SOFTWARE ARCHITECTURE
11. DISTRIBUTED FILE SYSTEM

Tatsuya Hagino
hagino@sfc.keio.ac.jp

lecture URL

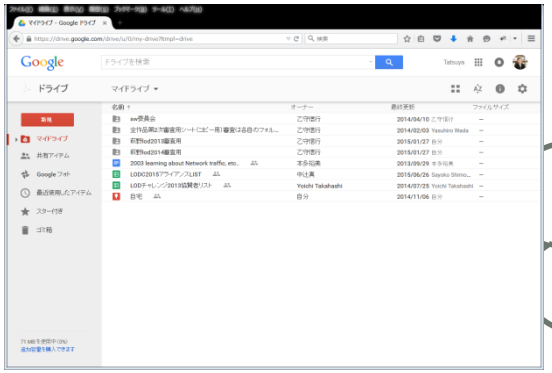
https://vu5.sfc.keio.ac.jp/slide/

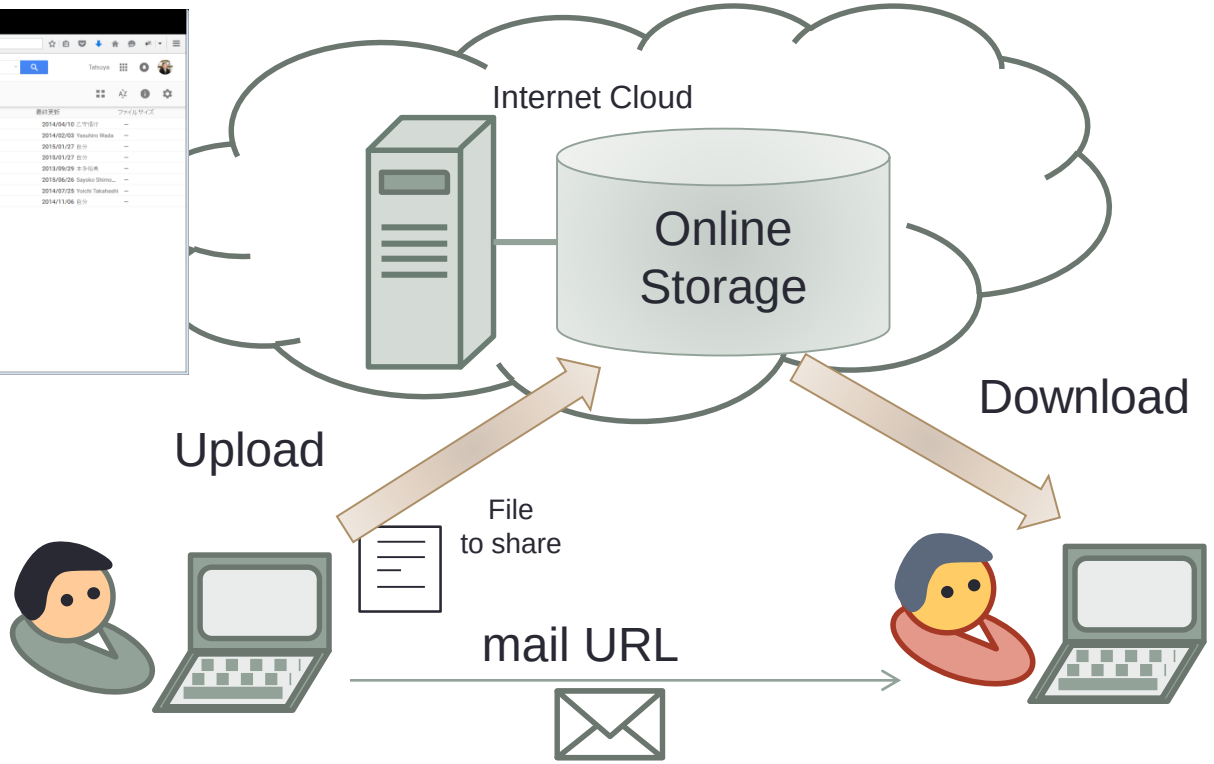
File Sharing

- Online Storage
  - Use Web site for upload and download files.
  - Use special client software.
  - Dropbox, Google drive, Sky drive, iCloud, etc.
- File Sharing Application
  - Use FTP for upload and download files.
  - Use P2P file sharing software.
  - Used in a groupware.
  - Some does version management .
- File Sharing by OS
  - Share remote files like local files.
  - Access transparency

Online Storage

- Place files in Internet Cloud.
  - Use Web interface to manipulate (upload, download, rename, delete, share with others, etc.)
  - Special software may be used to automatically synchronize with local folders.
  - Use accounts or URL to share files with others.

