection이 close됨)
   c. 연결 방식: Active mode, Passive mode
   d. default : 20번 포트

FTP

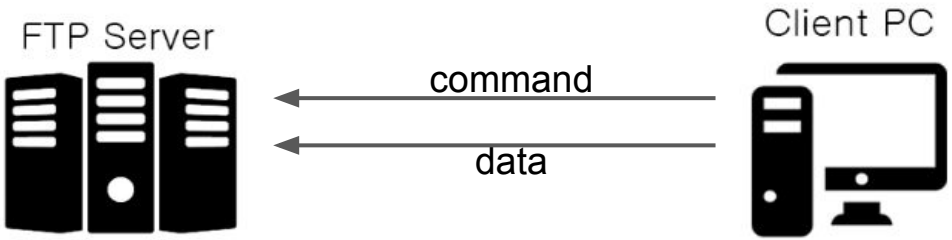
FTP connection: Active mode & Passive mode

:data connection 요청 방향의 차이:

1. Active Mode



2. Passive Mode



FTP

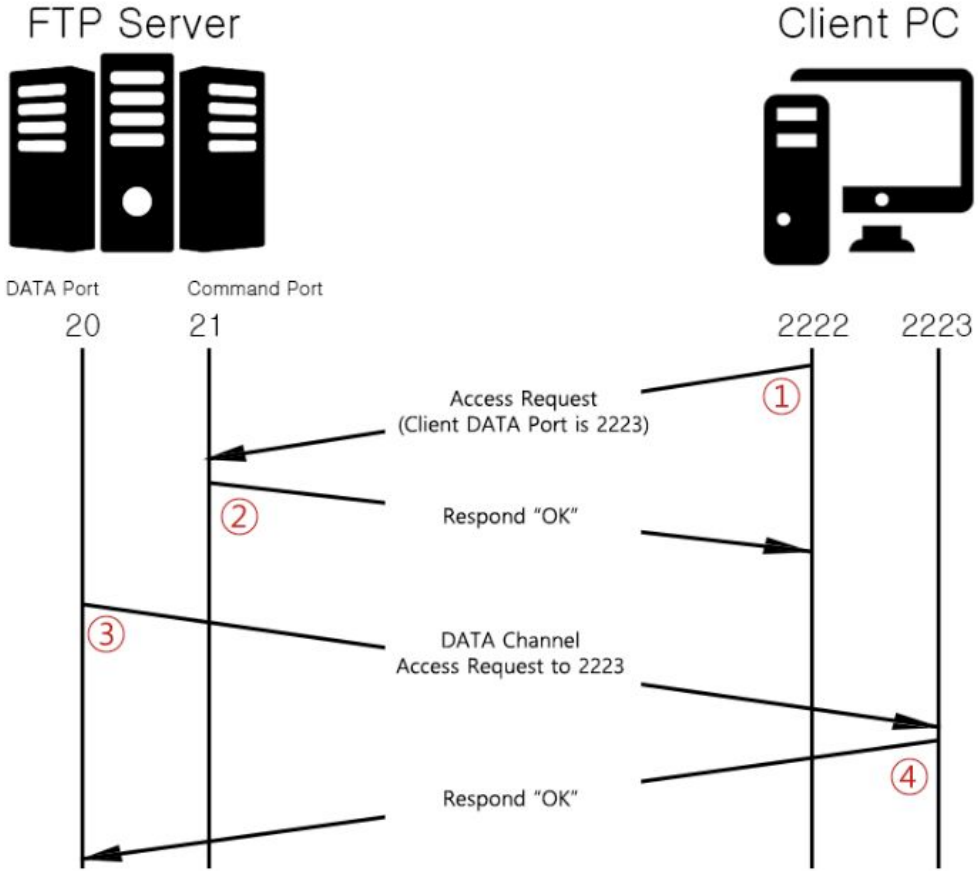
FTP connection: Active mode

1. FTP Client에서 Server 21번 포트로 인증 요청 + (Client 자신의 data channel을 위한 포트번호 정보를 패킷에 포함하여 전송)

