最終更新　2019.6.30

# ネットワークのしくみ

## IPアドレスとホスト名

　図7.1のように、異なるコンピューター上（パソコン、タブレット、スマホは全てコンピューターです）で動作する2つのプログラムがネットワークを介して通信する場合を考えます。Webブラウザでネットを閲覧している場合はそれに該当します。パソコンやスマホ上で動作するプログラムは「アプリ」、Webサーバ上で動作するプログラムは「サーバープログラム」と呼ぶことが多いです。どちらも「プログラム」なので、ここでは単に「プログラム」と呼びます。



図.　ネットワークを介した通信

　電話による通信では、個々の電話機が「電話番号」という唯一の識別番号を持っています。通常、電話機は待ち状態です。電話をかける側が通話先の電話番号を指定して、呼び出すと、待ち受け側のベルが鳴り、受話器を取ると双方向の通信が確立します。

　ネットワークを介して2つのプログラムが通信する場合もこれと似ています。ただし、相手を特定するのに2つの要素が必要です。各々のコンピューターは「IPアドレス」で識別します。コンピューターの中には多数のプログラムが動作しています。個々のプログラムは「ポート番号」で識別します。

　サーバープログラムは常に待ち受け状態にあります。そのサーバープログラムに対して接続を試みるプログラム（ex. Edge, Chrome, Safari, Firefox）をクライアントと呼びます。

◆ IPアドレス

　個々のコンピューターはIPアドレスと呼ばれる固有の識別番号を持っています。



図7.2　self.exeの外観

　図7.2のself.exeは自マシンのIPアドレスなどを表示するプログラムです。IPアドレスは192.168.107.2のように、ピリオドで区切った4つの数値で表されます。各々の数値は1 byteで表されるので、0～255の値をとります。

　4バイトで表せる数の範囲は232 = 約43億です。この数はインターネットに接続されているコンピューターの総数に比べると少なすぎます。そこで、以下のような仕組みが用いられています。

　奈良教育大学の中には多数のコンピューターがありますが、外部から接続する必要があるコンピューターは少数です（Webサーバ、メールサーバなど）。そこで、奈教内のコンピューターを以下の2つに分けます。

(a) 外部から接続する必要があるコンピューター

(b) 外部から接続する必要がないコンピューター

　(a) は世界で唯一のIPアドレスを持ちます。奈教の場合202.236.176で始まるアドレスです。グローバルIPアドレスと言います。電話に例えると外線番号に相当します。奈教外から接続するときはこの番号を用います。

　電話の場合、奈教内の電話機同士は内線番号で相手を呼び出します。同様に、(a), (b) ともに奈教内だけで通用するIPアドレスを持ちます。プライベートIPアドレスと言います。192.168あるいは172.16で始まるアドレスです[[1]](#footnote-1)。皆さんの前のパソコンは (b) であり、外部から接続することはできません。

　学外から (a) のコンピューターに接続するときは、大学の入口でIPアドレスの変換が行われ、大学構内ではプライベートIPアドレスを用いて通信が行われます。

課題1

　本日の課題は隣り合った2台のパソコンにloginして行います。それぞれのパソコンでself.exeを実行し、設定されているIPアドレスの値を答えなさい。

　http://denki.nara-edu.ac.jp/~yabu/　にアクセスし、下方の「ブラウザのチェック」を押し、Webサーバが取得するIPアドレスの値を答えなさい。

◆ ホスト名

　IPアドレスは覚えにくいので、接続相手を指定するとき、通常はwww.google.comのような形式を使います。「www.google.com」のような名前を「ホスト名」と言います。

　インターネットは「IPアドレス」と「ホスト名」を相互変換する仕組みを持っています。ネームサーバと呼ばれるコンピューターが変換役を担います。ネームサーバは学校や会社など1つの組織内に最低1台は設置されています。ネームサーバはその組織が保有するコンピューターの「IPアドレスとホスト名」の相互変換のための対応表を保持しています。そして、世界中のネームサーバはお互いに通信して自分が保持している表を交換する仕組みを持っています。