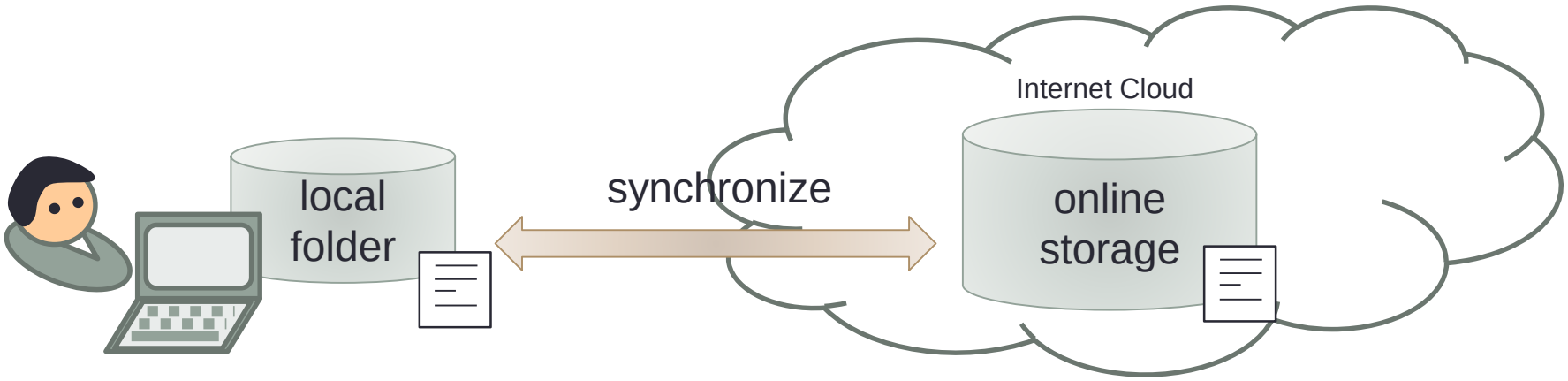




Automatic Synchronization

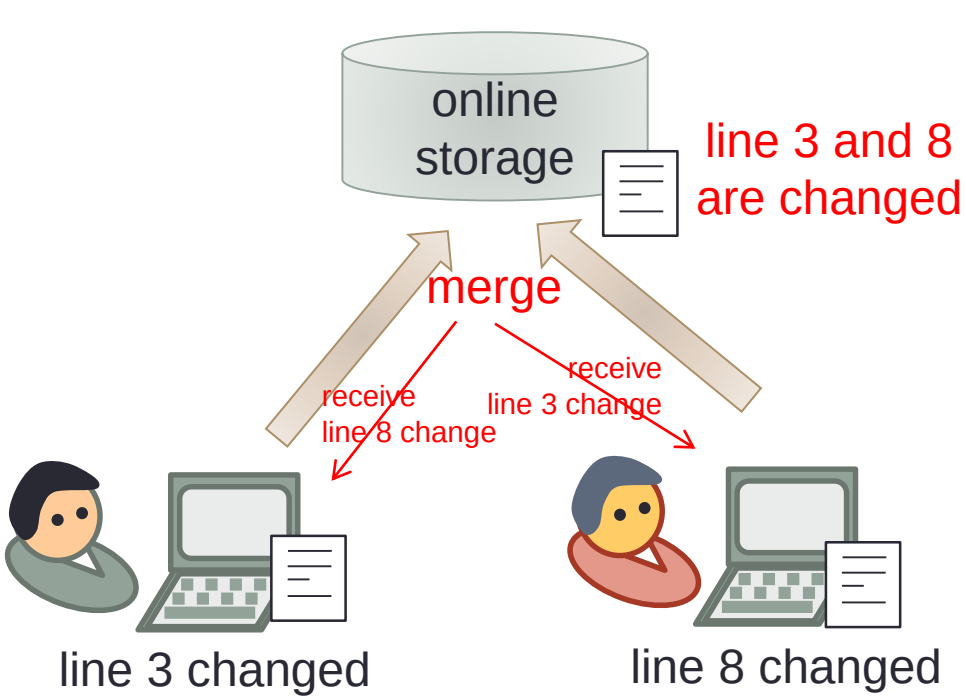
- Special software for synchronization
  - Each online storage has own synchronization software.
  - When local files are changed, they are automatically uploaded.
  - When online storage files are changed, they are automatically downloaded.

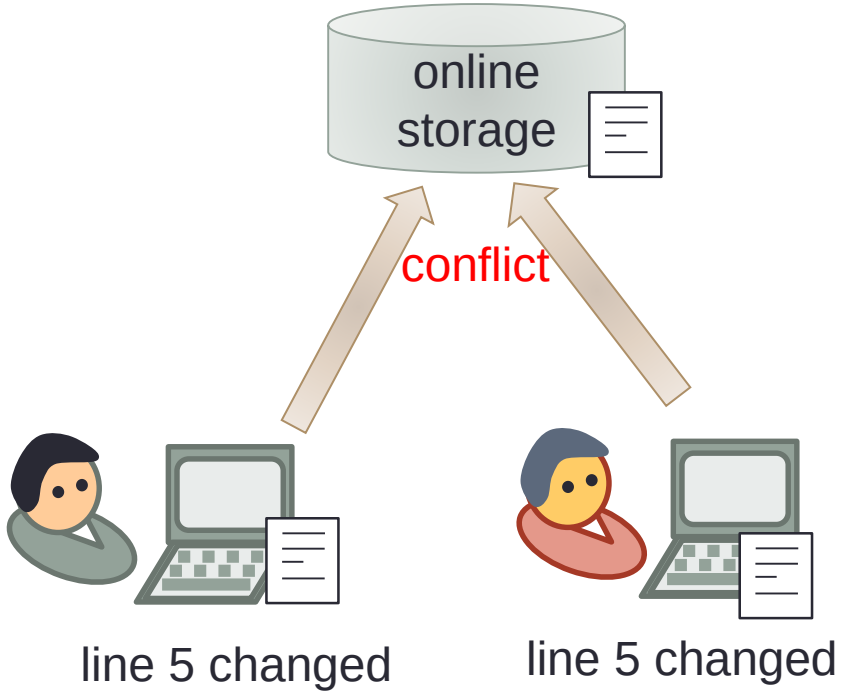
- Mechanism
  - Periodically compare the local folder and the online storage.
  - Update when there are changes.
  - If not connected to Internet, no update.
  - Update when connected.



Merging Modifications

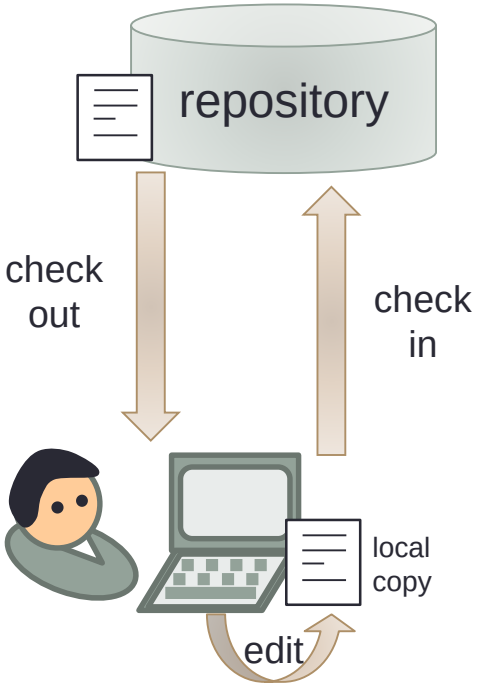
- When a file is shared by multiple people:
  - Multiple people may change the same file.
  - When different lines are changed:
    - merge changed
    - can be done automatically
  - When the same line is changed:
    - if the change is the same, no problem.
    - if it is different, conflict needs to be sorted manually.

