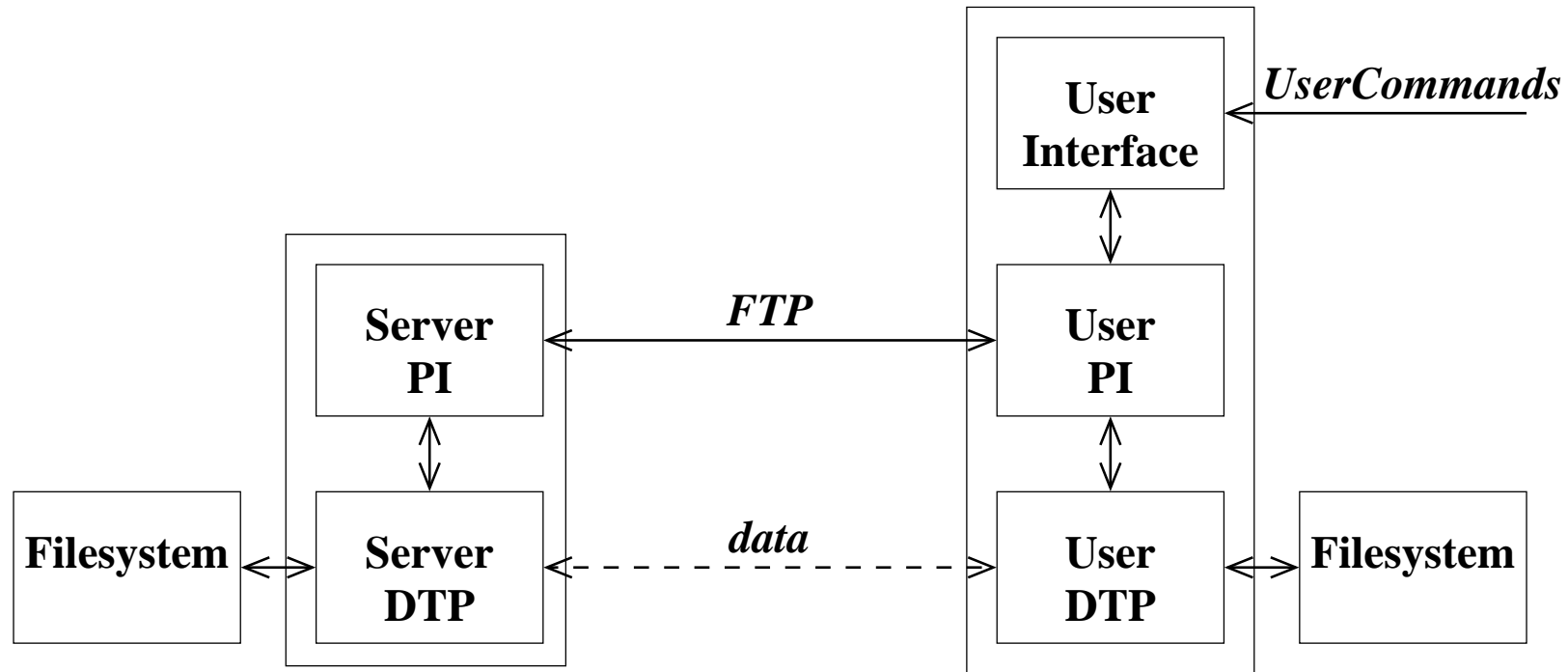


# 4 Dateitransfer

- FTP (File Transfer Protocol) RFC 959 (1985), but goes back to 1971
- objectives (from RFC 959):
  1. *to promote sharing of files (computer programs and/or data),*
  2. *to encourage indirect or implicit (via programs) use of remote computers,*
  3. *to shield a user from variations in file storage systems among hosts, and*
  4. *to transfer data reliably and efficiently.*
- FTP, though usable directly by a user at a terminal, is designed mainly for use by programs. :-)

# FTP Overview



- data connection initiated by server: user DTP must listen  
(exception: passive mode)

- user DTP may be on a *different* host than user PI

⇒ data transfer between two servers

- data ports: user DTP listens on control port, server DTP user adjacent to control port  
alternative ports may be specified

- FTP types: ASCII, EBCDIC, *IMAGE*, LOCAL  
mapping according to different representations: 36 Bit Words, 9 Bit Bytes, 32 Bit Words, ...

