電子メールとテキストエディタ

コンピュータリテラシ

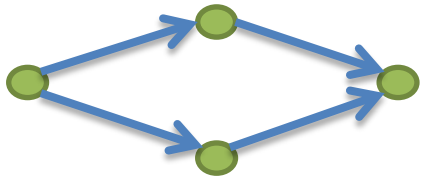
2016年4月26日

建部修見

Internet Protocol (IP)

- 1981年にインターネットプロトコルが標準化
  - RFC791, https://tools.ietf.org/html/rfc791
- パケット（データグラム）の転送
  - 情報をパケットで分割し転送

- 冗長なネットワーク経路
  - 後に標準化されるBGPによる経路選択
  →障害に強い



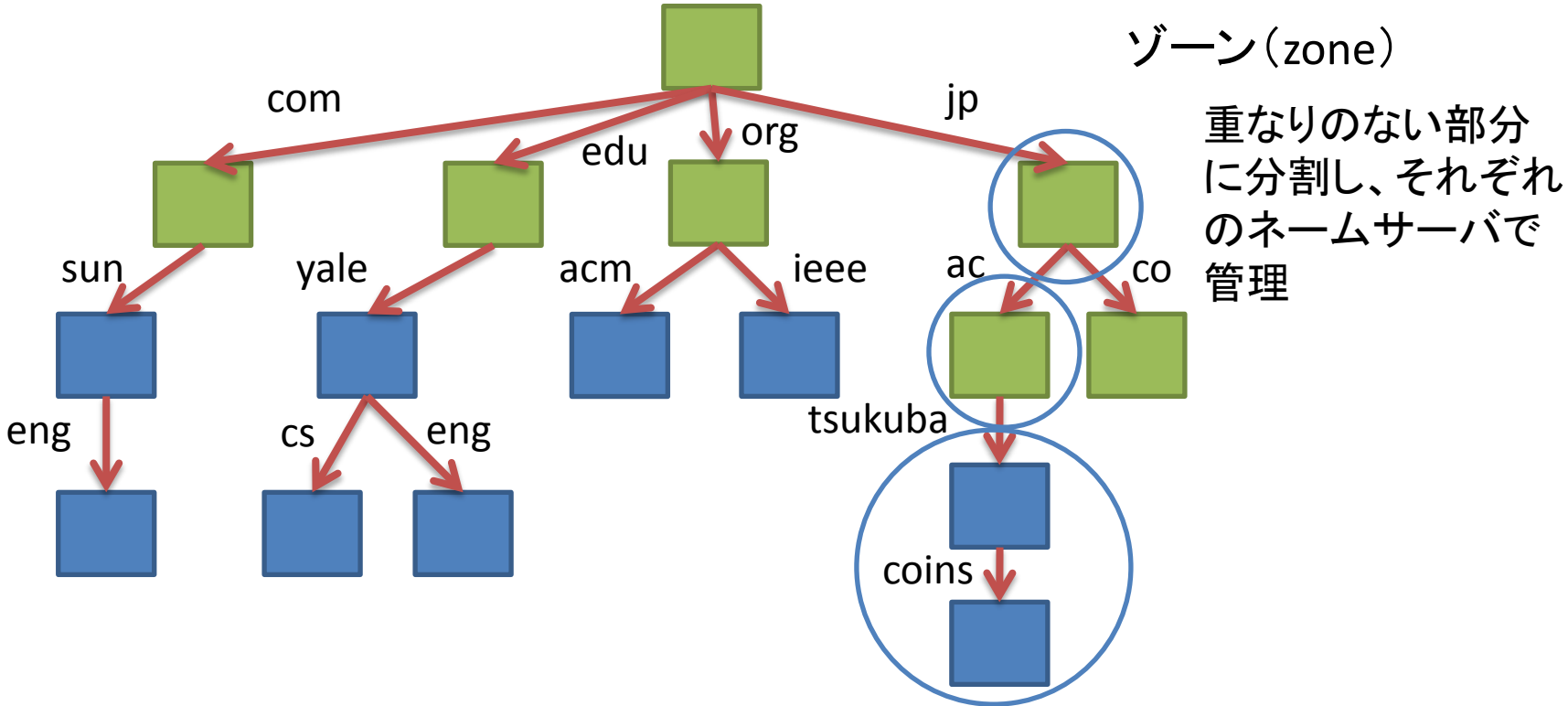
- IPアドレス
  - パケット転送の宛先、送信元
  - 130.158.0.1など32ビットのアドレス
  - 32ビットのアドレスは枯渇。1998年に128ビットのアドレス IPv6が標準化