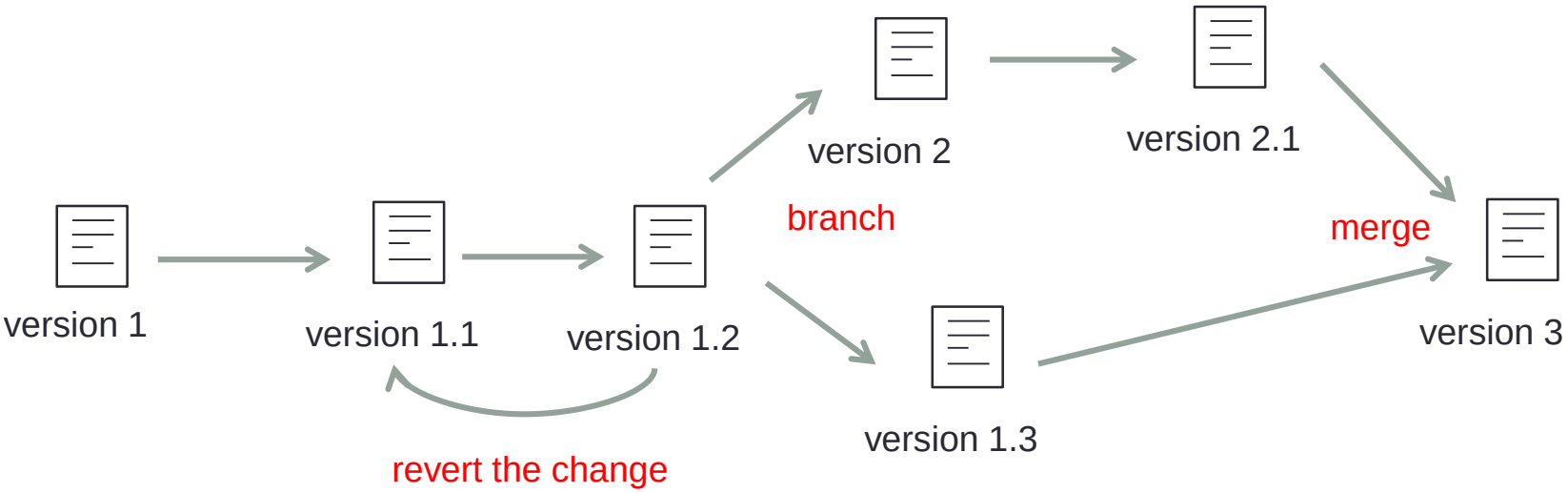




Version Control System

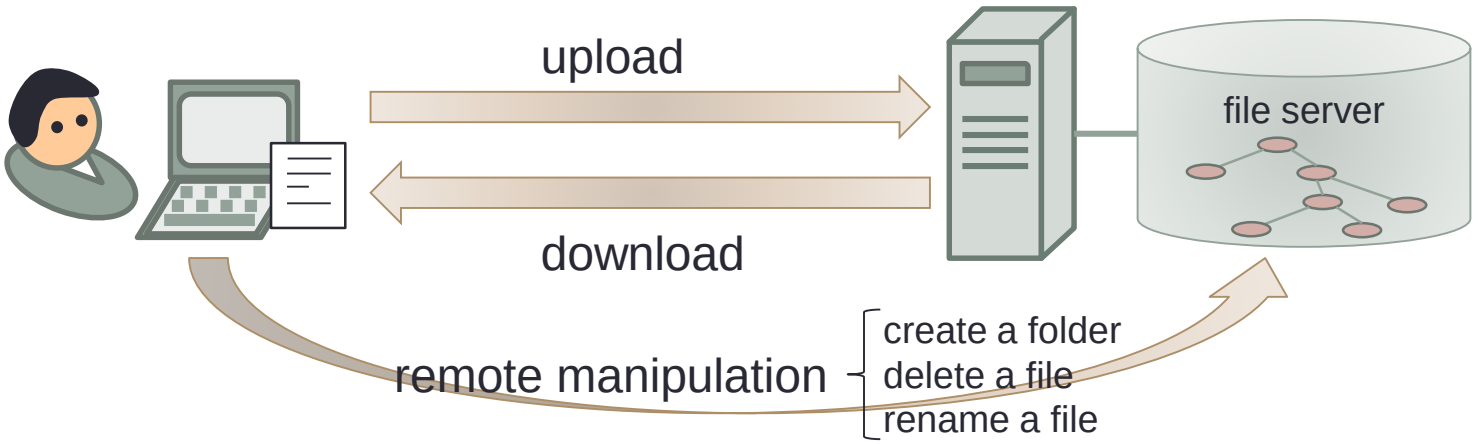
- Manage file changes.
  - Revision control.
  - Can revert the change.
  - Can create branches.
  - Necessary for large scale software development.
  - Version control system
    - RCS (Revision Control System)
    - CVS (Concurrent Versions System)
    - Subversion (svn)
    - git





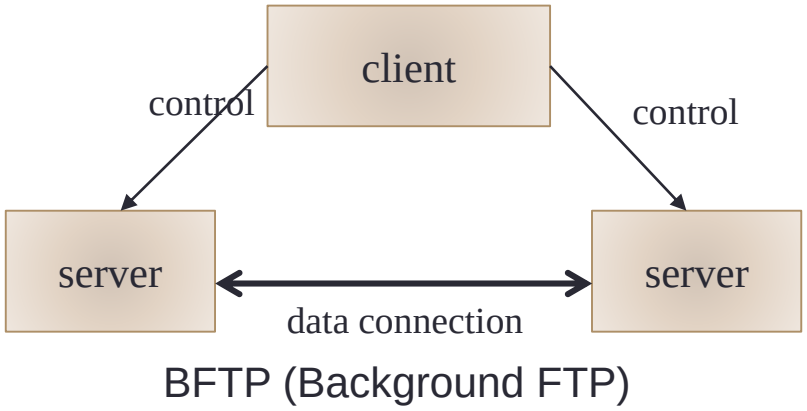
File Sharing Application

- Use file transfer program
  - Get files from a server, edit them and put them back to the server.
  - File transfer protocol
    - FTP (File Transfer Protocol)
      - remote file manipulation
    - HTTP (Hyper Text Transfer Protocol)
      - Web page protocol
      - remote file manipulation with WebDAV extension