2. Server에서 Client에 OK 응답을 전송

3. Server에서 생성한 20번 포트에서 -> Client 2223포트로 data channel 연결 요청

4. Client가 Server에 OK응답을 보내고, 데이터 채널 연결 완료

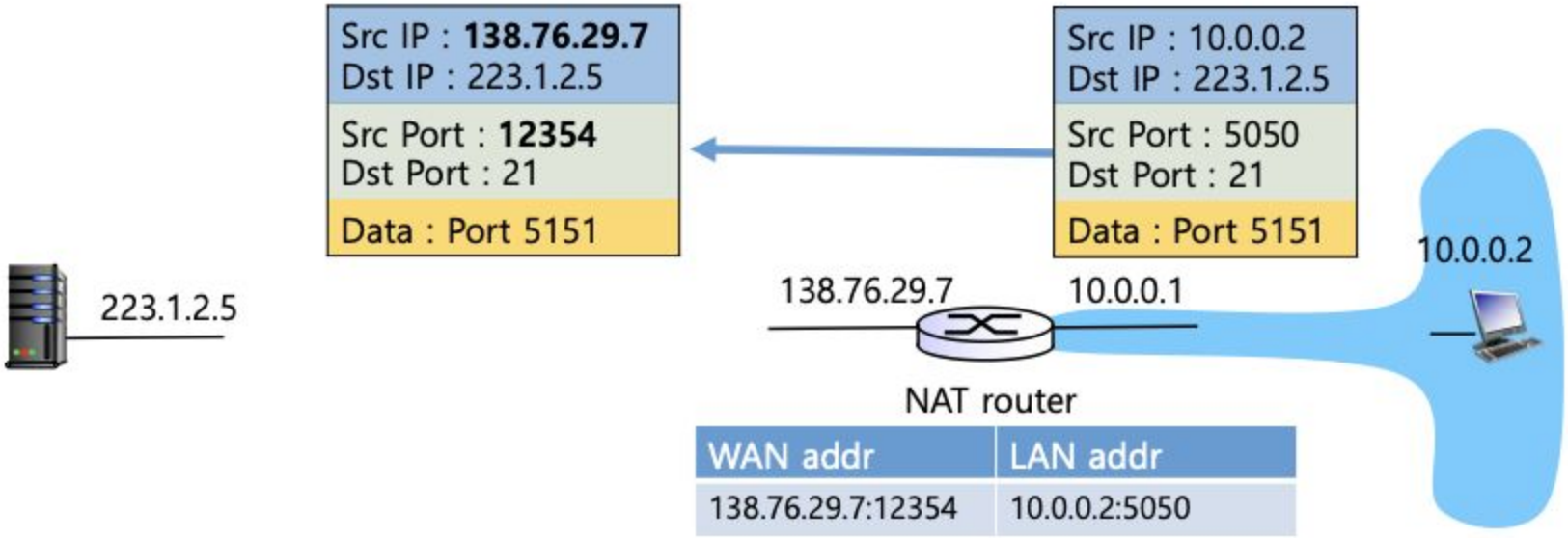


FTP

FTP connection: Active mode

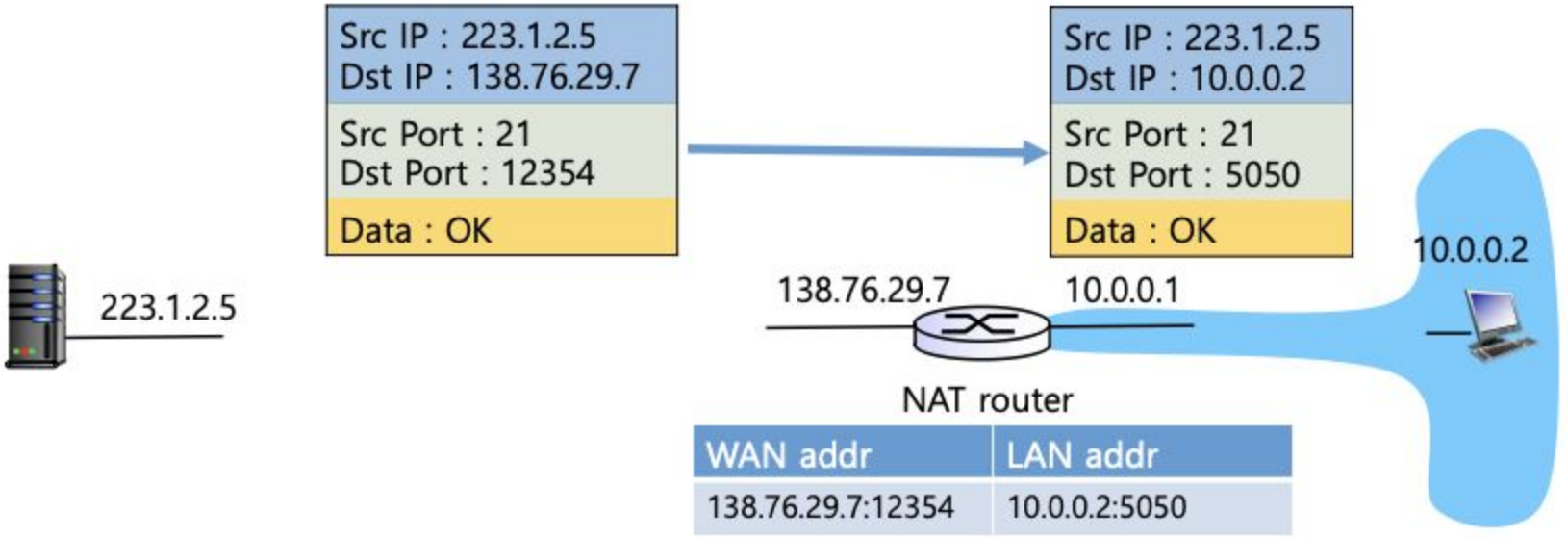
1. Data channel 요청: Server -> Client
2. Client의 방화벽에 20번 포트가 차단되어 있으면 데이터 채널 연결 불가능
3. 사실상 Client/Server 양쪽 모두 방화벽이 꺼져있어야 가능
4. NAT(공유기)에서 방화벽 문제로 작동이 안됨