Transmission Control Protocol (TCP)

- RFC793, https://tools.ietf.org/html/rfc793
- インターネットプロトコルの上でホスト間の信頼性のある通信プロトコル (TCP/IP)
- パケットの破損、損失、重複、順序の入れ替わりを検出し、修復あるいは再送
  - シーケンス番号、ACK、チェックサム
  - ACKがタイムアウト（Retransmission Timeout; RTO）したら再送
  - 後に高速再送、SACKなど [RFC2001]
- パケットを流しすぎてネットワークを輻輳させないための流量制御
  - 受信側がACKとともに送信可能量を指定
  - パケットロスがあると送信可能量を削減
- 単一ホストで複数の通信を行うためポート（0～65535）による多重送信

Domain Name System (DNS)

- 階層的な名前→IPアドレス
  - www.tsukuba.ac.jp→130.158.69.233
- 最大規模の分散システム



初期のインターネットサービス

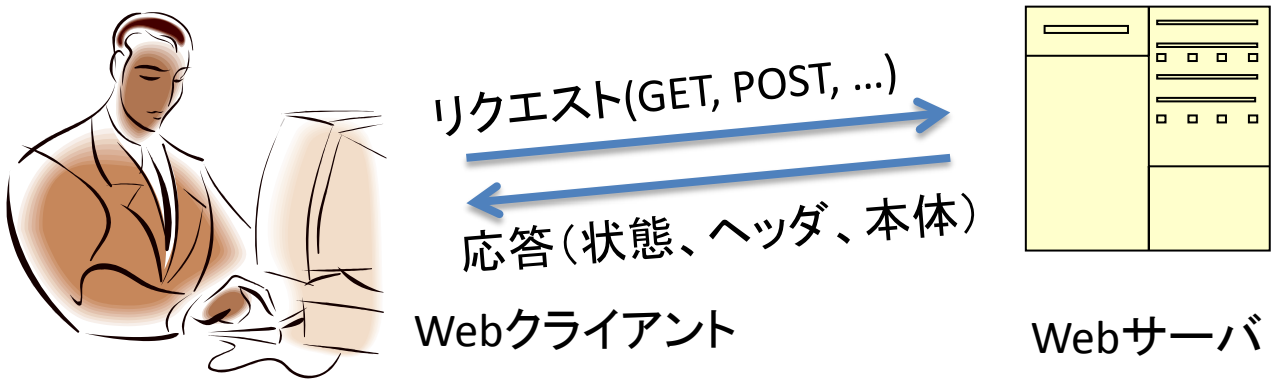
- メール配信 (25/tcp)
  - 1981年にSimple Mail Transfer Protocol (SMTP)が標準化
- ファイル転送 (21/tcp, 20/tcp)
  - 1985年にFile Transfer Protocol (FTP)が標準化
- 遠隔ホスト端末へのアクセス (23/tcp)
  - 1983年にTELNET Protocolが標準化
- echo (7/tcp, 7/udp), discard (9/tcp, 9/udp), character generator (19/tcp, 19/udp), quote of the day (17/tcp, 17/udp), active user (11/tcp, 11/udp), daytime (13/tcp, 13/udp), time (37/tcp, 37/udp), …

インターネットサービスの続き

- 1986年
  - Network News Transfer Protocol (NNTP)
- 1988年
  - Interactive Mail Access Protocol (IMAP)
- 1989年
  - Network File System Protocol (NFS)
  - Network Time Protocol (NTP)
- 1994年
  - Post Office Protocol version 3 (POP3)

Hypertext Transfer Protocol (HTTP)

- 1996年にTim Berners-Lee@CERNらがHTTP/1.0として策定
  - 1990年よりWorld-Wide Web global information initiativeで利用されていたプロトコルを標準化
- Uniform Resource Locator (URL)で文書などを参照
  - http://www.tsukuba.ac.jp/admission/index.html
- 応答メッセージはメールで利用されるMultipurpose Internet Mail Extensions (MIME)を拡張して用い、拡張可能で様々なコンテンツを表現
  - テキスト、画像、動画
  - 日本語、英語、フランス語