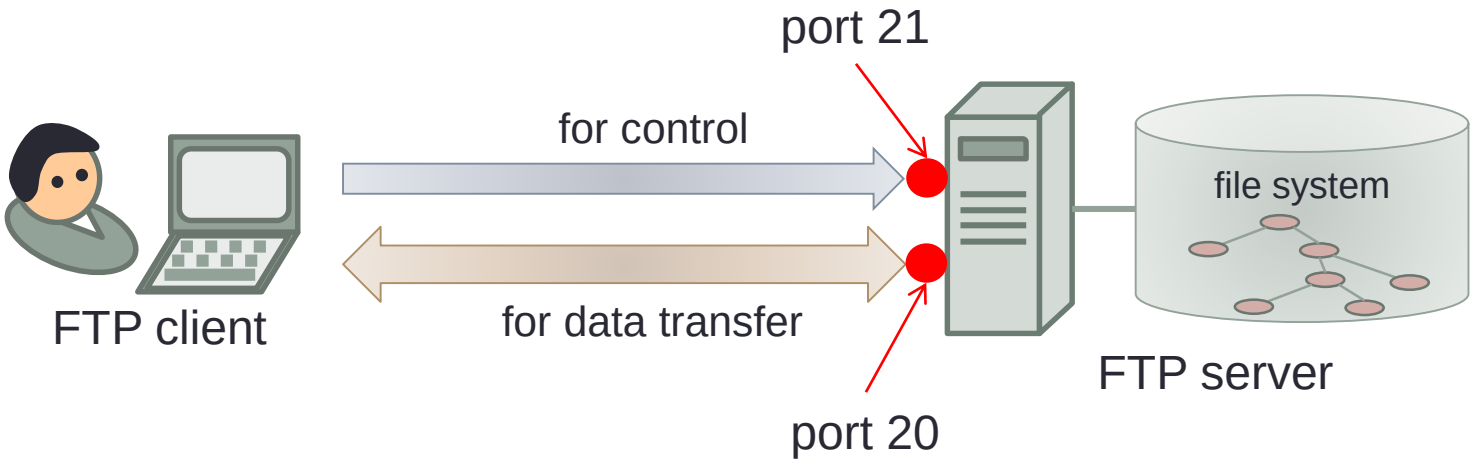


FTP

- File Transfer Protocol
  - One of the oldest protocols
  - TCP port 20 and 21
- Client server model
- Use two connections
  - Control connection: commands
  - Data connection: data transfer



BFTP (Background FTP)



Data Transfer Function

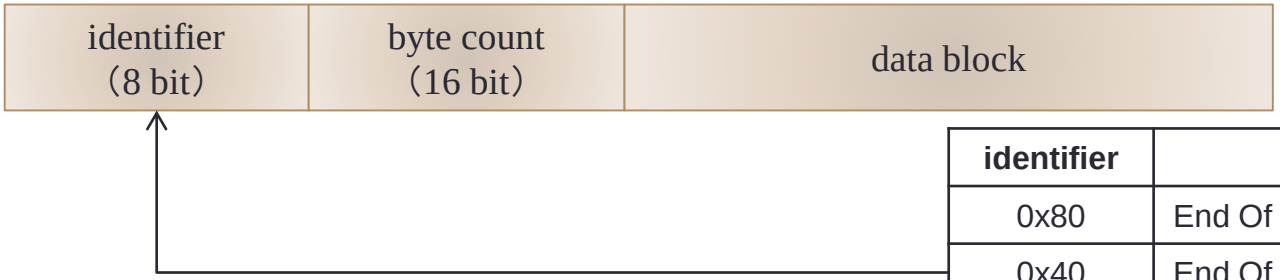
- Data type
  - ASCII
    - <CR><LF> for end of line
  - IMAGE or BINARY
    - records are padded with 0
- File format
  - none print
  - Telnet format control
    - <CR>, <LF>, <NL>, <FF>
  - carriage control
    - the first letter controls:
      - blank normal line
      - 0 double space line
      - 1 new page
      - + overwrite the same line
- File structure
  - Byte structure
    - No internal structure
    - Each file is a sequence of bytes.
  - Record structure
    - Each file consists of a set of records.
  - Page structure
    - Random access file

FTP Transfer Mode

- Stream mode
  - TCP stream as data
  - 0xff is used for ESC letter

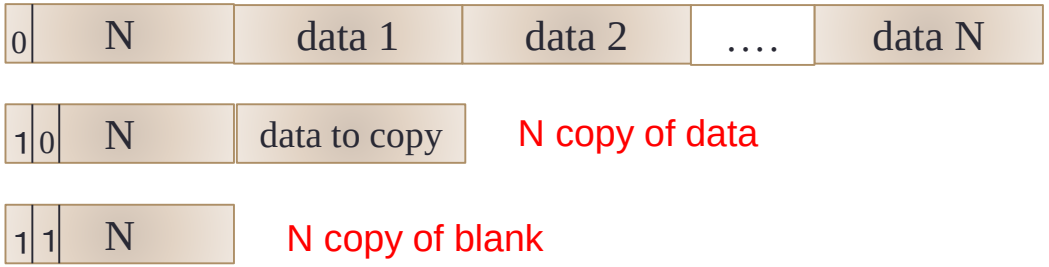
0xff 0x01	End Of Record
0xff 0x02	End Of File
0xff 0x03	EOR and EOF
0xff 0xff	0xff itself

- Block mode



identifier	meaning
0x80	End Of Record
0x40	End Of File
0x20	Error might be in data
0x10	Restart marker

- Compress mode



FTP Control Command (1)

- Access

command	meaning
USER	user name
PASS	password
ACCT	account information
CWD	change working directory
CDUP	change to parent directory
SMNT	mount
REIN	reinitialize
LOGOUT	logout