　ネームサーバが「ホスト名 ←→ IPアドレス」の相互変換をする仕組みをDNS (Domain Name System) と呼び、ネームサーバーはDNSサーバーとも呼ばれます。

　「IPアドレス」と「ホスト名」の関係を調査するプログラムが図7.3に示すresolve.exeです。矢印が書かれたボタンを押すと変換を行います。1台のコンピューターは1個のIPアドレスを持ちますが、ホスト名は複数持つことができます（別名機能と言います）。



図7.3　resolve.exeの外観

課題2

　自分がloginしているパソコンのうち、1台について、ホスト名を答えなさい。

　133.3.140.24のホスト名を答えなさい。このホストは何という組織のものか。

　www.google.co.jpのIPアドレスを答えなさい。

課題3

　.jp .uk .fr .de .au .cn .kr はどこの国のドメインか。

 .co.jp .ac.jp .go.jp はどのような組織を表しているか。

## IPアドレスとポート番号による双方向通信

　「コンピューターAで動作するプログラムα」が「コンピューターBで動作するプログラムβ」に接続する場合を考えます。「コンピューターB」はIPアドレスで識別します。「プログラムβ」はポート番号と呼ばれる番号で識別します。

　個々のコンピューターは0～65535までの65536個のポートを持っています。

　プログラムβがポート番号xを使いたい場合、OSに対して「ポート番号xを使わせて下さい」という依頼をします。OSはその番号が現在使われていないなら、そのポート番号をプログラムβに与えます。

　ポート番号の0～1023番はWell known portと呼ばれ、予約されています。1024～49151番は登録済みポート番号です。個人が作成したプログラムはプライベートポート番号である49152～65535番を使うことが推奨されます。

　結局、異なるパソコンの中で動作している2つのプログラムが通信をするときは、図7.4のような感じになります。



図7.4　クライアントとサーバー

　クライアントは接続先のプログラムを［IPアドレス○○番のポート番号△△番］という形式で指定します。「ホスト名 → IPアドレス」の変換は先述したようにネームサーバーに問い合わせることにより解決します。特別なホスト名としてlocalhostを指定すると、自マシンを意味します。同じくIPアドレス127.0.0.1も自マシンを意味します。

　クライアント側のポート番号はOSが自動的にその都度割り当てます。

　2つのプログラムが通信することを確認するためのミニソフトウェアが図7.6の server.exeと図7.5のclient.exeです。server.exeはサーバをシミュレートするプログラム、client.exeはクライアントをシミュレートするプログラムです。



図7.5　server.exeの外観

◆ server.exeの使い方

1. 待ち受け用ポート番号を指定して「待ち受け開始」ボタンを押す。そのポート番号でクライアントからの接続を待ち受ける状態に入る。

2. 接続を受けたら「プログラム動作状況」に表示する。

 受信データを「受信文字列」に表示する。

3. データを送信したいときは、送信先ソケット番号を選び、「送信文字列」に文字を入力してEnterキーを押す。1行の文字列を送信する。クライアントのソケット番号は「受信文字列」のボックスに表示された番号を手がかりにして指定する。

◆ client.exeの使い方

1. 接続先ホスト名に接続先のホスト名を入れる。

2. ポート番号を設定する。

3. 「接続」ボタンを押すと、接続し、通信を開始する。受信したデータは「受信データ」に表示する。

4. 「送信データ」のウィンドウに送りたい文字を入れ、Enterキーを押す。Enterキーを押した時点でその1行を送る。下方に送信用バッファが2つある。「送信」ボタンを押すと、そのボタンの左側のエディットボックスの中の内容が送られる。あらかじめ送りたい文字列が分かっていて、素早く送りたい場合に使う。