Hypertext Markup Language (HTML)

- 1993年にMIMEのコンテント型として草案を策定
- 構造化された文章、ハイパーテキスト（リンク）、画像などの埋め込み

```
<HTML>
<TITLE>A sample HTML instance</TITLE>
<H1>An Example of Structure</H1>
Here's a typical paragraph.
<P>
<UL>
   <LI> Item one has an <A NAME="anchor">anchor</A>
   <LI> Here's item two.
</UL>
</HTML>
```

- リンクの例

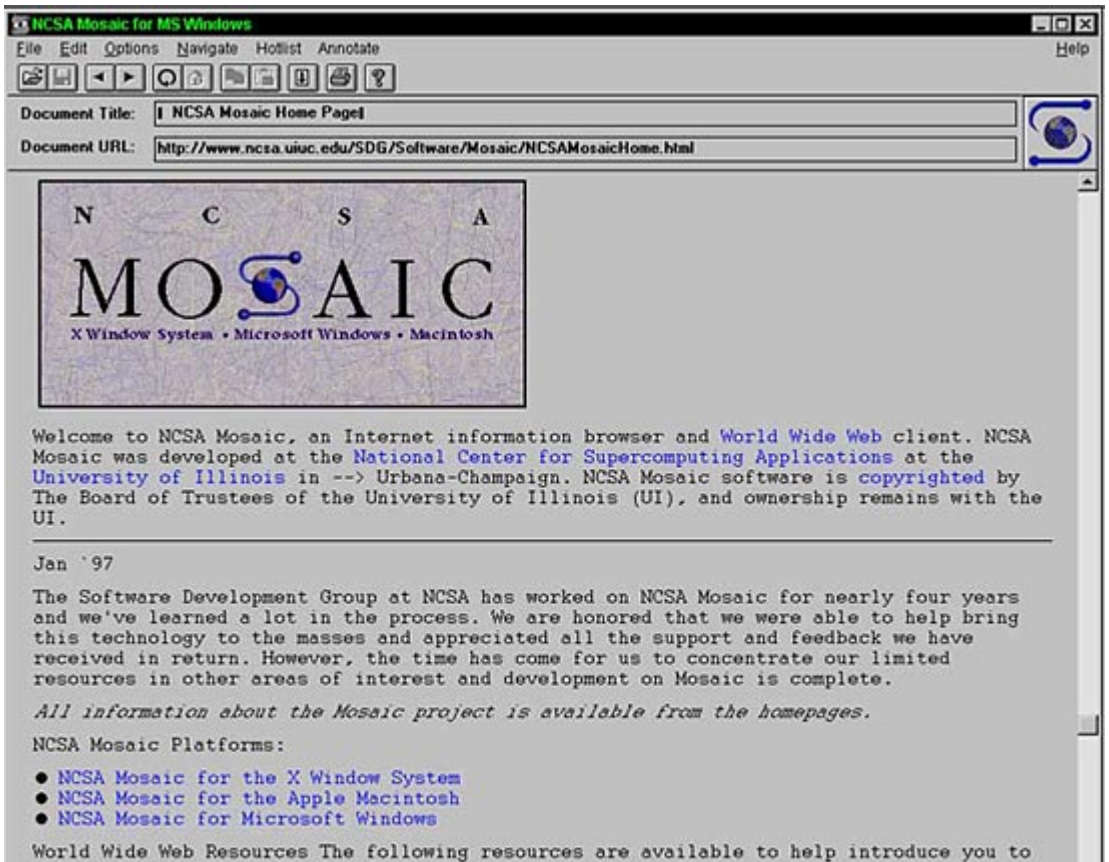
```
See <A HREF="http://info.cern.ch/">CERN</A>'s information for more details.
```

- 画像埋め込みの例

```
Warning: <IMG SRC="triangle.gif" ALT="Warning:"> This must be done by a qualified technician.
<A HREF="Go"><IMG SRC="Button">Press to start</A>
```