Active mode in NAT



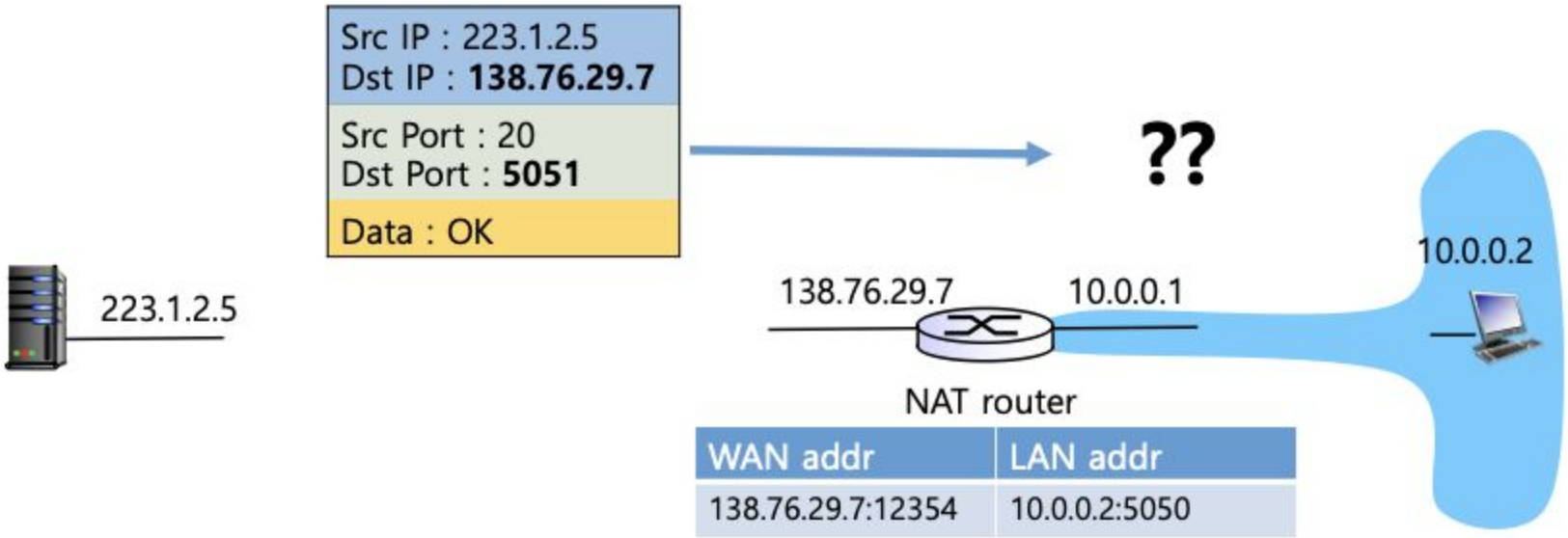
출처: 2017 NFS/FTP wheel seminar ppt

Active mode in NAT



출처: 2017 NFS/FTP wheel seminar ppt

Active mode in NAT



Active mode가 NAT에서 정상 작동하지 않는다.

