



Intelligent  
Systems  
Lab

---

# Administration of Operating Systems

FTP

Chapter 19

November 9, 2011

# FTP


---



- File Transfer Protocol
- RFC 959, 1579, 2228
- Transferring files to/from remote host
- User/Password authorisation
- Upload, download, delete, rename and list files
- Create and remove directories
- Uses two parallel connections
  - control connection
  - data connection
- Maintains state information about each user
  - current directory, transfer mode, ...

# Example

---



```
220 FTP Server ready.  
user slawek  
331 Password required for user slawek  
pass slawek  
230-User slawek logged in  
230-Enjoy your stay.  
quit  
221 Come back soon!
```

# Return Codes

---

- 1xx Positive Preliminary reply
- 2xx Positive Completion reply
- 3xx Positive Intermediate reply
- 4xx Transient Negative Completion reply
- 5xx Permanent Negative Completion reply
- 6xx Protected reply



# Return Codes

---

- x0x Syntax
- x1x Information
- x2x Connections
- x3x Authentication and accounting
- x4x Unspecified as of RFC 959
- x5x File system





# Return Codes

---

- 1 2 5 Data connection already open; transfer starting.
- 1 5 0 File status okay; about to open data connection.
- 2 0 0 Command okay.
- 2 0 2 Cmd not implemented, superfluous at this site.
- 2 1 2 Directory status.
- 2 1 3 File status.
- 2 1 4 Help message on how to use the server or the meaning of a particular non-standard command. This reply is useful only to the human user.
- 3 3 1 User name okay, need password.
- 3 3 2 Need account for login.
- 4 3 0 Invalid username or password
- 4 5 0 Requested file action not taken.
- 5 0 2 Command not implemented.
- 5 0 3 Bad sequence of commands.



# Control Connection

---



Intelligent  
Systems  
Lab

- Interactive protocol
  - several commands and replies in one session
- Connection starts with authorisation — most other commands are disabled until logged in
  - mkd directory
  - 530 Please login with USER & PASS
- Control connection is for user commands
  - server replies with status information
- Real data is sent over data connection
  - file upload & download
  - directory listing

# Data Connection

---



- Active mode:

```
port 130,235,16,76,16,2  
200 PORT command successful.
```

- next **STOR/RETR/LIST** will cause *the server* to connect to a given IP and port number

- Passive mode:

```
PASV
```

```
227 Entering Passive Mode
```

```
(130,235,16,75,16,2)
```

- for next **STOR/RETR/LIST** *the client* should connect to a given IP and port number



# Design Rationale

---



- Two connections offer a number of benefits
- Make some interesting features possible
  - abort a transfer
  - copy files directly between two FTP servers
  - perform actions during file transfer
  - transfer several files in parallel
  - use slave ftp servers

# Protocol Extensions

---

- Basic protocol is very old
  - reading RFC is pretty funny
- Several extensions to the basic protocol
  - download/upload part of the file (resume)
  - date manipulation and conversion
  - language and character encoding
  - control and/or data encryption
  - SITE command
  - FEAT command



# Usage

---



- Most commonly used in web browsers
  - file download can be done using FTP
- Not secure
  - both passwords and data are sent in the clear
  - best suited for public access
  - there are extensions for fixing that
- PORT vs PASV
  - important for firewall and NAT
- sftp
  - emulation of ftp protocol over ssh
  - user-level compatibility
  - not utility-level

# Service Installation

---



Intelligent  
Systems  
Lab

- Find package
  - documentation
  - Google
  - community
- Install software
- Review configuration
  - remove safety lock
- Run utility
- Verify it is working
  - log files
- Start / restart / reload
  - as a daemon

# Server Execution

---



- Stand-alone
  - regular application
  - started from shell
  - possibly by a regular user
- `/etc/init.d`
  - directory that contains shell scripts
  - those are executed at boot time
  - ... and during shutdown
  - starts and stops various daemons
  - can be done by hand
  - for example after configuration change



# Server Execution

---



- Upstart
  - asynchronous, event-based system
  - addition and removal of hardware
  - discovery of storage devices
  - mechanisms for respawning of services
- `inetd`
  - listens on commonly-used ports
  - starts appropriate server when needed
  - saves memory for rarely-used services
  - some security improvement
  - sometimes called *super server*

# Runlevels

---



- Mode of operation of operating system
  - based on Unix System V-style initialisation.
- Conventionally, numbered from 0 to 6
  - 0 – system shutdown
  - 1 – single-user mode
  - 2 – multi-user mode without network
  - 3 – multi-user mode with network
  - 4 – undefined
  - 5 – runlevel 3 plus display manager
  - 6 – system reboot
- Different daemons are started or stopped depending on current runlevel

# update-rc.d

---



- Manage links to init scripts
  - `/etc/rcrunlevel.d/NNname`
  - `/etc/init.d/name`
- `update-rc.d name defaults`
- `update-rc.d name start NN runlevel`
- `update-rc.d name stop NN runlevel`
- Never changes existing configuration
- `update-rc.d name remove`

# Upstart

---

• /etc/event.d/tty1

```
start on runlevel 2
start on runlevel 3
start on runlevel 4
start on runlevel 5
stop on shutdown
respawn
exec /sbin/getty 38400 tty1
```



# Chroot

---



- Change the *root* directory
  - often called *chroot jail*
- Prevents process from accessing files outside
  - those locations cannot be named
  - thus there is no way to get to them
- Also work for process's children
  - otherwise, it would not be that useful
- Not intended to defend against root
  - it is possible to break out of it
  - one known method involves a second chroot
- Can only be done by root
  - prevents users from abusing *setuid*





Intelligent  
Systems  
Lab

**Questions?**