WWW黎明期

- 1992年WebサイトがCERNに登場
- 1993年NCSA MOSAICの登場により利用者大幅増



Simple Mail Transfer Protocol (RFC821,RFC5321)

- Sender-SMTPとReceiver-SMTP間のプロトコル

```
MAIL FROM:<sender@domain.name>      # 送信者
250 Ok
RCPT TO:<recipient@domain1.name>    # 宛先1
250 Ok
RCPT TO:<recipient2@domain2.name>   # 宛先2
250 Ok
DATA                                # コンテンツ
354 Ok
Date: 23 Apr 2016 15:00 +0900
From: sender@domain.name

Message for you
.
250 Ok
quit
```

SMTP

- MAIL
  - メールトランザクション開始。Reverse-pathはエラー報告のため

```
MAIL <SP> FROM:<reverse-path> <CRLF>
```

  - 送信先の指定

```
RCPT <SP> TO:<forward-path> <CRLF>
```

  - メールデータ。ピリオド(.)だけの行で終了

```
DATA <CRLF>
```

- OPENINGとCLOSING
  - 送信チャネルの開始と終了。<domain>は自身の身元

```
HELO <SP> <domain> <CRLF>
QUIT <CRLF>
```

エンベロープとコンテンツ

- エンベロープ
  - SMTPで送信される送信者、送信先など
- コンテンツ
  - DATAコマンド
  - ヘッダとメッセージ本体（RFC5322で規定）
- メールはエンベロープで指定される送信先に送られる

Internet Message Format (RFC5322)

- コンテンツの規定
- 文字はUS-ASCII文字コード（1～127の範囲）
  - RFC1458による日本語（ISO-2022-JP）への拡張
- 行の長さ（CRLF 0x0D 0x0A 除く）
  - 998文字を超えてはならない
  - 78文字を超えない方が良い（表示のため）
- ヘッダと本体は空行で区切る
- ヘッダ
  - フィールド名: フィールド本体
  - スペースの前に改行して複数行にできる
- メッセージ本体

メールボックスとグループ

- メールボックス
  - user@domain.name
  - User name <user@domain.name>
- グループ
  - Empty Group:;
  - Group name:u1@domain1,u2@domain2;
- アドレス
  - メールボックス or グループ

ヘッダ

- 必須
  - Date: 25 Apr 2016 15:15 +0900　　＃投函時刻
  - From: foo@example.com　　＃著者
- 発信者
  - From: 著者（アドレスのリスト）
  - Sender: 送信者（Fromと違うときだけ。アドレス）
  - Reply-To: 返信して欲しい宛先（ないときはFromに返信。アドレスのリスト）

注：RFC6854 (2013)でメールボックスのリストからアドレスのリストに変更された

- Nightly Monitor Robot:; など自動送信で返信を期待しないメール送信のため

ヘッダ（2）

- 送信者（アドレスのリスト）
  - To: 主要な宛先
  - Cc: 関係する人々の宛先
  - Bcc: 送信されるが送信メールのヘッダから削除

注：実はここに書いても送信されない。エンベロープで送信先を指定する必要がある。エンベロープは受信者には分からないのでヘッダで送信者を知らせる

ヘッダ（３）

- 全送信メッセージ
  - Message-ID: メッセージID
- 返信メッセージ
  - In-Reply-To: 返信元メッセージID（返信元メールの指定）
  - References: 返信元メッセージID（会話のスレッド作成のため）

ヘッダ（４）

- 情報フィールド
  - Subject: 題目
    - 返信は Re: を題目の先頭につけることが多い
  - Comments: コメント
  - Keywords: キーワード1,キーワード2

ヘッダ（５）

- 再送
  - Resent-Date, Resent-From, Resent-Sender, Resent-To, Resent-Cc, Resent-Bcc, Resent-Message-ID
- トレース
  - Return-Path, Received
  - Receivedでどのサーバがいつどこから受信したかを残す

注：送信者、メッセージ本体は当てにならない。保証するためには電子署名が必要。See S/MIME, OpenPGP (GnuPG)

Multipurpose Internet Mail Extensions (MIME)

- RFC2045, RFC2046, RFC2049
- US-ASCII以外のメッセージ本体
- テキスト以外のフォーマット
- マルチパート（ファイルの添付など）
- US-ASCII以外のヘッダ情報