출처: 2017 NFS/FTP wheel seminar ppt

FTP

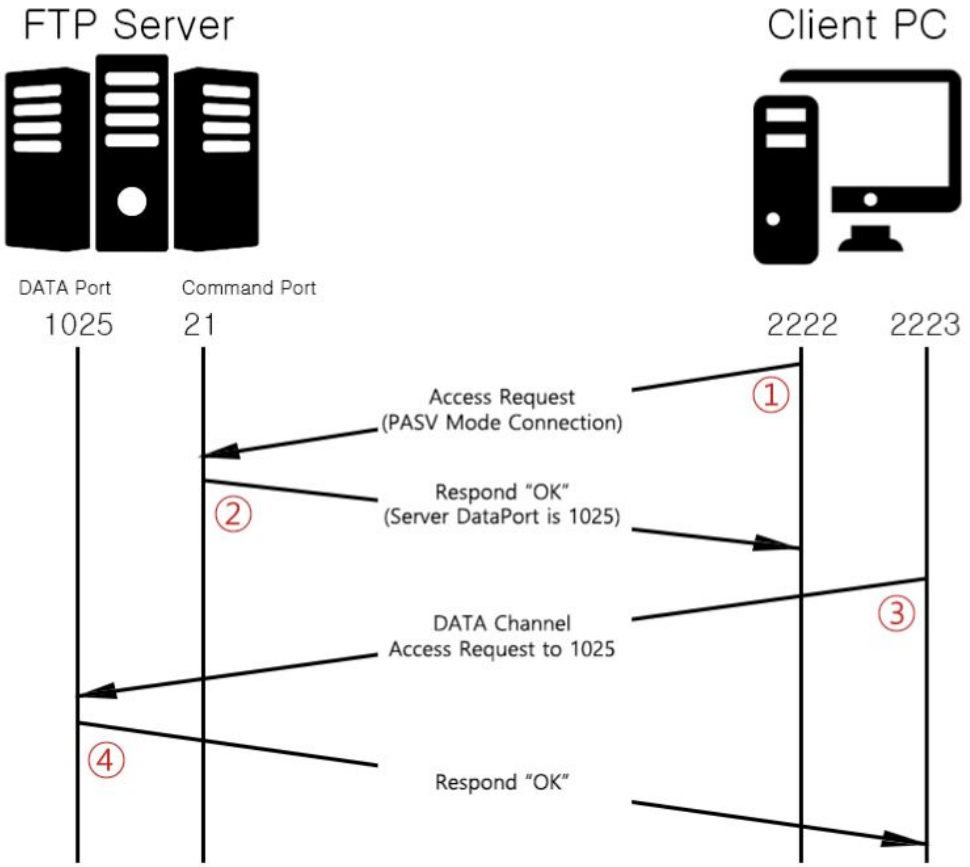
FTP connection: Passive mode

1. FTP Client에서 Server 21번 포트로 인증 요청 + (passive mode로 연결할 것임을 알림)

2. Server에서 Client에 OK 응답을 전송 + **(Server 자신의 data channel 포트번호 정보를 함께 전송)**

3. **Client에서 생성한 2223포트에서 -> Server 데이터포트로 연결 요청**

4. Server가 Client에 OK응답을 보내고, 데이터 채널 연결 완료



FTP

FTP connection: Passive mode

1. Data connection 요청: Client -> Server
2. Active mode의 문제(NAT, 방화벽)을 막을 수 있음
3. Server의 방화벽에 Client의 데이터포트가 차단되어 있으면 데이터 채널 연결 불가능
   a. 서버의 모든 포트 방화벽을 열어둬야 함 (1024 ~ 65535)
4. ftp 데몬 : 포트 개수 범위를 제한 할 수 있음 -> 지정한 범위의 포트들만 서버가 허용해주면 되기 때문에 어느정도 해결이 가능

FTP

Problem of FTP

1. Client
    a. Username
    b. Password

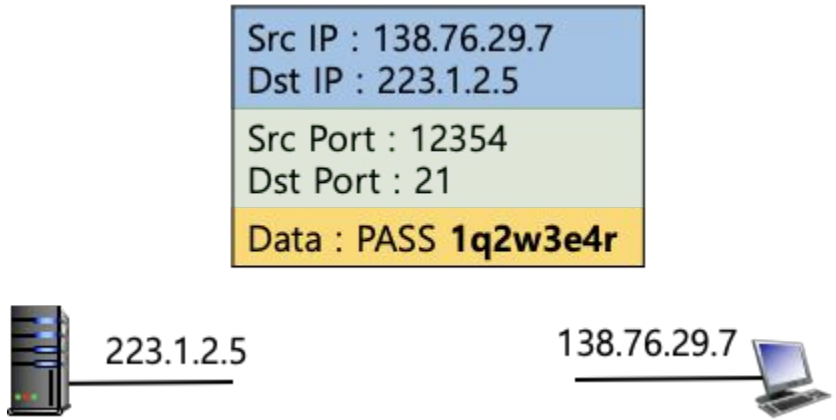
(+) anonymous FTP

- 익명으로 접근가능한 FTP도 존재.
- 공식적인 사용자 계정 및 암호 입력이 필요없는 공개형 FTP Server
- 하나의 공용 계정을 모두가 사용 -> ‘Anonymous’라는 계정으로 접속 가능

FTP

Problem of FTP

1. Client
   a. Username
   b. Password -> FTP는 비밀번호를 plain text으로 전송합니다….



FTP

Problem of FTP

1. Client
   a. Username
   b. Password -> FTP

전송합니다....