- Transfer parameter

command	meaning
PORT	data connection host port
PASV	passive listener port
TYPE	data type • ASCII • BINARY
STRU	file structure • F (byte structure) • R (record structure) • P (page structure)
MODE	transfer mode • S (stream mode) • B (block mode) • C (compress mode)

FTP Control Command (2)

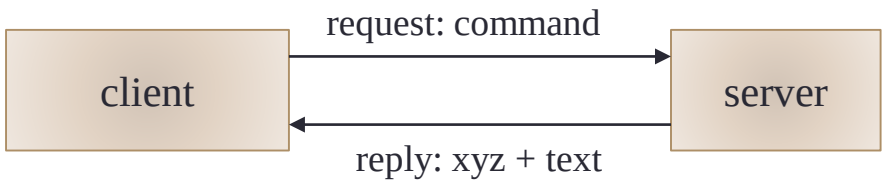
- Service

command	meaning
RETR	Get or retrieve a file
STOR	Put or store a file
STOU	STOR with unique name
APPE	append
ALLO	allocate a new file
REST	skip to the restart marker
RNFR	rename from
RNTO	rename to
ABOR	abort transfer

command	meaning
DELE	delete a file
RMD	delete a directory
MKD	make a directory
PWD	show current working directory
LIST	list files in the current working directory
NLST	LIST with options
SITE	execute command
SYST	get system name
STAT	get current status
HELP	show FTP commands
NOOP	no operation

FTP Reply

- Format
  - 3 letter code + text

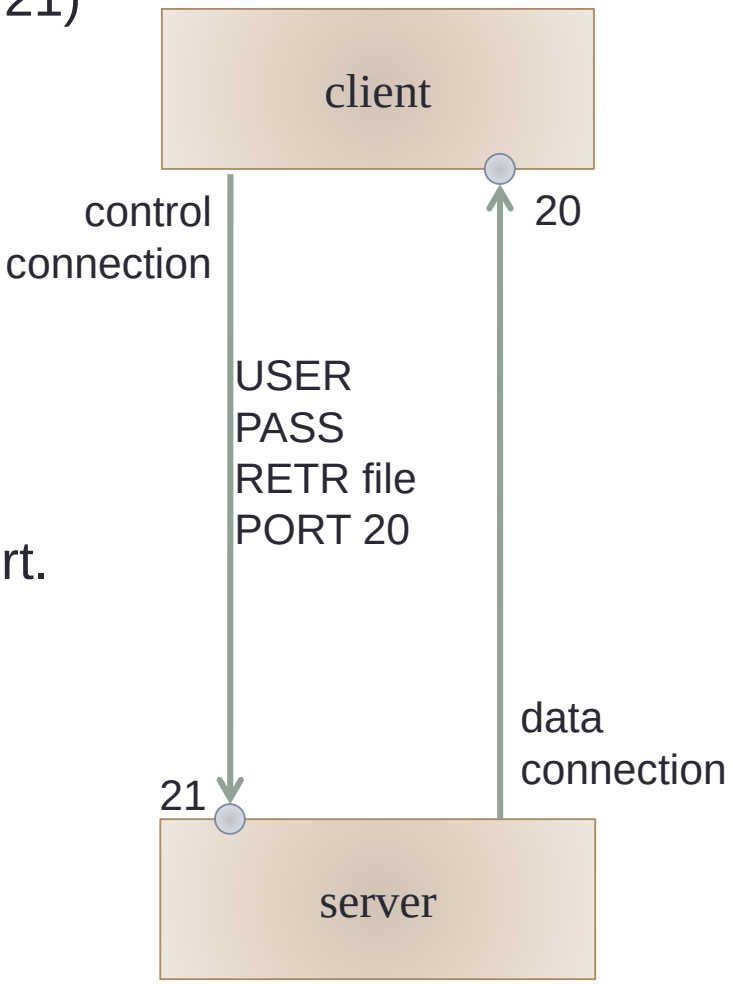


- The first letter
  - 1yz positive preliminary reply
  - 2yz positive completion reply
  - 3yz positive intermediate reply
  - 4yz transient negative completion reply
  - 5yz permanent negative completion reply
  - 6yz protected reply

- The second letter
  - x0z syntax
  - x1z information
  - x2z connection
  - x3z authentication and accounting
  - x5z file system

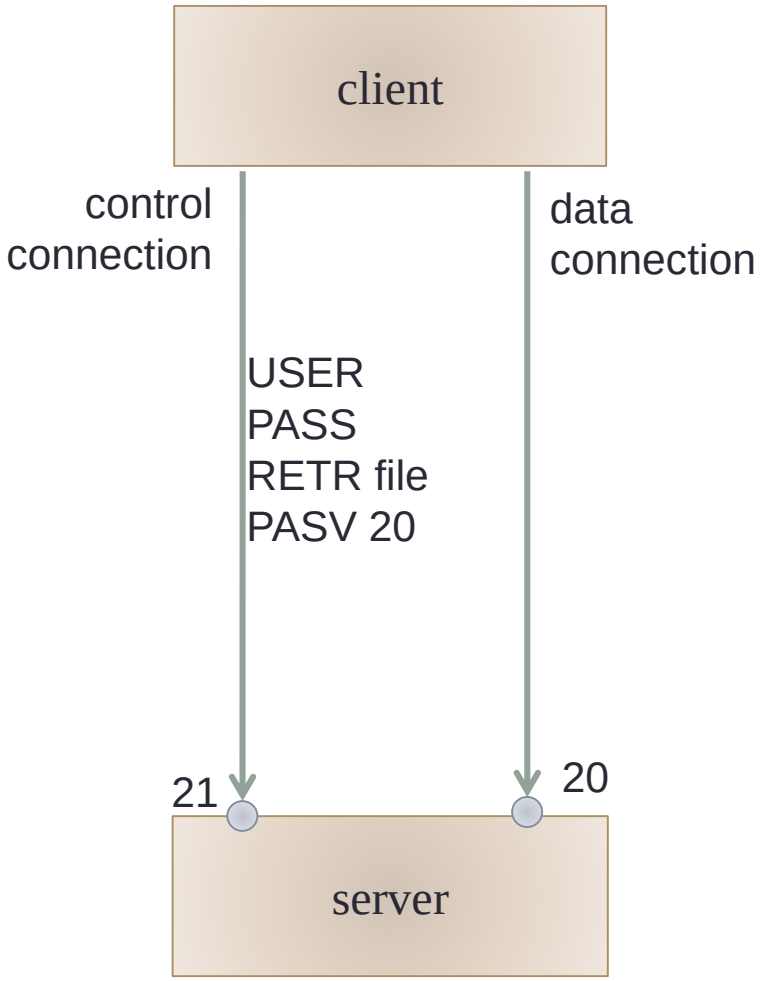
FTP Session Example

- Client opens TCP control connection (port 21)
- Login with USER and PASS
- Specify file by RETR
- Specify data connection port by PORT
- Server connects to the data connection port.
- Transfer data from server to client.
- QUIT