Secure/Multipurpose Internet Mail Extensions (S/MIME)

- RFC5751 S/MIME Version 3.2
- MIMEデータをセキュアに送受信する
- S/MIME証明書
  - メールアドレス、公開鍵、CAの電子署名
- 電子署名による認証、完全性の保持、また起源の証明の否定不可
  - 電子署名とはハッシュ値を秘密鍵で暗号化したもの。公開鍵で検証可能。秘密鍵の所有者しか暗号化（署名）できない。
- 暗号化による機密性保持（盗聴防止）
- 圧縮によるデータサイズ削減

メールボックス

- メールはメールサーバに配信される
  - mbox形式、Maildir形式
- COINSではMaildir形式
  - ~/Maildirに格納される

POP3とIMAP

- Post Office Protocol (RFC1939)
  - メールサーバからメールを取得
  - メールサーバのメールを残す or 消す
- Internet Message Access Protocol (RFC3501)
  - メールサーバのメールボックスの操作
  - オフラインのクライアントとの同期

SMTPの認証、暗号化、完全性

- 認証はRFC4954で拡張された
  - AUTHコマンド
- 通信は暗号化されない
- データの改竄・破損（データ完全性）の保証ができない
  - Transport Layer Security (TLS)を用いる

Transport Layer Security (TLS)

- RFC5246
- プライバシーとデータ完全性を提供
- TLSレコードプロトコル
  - 対称暗号系（AES、RC4）による暗号化
  - SHA-1などのハッシュ関数によるデータ完全性
- TLSハンドシェイクプロトコル
  - 公開鍵暗号系（RSA、DSAなど）による接続先認証
  - 対称暗号系の共通鍵の安全な交換
  - 信頼性のある接続時の暗号方式などの交渉
- アプリケーションプロトコルに独立

悪意を持った使い方

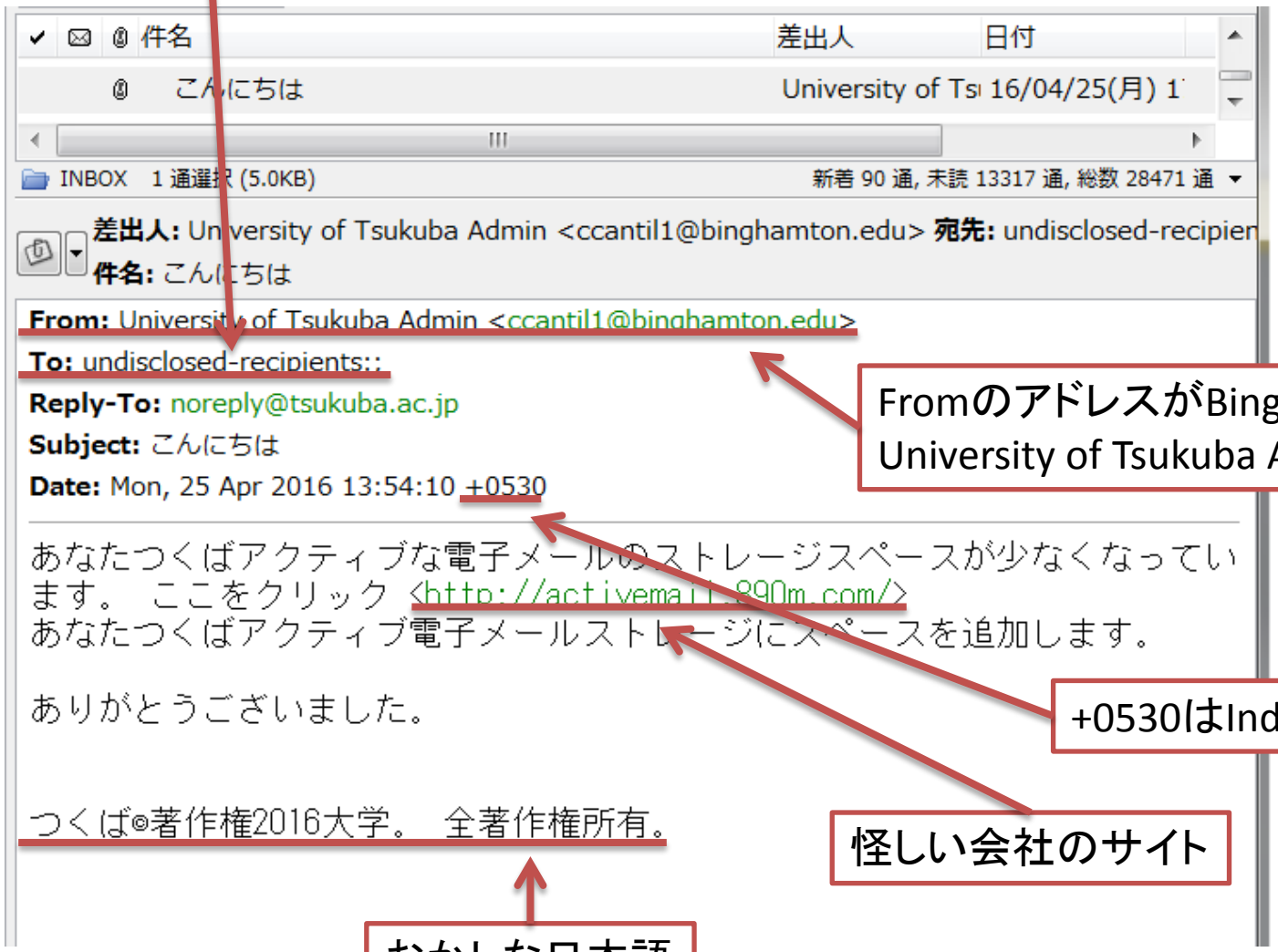
- メールのそれなりに信頼できる情報はReceivedのサーバアドレス
  - （電子署名がない場合）Fromは信用できない
- スパムメール（迷惑メール）
  - 多数の受信者に送付するジャンクメール
  - 広告、宣伝、ウイルス、ワーム、スパイウェア、フィッシング
  - HTMLのプレビュー機能がある場合自動感染も
- なりすましメール
  - Fromはあてにならない