Secure FTP

Secure FTP

FTPS

1. FTP Secure, FTP-SSL
2. TLS/SSL 암호화 과정을 거침
    a. https와 동일한 과정

Secure FTP

FTPS: SSL/TLS example

SSL Client:

- **Client Public Key**
- Client Private Key
- Server Private Key

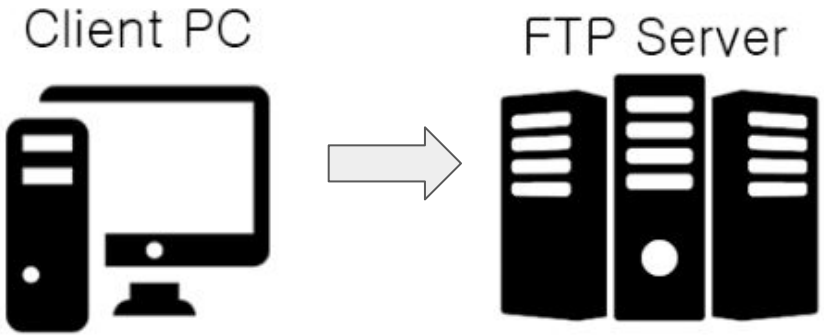
SSL Server:

- Client Public Key
- Server Private Key

Secure FTP

FTPS: SSL/TLS example

SSL Client:
- Client Public Key
- Client Private Key
- Server Private Key



SSL Server:
- **Client Public Key**
- Server Private Key

Secure FTP

FTPS: SSL/TLS example

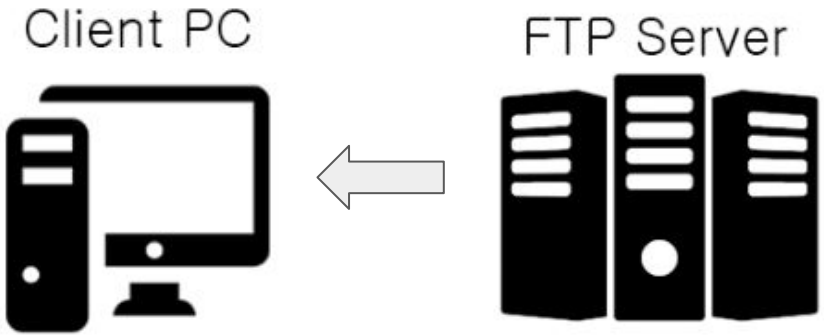
SSL Client:
- Client Public Key
- Client Private Key
- Server Private Key

SSL Server:
- Client Public Key
- **Server Private Key**

Secure FTP

FTPS: SSL/TLS example

SSL Client:
- Client Public Key
- Client Private Key
- Server Private Key

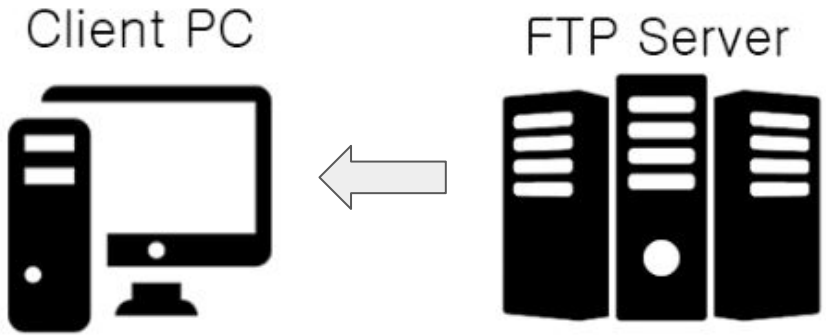


SSL Server:
- Client Public Key
- **Server Private Key**

Secure FTP

FTPS: SSL/TLS example

SSL Client:
- Client Public Key
- Client Private Key
- **Server Private Key**



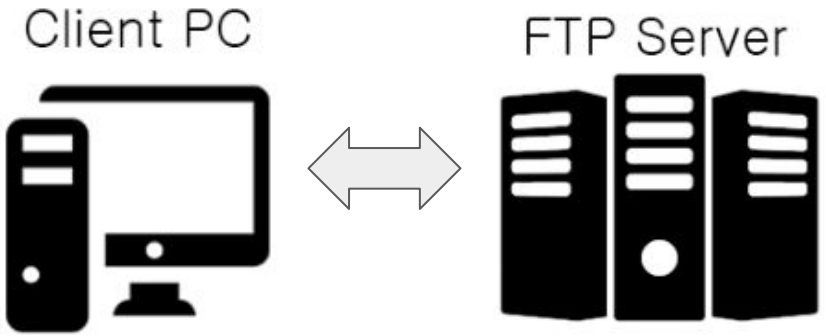
SSL Server:
- Client Public Key
- Server Private Key