図7.6　client.exeの外観

課題4

　パソコンAでserver.exeを2個起動しなさい。ポート番号は50001と50002に設定し、「待ち受け開始」ボタンを押します。

　パソコンBでclient.exeを起動し、パソコンAの2個のserver.exeと双方向の通信をしなさい（お互いに文字を送り、相手先の画面に表示されることを確認する）。通信している状況を示すため、パソコンAの画面のスクショをPrint Screenでとり、レポートに貼り付けなさい。

## NAPT (Network Address Port Translation) について

　奈教の入口にはルータと呼ばれるIPアドレスの付け替えをする装置があります。自宅にファイバーを引いた場合もONU (Optical Network Unit) が設置されますが、これもルータです。ルータでIPアドレスの変換が行われます。このしくみをNAPT (Network Address Port Translation: IPマスカレードとも言う）と言います。



図7.7　NAPTのしくみ

　図7.7は自宅に光ファイバーを引いて、ONUを置いているときの状況です。ルータより左側が自宅内のネットワークです。プライベートネットワークあるいはLAN (Local Area Network) と呼びます。右側はインターネットあるいはWAN (Wide Area Network) と呼びます。WAN側から見るとプライベートネットワーク中の全てのパソコンが202.232.5.16というIPアドレスを持った1台のパソコンに見えます。ルータがIPアドレスの付け替えを行います。

　インターネットにおける通信は、パケット通信と言って、情報をパケットと呼ばれる小さな固まりに分けて送ります。パケットを送る場合「最終的な送り先」と「次の送り先」を設定して送信します。例えばパソコンAが外部のWebサーバ133.42.3.3と通信する場合は「最終的な送り先は133.42.3.3, 次の送り先は192.168.1.1」です。パソコンAが内部のWebサーバーパソコンCと通信する場合は「最終的な送り先は192.168.1.100, 次の送り先も192.168.1.100」です。次の送り先がプライベートネットワーク内の場合は、直接送ります。そうでない場合は、次の送り先はルータ（この場合は192.168.1.1）です。

　あるIPアドレスが、自マシンが所属するプライベートネットワークの内か外かを判別するのが、サブネットマスクです。この場合、255.255.255.0なので、上位の24 bitに1が立っています。このことは、上位24 bit が共通なアドレスであるマシン（すなわち192.168.1.のアドレスを持つマシン）はプライベートネットワーク内にあることを意味します。

課題5

　あるネットワークでサブネット255.255.255.0であった。そのサブネット内の192.168.1.10のマシンが以下のIPアドレスと通信する。A, B, C, Dのマシンはサブネット内か外かどちらかを答えなさい。

 A 192.168.1.20

 B 192.168.2.20

 C 192.200.1.20

 D 192.200.2.20

　通常はWAN側からプライベートネットワークの中のパソコンに接続することは出来ないのですが、図7.7のようにプライベートネットワークの中にWebサーバのように外からの接続を受け付けるコンピューターがある場合（192.168.1.100のコンピューター）は、そのコンピューターに固定IPアドレスを割り当てます。この場合は、外から202.232.5.16の80番ポートへのアクセスがあった場合、192.168.1.100の80番ポートへ転送します。

　そうでないコンピューターはDHCP (Dynamic Host Configuration Protocol) というしくみを使って、IPアドレスは自動的に割り当てられます。パソコン, スマホはネットワークに接続するときに、DHCPサーバからIPアドレスを割り当てられます。DHCPサーバはサブネットに1台あります。通常はルータがDHCPサーバを兼ねます。

## ポート番号とプロトコル

　コンピュータ（パソコン、スマホ）の中では多数のプログラムが同時に動作しています。その中にはネットワークを通して他のプログラムと通信を行うものがあります。イメージ図を図7.8に示します。



図.8　ネットワーク通信のイメージ

　クライアント側の代表的なプログラムはWebブラウザ（ex. Edge, Chrome, Safari）です。サーバ側の代表的なプログラムはWebサーバ、メールサーバなどです。