ブラックリスト

- 迷惑メールを送信するクライアント（サイト）はブラックリストにのる
  - Spamhaus, http://www.spamhaus.org/
  - SURBL, http://www.surbl.org/
  - Barracuda Reputation Block List, http://www.barracudacentral.org/rbl
  - …
- メールサーバが受信を拒否する

怪しいメール

Toに何も指定されていない



Fromのアドレスが Binghamton大学なのに University of Tsukuba Adminという名前

+0530はIndia Standard Time

怪しい会社のサイト

おかしな日本語

もう少し調べる

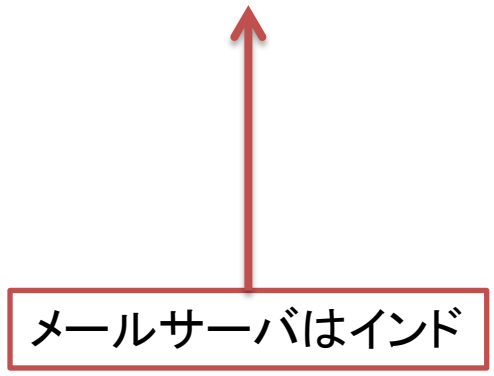
```
$ whois 890m.com
```

- GoDaddy（レジストラ）が所有するドメイン名

```
$ host activemail.890m.com
activemail.890m.com has address 31.220.16.218
activemail.890m.com mail is handled by 10 mx1.hostinger.in.
```

```
$ whois 31.220.16.218
```

- Hostinger Shared Hosting Servers



メッセージヘッダ

```
Received: by mail-io0-f196.google.com with SMTP id u185so2289518iod.2
        for <tatebe@cs.tsukuba.ac.jp>; Mon, 25 Apr 2016 01:24:51 -0700 (PDT)
...
X-Received: by 10.107.142.205 with SMTP id q196mr18901203iod.167.1461572690805;
 Mon, 25 Apr 2016 01:24:50 -0700 (PDT)
...
Received: by 10.36.85.129 with HTTP; Mon, 25 Apr 2016 01:24:10 -0700 (PDT)
Reply-To: noreply@tsukuba.ac.jp
From: University of Tsukuba Admin <ccantil1@binghamton.edu>
Date: Mon, 25 Apr 2016 13:54:10 +0530
Message-ID: <CA+O5kQnmrkNe0E4Ezw0Fjetb1FVWKJWGvv1f4y=7M=tke3HUaA@mail.gmail.com>
...
```