# FTP Transmission Modes

- stream mode (default)
  - octet stream (unstructured)
  - eof indicated by closing connection
- block mode
  - blocks of different types with length count
  - restartable
- compressed mode
  - run-length encoding
  - block descriptors embedded (restartable)

# FTP Command Overview

- access commands (enables the remaining commands)  
**USER, PASS, QUIT**
- configuration commands  
**PORT, PASV, TYPE, MODE**
- file commands  
**RETR, STOR, (RNFR, RNT0), DELE**
- directory commands  
RMD, MKD, PWD, CWD, CDUP, LIST
- service commands  
ABOR, STAT, HELP, **NOOP**
- (required commands in **bold face**)
- 3-digit server reply codes

# FTP Example Conversation

C: *establishes control connection to ftp port*  
S: 220 Service ready  
C: USER john  
S: 331 User name ok, need password  
C: PASS mumble  
S: 230 User logged in  
C: *start listening on data port*  
C: *opens local file for writing*  
C: RETR testfile  
S: 150 File status okay; about to open data connection  
S: *connects to data port on client*  
S: *transmits data through data connection*  
S: 226 Closing data connection, file transfer successful  
C: TYPE I  
S: 200 Command OK  
C: *start listening on data port*  
C: *opens local file for reading*  
C: STOR remotefile  
S: 550 Access denied<CRLF>  
C: QUIT  
S: *closes all connections*

# Trivial File Transfer Protocol

- RFC 1350; UDP; very simple
- 5 kinds of packets
  1. Read request (RRQ) with filename and mode
  2. Write request (WRQ) with filename and mode
  3. Data (DATA) with block# and 0-512 octets data
  4. Acknowledgment (ACK) with block#
  5. Error (ERROR) with error code and message

# 5 Mail

## 5.1 Format

- RFC 822 revised as RFC 2822
- original sehr restriktiv:
  - US-ASCII Zeichen 1–127
  - Nachricht besteht aus **Zeilen**  
“Each line of characters **MUST** be no more than 998 characters, and **SHOULD** be no more than 78 characters, excluding the CRLF.”
- Teile
  - Header (*Feldname: Wert*)
  - Leerzeile
  - Rumpf (einzige Einschränkung: Zeilenlänge)



# Headerfelder

orig-date	!	
from	!	
sender	*	
reply-to		
to		aus envelope
cc		aus envelope
bcc		aus envelope
message-id	*	
in-reply-to	R	
references	R	
subject		

- Reihenfolge spielt keine Rolle
- “!” erforderlich, “\*” empfohlen, “R” in Antworten empfohlen

Message-Id: <5.0.0.25.0.20030521140008.00a44ca0@mailgw.ub.uni-freiburg.de>  
Sender: maurer@mailgw.ub.uni-freiburg.de  
Date: Wed, 21 May 2003 14:05:30 +0200  
To: mitarbeiter@informatik.uni-freiburg.de  
From: Beate Maurer <maurer@ub.uni-freiburg.de>  
Subject: Neuerwerbungen  
Mime-Version: 1.0  
Content-Type: text/plain; charset="iso-8859-1"; format=flowed  
Content-Transfer-Encoding: 8bit

Liebe Mitarbeiterinnen und Mitarbeiter,

in der Sitzung der Bibliothekskommission wurde vermutet, dass die  
Neuerwerbungsliste zu wenig bekannt sein könnte.

Deswegen für alle die diesen Dienst noch nicht kennen, der Hinweis auf  
die folgende URL.

<http://www.ub.uni-freiburg.de/neuerwerb.html>

Mit freundlichen Grüßen  
Beate Maurer

## 5.2 Multipurpose Internet Mail Extensions (MIME)

- RFC 2045 - RFC 2049
- RFC 2822 Format zu restriktiv
  - außeramerikanische Zeichensätze verwenden 128-255 im Rumpf und in Headerfeldern
  - Binärdaten
    - \* sind nicht zeilenorientiert
    - \* enthalten "0" Oktette
  - mehrere Ressourcen pro Nachricht (Anhänge, etc)
  - getypte Ressourcen

## 5.2.1 MIME Datenkodierung

- Abbildung von Binärdaten → Textzeilen a la RFC 2822
- Mime-Version: 1.0
- Content-Transfer-Encoding:
  - 7bit für RFC 2822 konforme Daten
  - 8bit Textzeilen wie RFC 2822, aber Zeichen 0-255
  - binary beliebige Folge von Oktetten
  - quoted-printable Kodierung für textnahe Daten
  - base64 Kodierung für beliebige Daten

## quoted-printable **Encoding**

- Idee: falls überwiegend Text, soll die kodierte Nachricht noch lesbar sein
- jedes graphische Zeichen (außer =) darf durch sich selbst repräsentiert werden
- jedes beliebige Zeichen mit ASCIIkode  $n$  kann durch  $=HH$  repräsentiert werden, wobei HH zweistellige Hexrepräsentation von  $n$
- zu lange Zeilen müssen mit = auf 76 Zeichen gekürzt werden