Secure FTP

FTPS: SSL/TLS example

SSL Client:

- Client Public Key
- Client Private Key
- **Server Private Key**



SSL Server:

- Client Public Key
- **Server Private Key**

공유 비밀 키 (server private key)를 사용하여 암호화된 정보 주고받기 가능

Secure FTP

FTPS

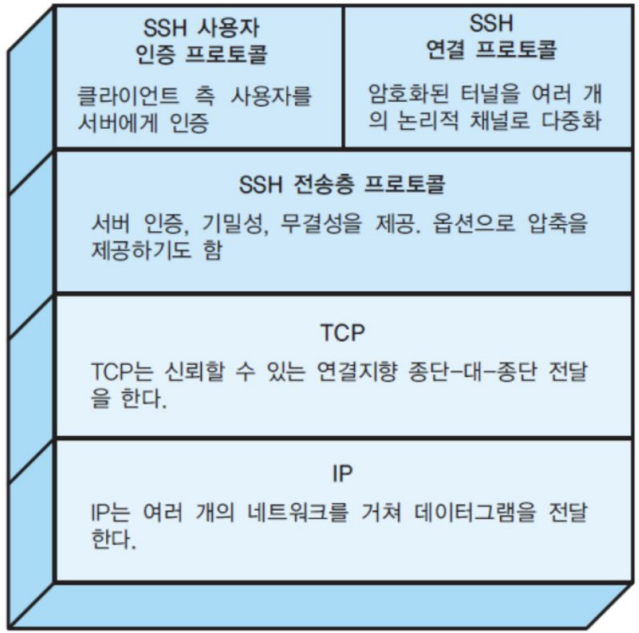
1. 방화벽에서 막힘
   a. 대부분의 방화벽은 client-server 소통 전 미리 데이터 커넥션을 새로 여는지 확인을 하고 열 준비를 함
   b. 새로 열릴 data channel 포트 번호 정보가 암호화되어 전송되면,
   c. 방화벽이 패킷 내부 내용을 제대로 알지 못하기 때문에 결국 새로운 포트를 열지 못하게 막음
2. https://en.wikipedia.org/wiki/FTPS : Firewall imcompatibility 참조

Secure FTP

SFTP

1. SSH FTP, Secure FTP
2. SSH(Secure shell)으로 FTP의 기능을 구현한 것
   a. (ftp와 별개)
   b. default port : 22번
   c. 커넥션이 1개 -> 방화벽 문제 해결

3. sparcs 서버와 연결할때는 sftp를 이용합시다 :)



SSH 프로토콜 스택

FTP Server daemon

FTP Server daemon

1. Wu-ftpd
   a. 2000년대 초까지 많이 사용됐던 프로그램
   b. 보안문제 취약
2. ProFTPD
   a. 널리 사용되고 있는 ftp 서버 프로그램
   b. 누구나 자유롭게 수정, 배포, 사용 가능
3. vsftpd
   a. Very Secure FTP Daemon
   b. IPv6, FTPS 지원

FTP 실습

vsftpd

FTP 실습: vsftpd

vsftpd

$ sudo apt-get install vsftpd

// FTP 환경설정 : **/etc/vsftpd.conf**

$ sudo vim /etc/vsftpd.conf

```
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
#
# Run standalone?  vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
#write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
#local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
```

FTP 실습: vsftpd

vsftpd options

1. listen
   a. YES -> standalone
   b. NO -> sinetd
2. anonymous_enable
   a. allow anonymous FTP
   b. Anonymous vs. Full service ftp(user, pass 필요)
3. local_enable
   a. 로컬 사용자 접속 허용
4. write_enable
   a. 쓰기 허용
5. pasv_enable
   a. Passive mode 사용

FTP 실습: vsftpd

vsftpd options

1. listen
   a. **YES -> standalone**
   b. **NO -> sinetd**
2. anonymous_enable
   a. allow anonymous FTP
   b. Anonymous vs. Full service ftp(user, pass 필요)
3. local_enable
   a. 로컬 사용자 접속 허용
4. write_enable
   a. 쓰기 허용
5. pasv_enable
   a. Passive mode 사용