googleのメールサーバで送信されている

10.0.0.0/8（10.0.0.0〜10.255.255.255）はプライベートアドレス

メール転送

- 受信メールを他のメールアドレスに転送する
  - ~/.forward

XXX@ezweb.ne.jpとYYY@docomo.ne.jpに転送

```
XXX@ezweb.ne.jp,YYY@docomo.ne.jp
```

メールを残し、YYY@docomo.ne.jpに転送

```
¥sXXXXXXX,YYY@docomo.ne.jp
```

注：¥はこれ以上の展開を抑制する。¥を付けないと無限ループする

メールクライアント

- Mew（手引き3.5節）
- Thunderbird（手引き4.2節）
- SquirrelMail（手引き4.2.6節）

テキストエディタ

- ed – line editor (1969 Ken Thompson)
- ex, vi – text editor (1976 Bill Joy)
- GNU Emacs – the extensible, customizable, self-documenting, real-time display editor (1984 Richard Stallman)
  - Emacs Lispで拡張
  - 手引き第3章

演習（１）

- 手引き3.2節に従い、Emacsを起動、停止させる
- 手引き3.3節に従い、文字の入力、日本語の入力、カーソル移動、文字・行のカット、コピー、ペースト、編集の取り消し、検索・置換を行う
- 授業の感想を書いて提出する

演習（2）

- 手引き3.5節に従い、Mewでメールの送信、受信、マルチパートのメールの送信を行う
- 手引き4.2節に従い、Thunderbirdでメールの設定、メールの送受信を行う

	プロトコル	サーバ	ポート	暗号	認証
受信	IMAP	violet-nwh.coins.tsukuba.ac.jp	993	SSL/TLS	自動検出
送信	SMTP	violet-nwe.coins.tsukuba.ac.jp	465	SSL/TLS	自動検出

- 手引き4.2.6節に従い、SquirrelMailでメールの送受信を行う

演習（3）

- ~/.forwardファイルを作成し、メールを残しつつ、自分の他のメールアカウントに転送する
  - 携帯メールはフィルタリングされることがある。tsukuba.ac.jpからのメールは受信できるようにする
- 実験に成功したら~/.forwardを以下で削除する

  $ rm .forward
- 削除後、転送されなくなったか確認する

演習（4）

- 手引き4.2節に従い、Thunderbirdで全学メールの設定、メールの送受信を行う
  - メールアカウント sXXXXXXX@u.tsukuba.ac.jp

	プロトコル	サーバ	ポート	暗号	認証
受信	IMAP	mail.u.tsukuba.ac.jp	993	SSL/TLS	通常のパスワード認証
送信	SMTP	mail.u.tsukuba.ac.jp	465	SSL/TLS	通常のパスワード認証

- 共通科目「情報（実習）」手引き3.2節に従い、Active! Mailでメールの送受信を行う
- 緊急時一斉メールなど流れるため定期的に受信確認をすること。あるいはメール転送設定する
  - https://www.u.tsukuba.ac.jp/icho13/forward.html

オプション演習（1）

- メールサーバのport 25にTCPで接続する

```
$ nc violet-nwe.coins.tsukuba.ac.jp 25
```

- SMTPによりメールを送信する

```
MAIL FROM:<sXXXXXXX@coins.tsukuba.ac.jp>
RCPT TO:<sYYYYYYY@coins.tsukuba.ac.jp>
DATA
...
.
QUIT
```

オプション演習（2）

- 送信先メールサーバの調査する。送信先メールサーバはDNSのMXフィールドに登録されている

```
$ host -t mx coins.tsukuba.ac.jp
coins.tsukuba.ac.jp mail is handled by 10 smtpgwin.cc.tsukuba.ac.jp.
$ host -t mx docomo.ne.jp
docomo.ne.jp mail is handled by 10 mfsmax.docomo.ne.jp.
```

- 送信先メールサーバにSMTPで接続して、メールを送信してみよう

オプション演習（３）

- メールを送信するプログラムを書いてみよう。プログラミング言語はなんでもよい。できたらそれも提出しよう