FTP Passive Mode Example

- Client open TCP control connection (port 20)
- Login with USER and PASS
- Specify file with RETR
- Specify data connection port by PASV
- Server waits at the data connection port
- Client connects the data connection port
- Transfer data from server to client
- QUIT



FTP

- Security issue:
  - PASS send the password as plain text
  - Should be careful to use on Internet

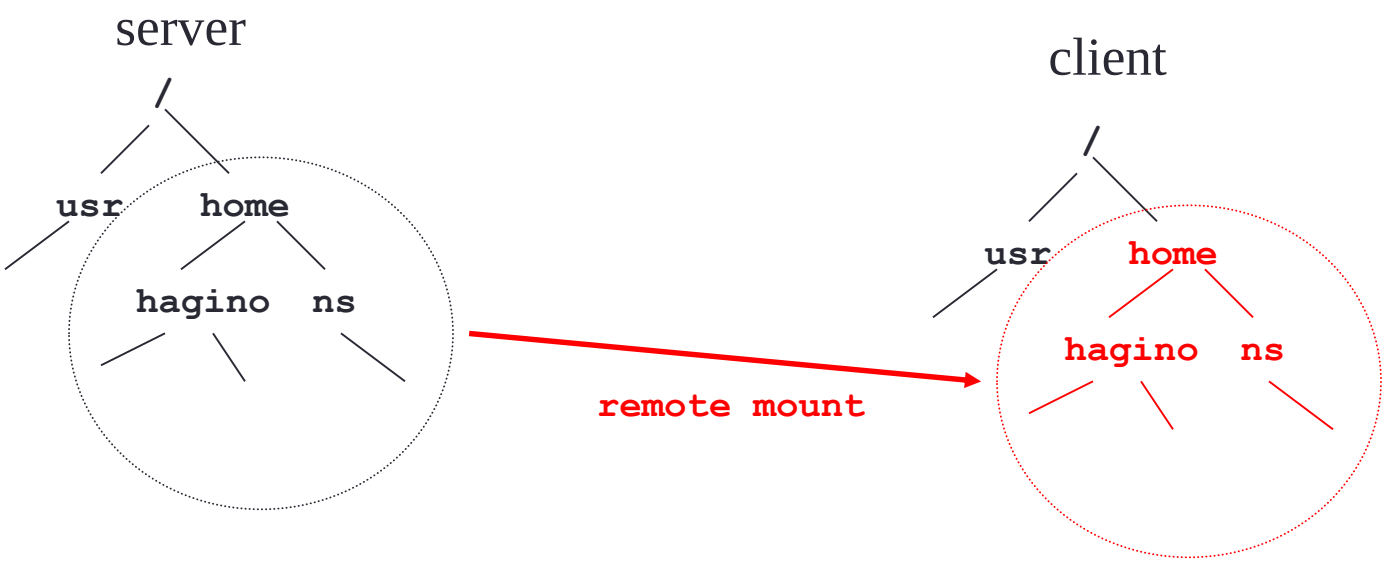
- Anonymous FTP
  - USER anonymous or ftp
  - PASS for mail address (no authentication)
  - Used for distributing free software
  - HTTP is known as an improved version of Anonymous FTP.
  - Web browser is often capable of an anonymous FTP client.

File Sharing by OS

- UNIX
  - NFS
    - Network File System
    - Sun Microsystems (now Oracle) developed
  - AFS
    - Andrew File System
    - CMU developed
- Windows
  - SMB protocol (its extension CIFS)
    - Server Message Block, Common Internet File System
    - IBM designed for NetBIOS
    - Microsoft extended as CIFS
- Mac OS
  - AFP
    - Apple Filing Protocol
    - One of AppleTalk protocol
    - `AFP over TCP' for TCP/IP

NFS (Network File System)

- Distributed file sharing protocol for UNIX
  - Sun Microsystems developed
  - Competed once with RFS (Remote File Sharing) developed by AT&T, but NFS won.
- Mount a remote file server tree like a local disk.
  - Sever exports file system sub trees
- Originally UDP protocol, now available for TCP
  - port 2049