FTP 실습: vsftpd

vsftpd options: Standalone vs. Xinetd

1. Standalone : 데몬이 독립적으로 실행됨 (항상 실행되있는 데몬)
2. Xinetd : 요청이 있을때만 실행됨

Standalone(독립형)	Xinetd(Extended Internet Services Daemon)
항상 준비되어 있는 데몬	요청이 들어오면 준비하는 데몬
/etc/init.d	/etc/exinetd.d
메모리 부하	메모리 효율
응답속도 빠름	응답속도 느림
Sendmail, apache, mysql, 메일서버, 웹서버	TELNET(원격접속, PuTTY같은 것)

FTP 실습: vsftpd

vsftpd commands

$ sudo service vsftpd start

$ sudo service vsftpd stop

$ sudo service vsftpd restart

$ sudo service vsftpd status

FTP 실습: vsftpd

FTP Client?

1. Filezilla, 알FTP 등 사용
2. Console : **sftp**

$ sftp [ftp_server ip] ---> 콘솔 창 열림

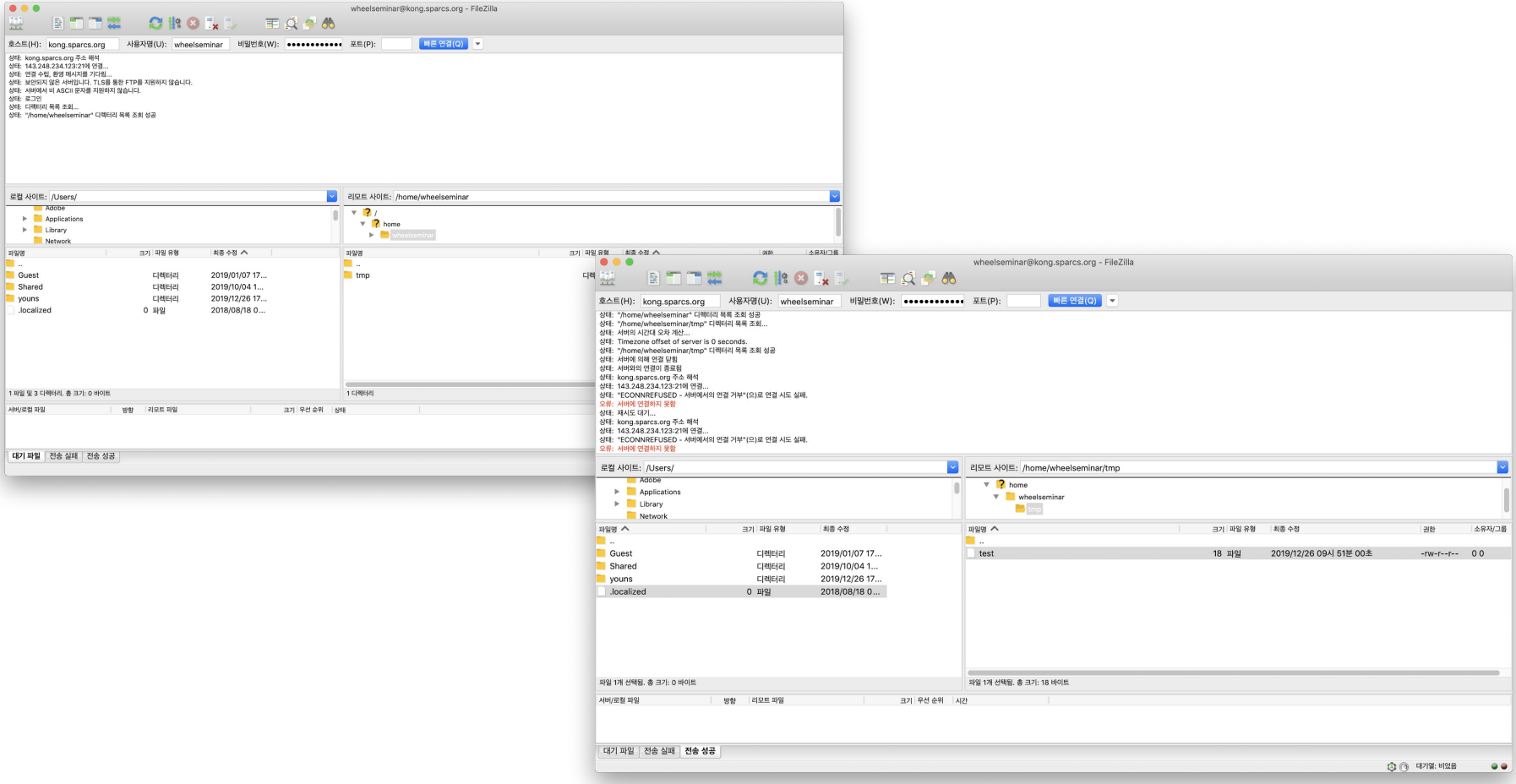
>> 기본 명령이 (ls, cd, pwd, mkdir, chmod) 리눅스와 동일
>> get / delete / put [file] : [file]을 가져오기/지우기/올리기
>> help : 도움말