## base64 **Encoding**

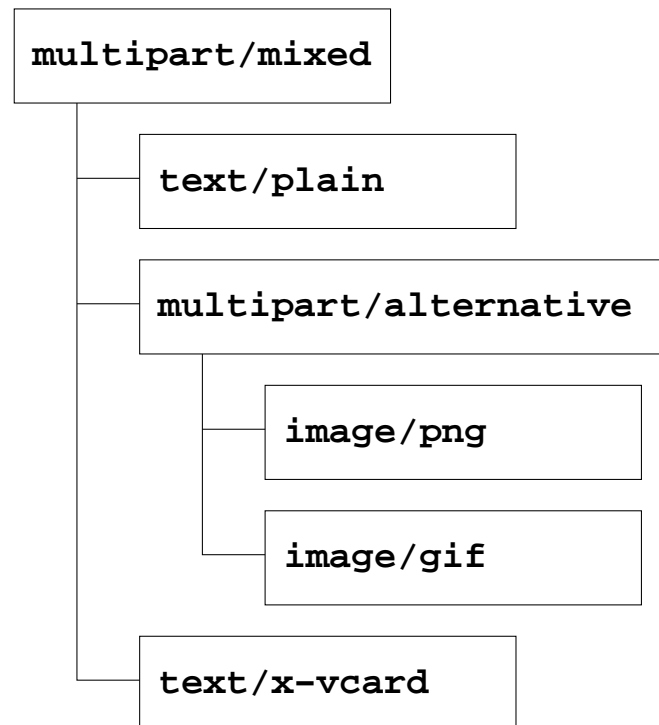
- Kodierung für Binärdaten
- 64 graphische Zeichen zur Kodierung von je 6 Bits (52 Buchstaben, 10 Ziffern, +, /)
- Gruppe von 3 Oktetten  $\mapsto$  Gruppe von vier Zeichen
- Länge implizit; Padding (Auffüllung) durch =

## 5.2.2 MIME Typen

- Content-Type:  $\langle mediatype \rangle / \langle subtype \rangle \langle parameter \rangle^*$
- $\langle mediatype \rangle ::=$   
text | image | audio | video | application | multipart | message
- weitere können bei IANA registriert werden
- $\langle parameter \rangle$  hängt vom Typ/Subtyp ab  
Beispiel:  
Content-Type: text/plain; charset="iso-8859-1"; format=flowed

# multipart Nachrichten

- Nachricht hat Baumstruktur
  - innere Knoten: multipart
  - Blätter: andere Mediatypen





# Kodierung von multipart

- Abtrennung der *Teilbäume* durch Boundary String  
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
- darf *nicht* im Rumpf (Teilbaum) vorkommen
- jedes Vorkommen von multipart hat eigenen Boundary String
- Voreinstellung: Teile haben Typ text/plain

*Vorspann; ignoriert*

--gc0p4Jq0M2Yt08j34c0p

*Header und Body Teilbaum 1*

--gc0p4Jq0M2Yt08j34c0p

*Teilbaum 2*

--gc0p4Jq0M2Yt08j34c0p

*Teilbaum 3*

--gc0p4Jq0M2Yt08j34c0p--

*Nachspann; ignoriert*

## multipart Subtypen

- `mixed`: Sequenz von Ressourcen
- `alternative`: Auswahl von verschiedenen Repräsentationen der gleichen Ressource
- `digest`: Typvoreinstellung `message/rfc822`
- `parallel`: Mehrere Ressourcen, die nebenläufig dargestellt werden sollen

## message Subtypen

- rfc822: Mail Nachricht oder News Artikel
- partial: Teil einer Nachricht, die in Stücken verschickt wurde (z.B. wegen Größenbegrenzung)
- external-body: enthält Verweis auf den eigentlichen Rumpf der Nachricht; Zugriffsmethode durch access-type Parameter:
  - FTP, ANON-FTP, TFTP, LOCAL-FILE, and MAIL-SERVER

## 5.2.3 8Bit Daten in Headerfeldern

- RFC 2047
- Kodierung basierend auf `quoted-printable` oder `base64`
- Umschaltung des Zeichensatzes
- Beispiel:

Subject: =?ISO-8859-3?B?dGhpZW1hbm4sTG9zZSBXZWlnaHQgV2l0aG91dCBEaWV0aW5n?=  
[Disclaimers are not displayed]

[Disclaimer: stammt aus einer SPAM Mail; ich habe es nicht dekodiert]