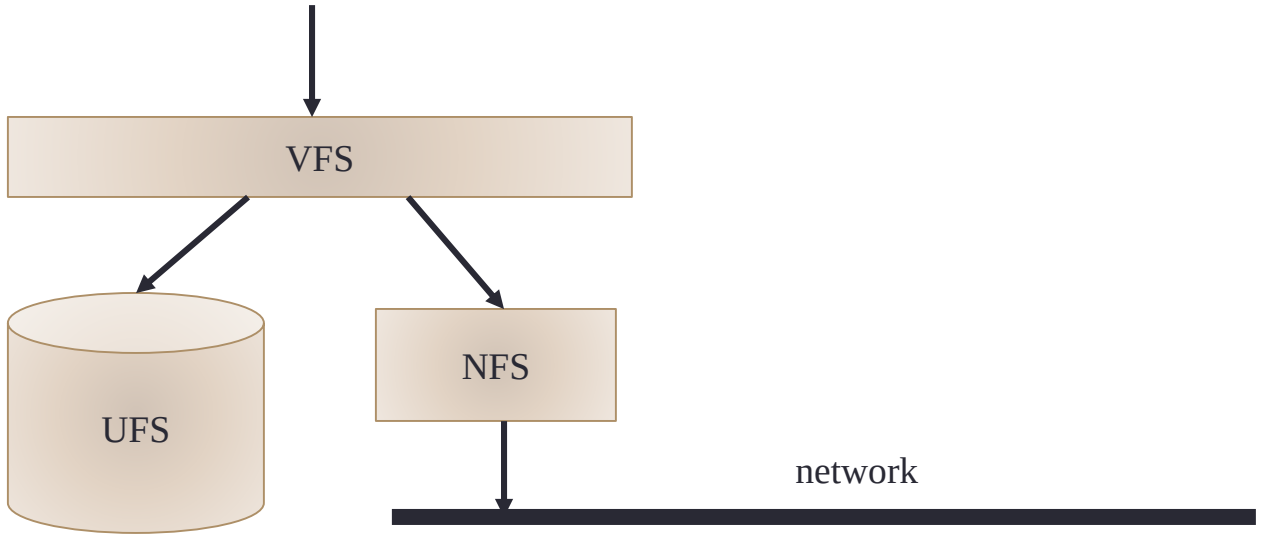


NFS Transparency

- Access transparency
  - Any distributed file system needs to have access transparency.
- Location transparency
  - Name space is local
  - Can be mounted anywhere
- Failure transparency
  - Stateless
  - Can retry failed operations
- Performance transparency
  - Use cache for improving performance
- No replication transparency
  - NFS server cannot be replicated.
- No concurrent transparency
  - Remote files cannot be locked.
- No scale transparency
  - NFS for one organization.
  - Users need to be managed.
  - Use NIS (YP) for user management.

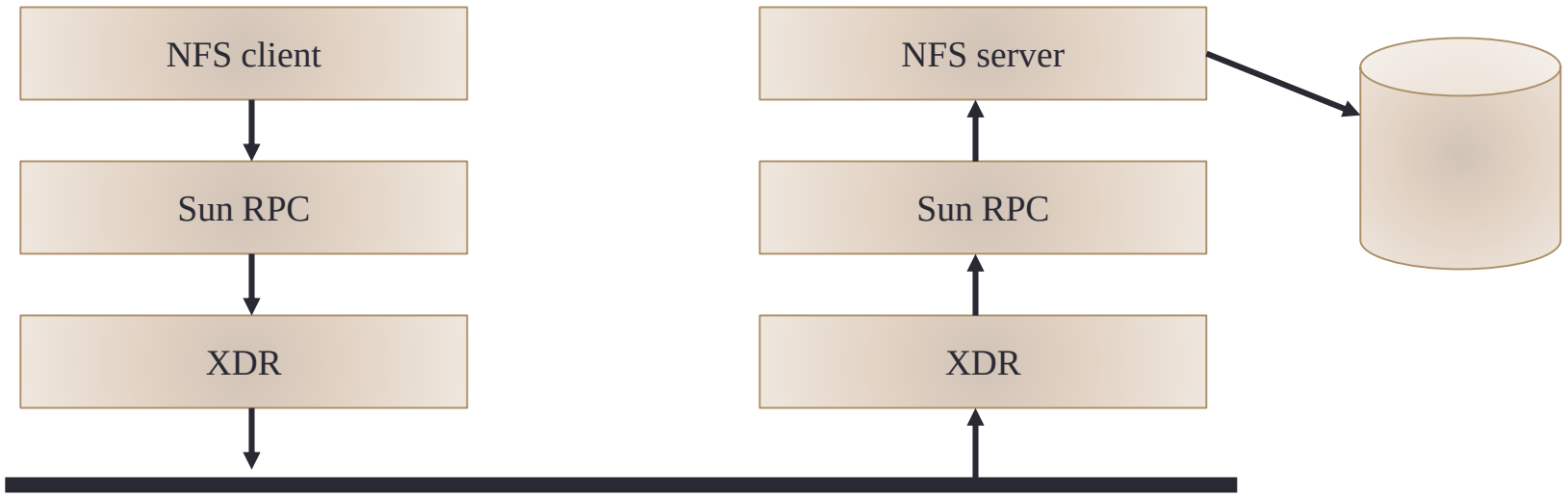
NFS Implementation (1)

- VFS (Virtual File System) layer switches local and remote file systems.



NFS Implementation (2)

- Use Sun RPC
  - Use XDR (External Data Representation) to represent network data.

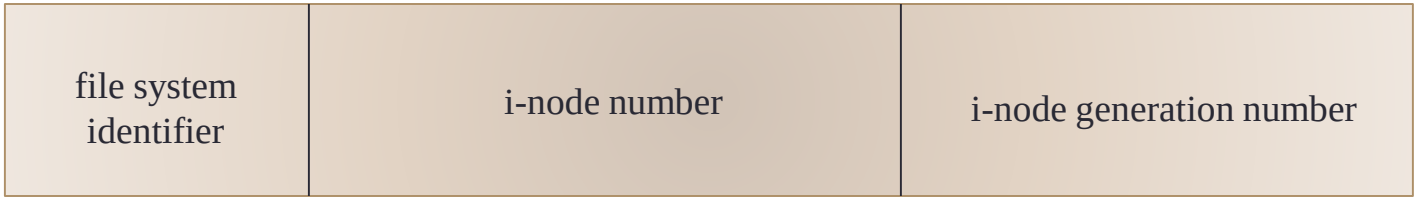


NFS Protocol Interface

- `lookup(dirfh, name)`
  → `fh, attr`
- `create(dirfh, name, attr)`
  → `newfh, attr`
- `remove(dirfh, name)`
  → `status`
- `getattr(fh)`
  → `attr`
- `setattr(fh, attr)`
  → `attr`
- `read(fh, offset, count)`
  → `attr,data`
- `write(fh, offset, count, data)`
  → `attr`
- `rename(dirfh, name, todirfh, toname)`
  → `status`
- `link(newdirfh, newname, dirfh, name)`
  → `status`
- `symlink(newdrfh, newname, string)`
  → `status`
- `readlink(fh)`
  → `string`
- `mkdir(dirfh, name, attr)`
  → `newfh, attr`
- `readdir(dirfh, cookie, count)`
  → `entries`
- `rmdir(dirfh, name)`
  → `status`
- `statfs(fh)`
  → `fsstatus`