FTP 실습: vsftpd

FTP Client : Filezilla

1. FTP 연결할 Server의 21번 포트가 열려 있어야 접근이 가능

$ (in FTP Server)sudo service vsftpd start


wheelseminar@kong.sparcs.org - FileZilla
호스트(H): kong.sparcs.org
사용자명(U): wheelseminar
비밀번호(W):
포트(P):
빠른 연결(Q)
상태: kong.sparcs.org 주소 해석
상태: 143.248.234.123:21에 연결...
상태: 연결 수립, 환영 메시지를 기다림...
상태: 보안되지 않은 서버입니다. TLS를 통한 FTP를 지원하지 않습니다.
상태: 서버에서 비 ASCII 문자를 지원하지 않습니다.
상태: 로그인
상태: 디렉터리 목록 조회...
상태: "/home/wheelseminar" 디렉터리 목록 조회 성공
로컬 사이트: /Users/
Adobe
Applications
Library
Network
파일명
크기
파일 유형
최종 수정
..
Guest
디렉터리
2019/01/07 17...
Shared
디렉터리
2019/10/04 1...
youns
디렉터리
2019/12/26 17...
.localized
0 파일
2018/08/18 0...
1 파일 및 3 디렉터리. 총 크기: 0 바이트
리모트 사이트: /home/wheelseminar
/
home
wheelseminar
tmp
1 디렉터리
서버/로컬 파일
방향
리모트 파일
크기
우선 순위
상태
대기 파일
전송 실패
전송 성공
wheelseminar@kong.sparcs.org - FileZilla
호스트(H): kong.sparcs.org
사용자명(U): wheelseminar
비밀번호(W):
포트(P):
빠른 연결(Q)
상태: "/home/wheelseminar" 디렉터리 목록 조회 성공
상태: "/home/wheelseminar/tmp" 디렉터리 목록 조회...
상태: 서버의 시간대 오차 계산...
상태: Timezone offset of server is 0 seconds.
상태: "/home/wheelseminar/tmp" 디렉터리 목록 조회 성공
상태: 서버에 의해 연결 닫힘
상태: 서버와의 연결이 종료됨
상태: kong.sparcs.org 주소 해석
상태: 143.248.234.123:21에 연결...
상태: "ECONNREFUSED - 서버에서의 연결 거부"(으)로 연결 시도 실패.
오류: 서버에 연결하지 못함
상태: 재시도 대기...
상태: kong.sparcs.org 주소 해석
상태: 143.248.234.123:21에 연결...
상태: "ECONNREFUSED - 서버에서의 연결 거부"(으)로 연결 시도 실패.
오류: 서버에 연결하지 못함
로컬 사이트: /Users/
Adobe
Applications
Library
Network
파일명
크기
파일 유형
최종 수정
..
Guest
디렉터리
2019/01/07 17...
Shared
디렉터리
2019/10/04 1...
youns
디렉터리
2019/12/26 17...
.localized
0 파일
2018/08/18 0...
파일 1개 선택됨. 총 크기: 0 바이트
리모트 사이트: /home/wheelseminar/tmp
home
wheelseminar
tmp
파일명
크기
파일 유형
최종 수정
권한
소유자/그룹
..
test
18 파일
2019/12/26 09시 51분 00초
-rw-r--r--
0 0
파일 1개 선택됨. 총 크기: 18 바이트
서버/로컬 파일
방향
리모트 파일
크기
우선 순위
시간
대기 파일
전송 실패
전송 성공
대기열: 비었음

FTP 실습: vsftpd

FTP Client : console (sftp)

허허허… permission 얼른 고치겠습니다 …….허허허….

```
/.credentials sftp wheelseminar@kong.sparcs.org
wheelseminar@kong.sparcs.org's password:
Connected to wheelseminar@kong.sparcs.org.
sftp> cd tmp
sftp> get test
Fetching /home/wheelseminar/tmp/test to test
Couldn't open local file "test" for writing: Permission denied
sftp>
```