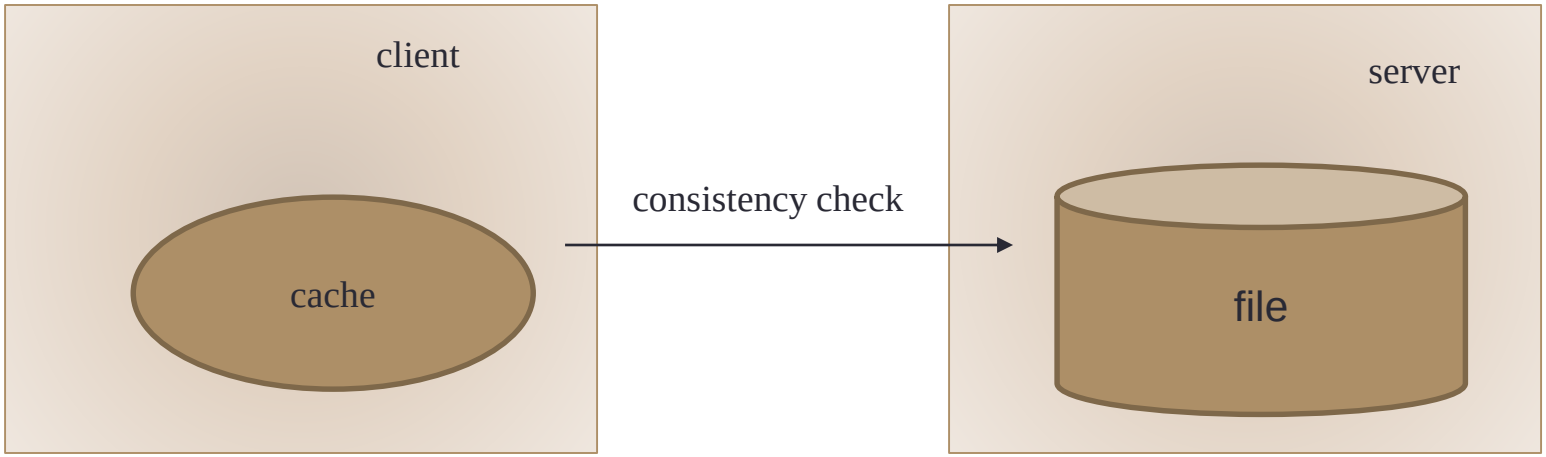
VFS

- Virtual File System
- For local file system, an i-node number is used to identify a file.
  - i-node = index node
- NFS uses a file handle for identification of each file.
  - file system identifier
  - i-node number of the file
  - i-node generation number
- File handle is opaque (not transparent)
  - Client should not look inside the file handle.
  - Capability token for accessing the file.



NFS Cache

- NFS client caches file blocks.
  - Need to check consistency of cached blocks.
  - Server does not notify clients about change. (stateless server)
  - For ordinary files, cache TTL is 3 seconds.
  - For directories, cache TTL is 30 seconds.



Stateless vs Statefull

- NFS server does not mange NFS client state
  - High failure transparency
  - Use cache for performance
  - Cannot lock files on NFS
  - UNIX file semantics may not be applied to remote files.

- Statefull
  - After failure recovery, server needs to know all the client state.
  - May support locking files.
  - Sever can be replicated.
  - RFS tried to keep UNIX file semantics for remote files.

AFS (Andrew File System)

- Developed by CMU (Carnegie Mellon University)
  - Share files amount 5000 workstations on campus

- Features
  - Cache whole files
  - Use kerberos for authentication
  - Can set ACL (Access Control List) to directories
  - Can share files among different organizations.
    - Keio used to share files with CMU using AFS.

AFS

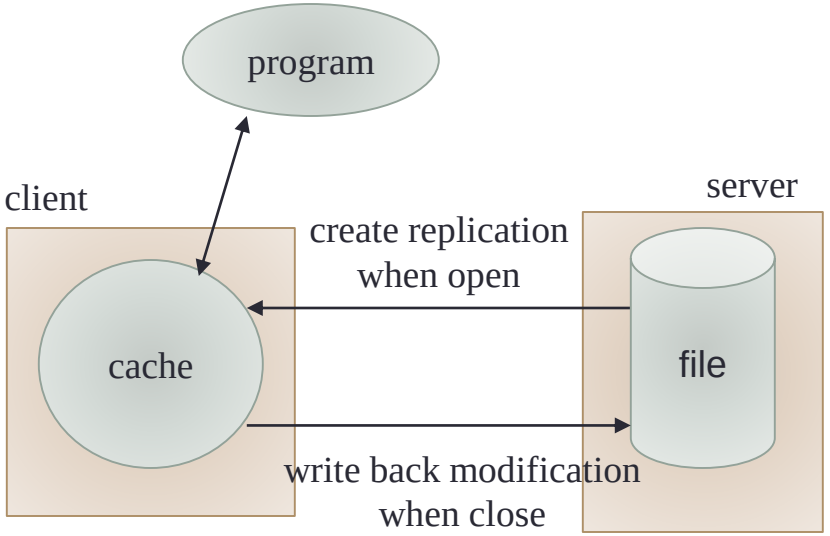
1. Client opens a file.

2. Create a replication of the file in the local cache.
   - If the cache is up-to-date, no need to replicate.
   - Server needs to manage replication.

3. Modify the file in the cache locally.

4. Client closes the use of the file.

5. Write back the modification.
   - Keep the replicated file.
   - The entire file is overwritten by the last write.



NFS vs AFS

	NFS	AFS
Features	• Stateless • High failure transparency • Can be used on various platforms (windows, mac, ...)	• Server can mange replication. • Can share files beyond one organization. • Kerberos authentication • Can set ACL to directories. • Coda for mobile environment
Issues	• No replication • Limited in one organization • User authentication is left to OS. • No ACL	• ACL is complicated. • Less failure transparency • Performance problem for replication

Summary

- File sharing using online storage
  - Web interface
  - Dropbox, Google Drive, Sky Drive, iCloud, etc.
  - Automatic synchronization software
- File sharing by file transfer program
  - FTP
- File sharing by OS
  - Distributed file system
  - NFS
  - AFS