　通信を行うプログラムは相手を識別するのにIPアドレスとポート番号を使用します。ポート番号のうち、0～1023番はWell known portと呼ばれ、用途が決まっています。有名なポート番号として表7.1のような番号があります。

表.　代表的なポート番号

|  |  |  |
| --- | --- | --- |
| ポート番号 | プロトコル名 | 用途 |
| 20, 21 　 | 　ftp | 　ファイルの送受信 |
| 22 　 | 　ssh | 　遠隔ログイン（暗号化付き） |
| 23 　 | 　telnet | 　遠隔ログイン |
| 25 　 | 　smtp | 　メール配送 |
| 80 　 | 　http | 　Web |
| 110 　　　 | 　pop3 | 　メール受信 |
| 114 　 | 　imap4 | 　メール受信 |
| 115 　 | 　sftp | 　ファイルの送受信（暗号化付き） |
| 443 　 | 　https | 　Web（暗号化付き） |
| 587 　 | 　smtp AUTH | 　認証付きメール配送 |

　ここではhttpとsmtpについて学習します。各プロトコルの詳細は以下に分かりやすい解説があります。

http://www.atmarkit.co.jp/fnetwork/rensai/netpro00/netpro01.html

## http (hypertext transfer protocol)

　2019年現在、Webサーバとの通信はほとんどの場合、httpを暗号化したhttpsプロトコルが使われます。ここでは、しくみを学習するため、簡易なhttpプロトコルを扱います。httpはWebブラウザがWebサーバと通信するためのプロトコルです。まずは、Webブラウザで次のページを開いて下さい。

http://denki.nara-edu.ac.jp/~yabu/edu/jyoho/html/simple.html

　ブラウザのurl（Uniform Resource Locator）入力欄に上のようなアドレスを打つと、ブラウザは次のような動作を行います。

1. コンピューター denki.nara-edu.ac.jp の80番ポートに接続する。

2. ファイル /~yabu/edu/jyoho/html/simple.html を要求する。

3. 取得した内容を表示する。

　Webブラウザが行っている作業を手作業で行ってみましょう。汎用通信プログラムclient.exeを使います。Webサーバに接続した後、以下のような文字列を送信します[[2]](#footnote-2)。

GET /~yabu/edu/jyoho/html/simple.html HTTP/1.0

　　（空行）

　GET はファイルを取得する命令です。取得したいファイル名をその後に書き、使うプロトコルのバージョンを書きます。GETを含む行の次に、空行を1行送ることに注意して下さい。空行は命令の終わりを表します。次のような文字列が送られてくるはずです。

HTTP/1.1 200 OK

Date: Fri, 08 May 2009 04:30:09 GMT

Server: Apache

Last-Modified: Thu, 02 Oct 2003 03:04:29 GMT

ETag: "75295b-67-81932940"

Accept-Ranges: bytes

Content-Length: 103

Connection: close

Content-Type: text/html

<html>

<head>

<title>The simplest page</title>

</head>

<body>

This is a simplest page.

</body>

</html>

　Webブラウザはこの文字列を受け取って、内容を画面に表示します。ブラウザで「表示」→「ソース」と操作すると、<html>から下の部分を見ることが出来ます。ここでは簡略化のため、client.exeからWebサーバに送る文字列は1行だけにしましたが、実際のWebブラウザは「使っているOS, ブラウザのアプリ名, 直前に見たページのurl, クッキー」などをWebサーバに送っています。

　ブラウザからGET .... という命令を送ると、サーバは要求された内容を送ってきます。このような、通信をするときの取り決めをプロトコルと言います。

　Webサーバは1回通信すると、コネクションを切断してしまいます。従って、Webサーバに対して何かをGETするときは、その都度、接続を要求する必要があります。「汎用通信プログラム」の「接続」ボタンは毎回押してください。

　GETは指定したファイルを取得する命令でした。httpプロトコルの主な命令は2つしかありません。1つはここで紹介したGETであり、もう1つはブラウザからWebサーバへ大きなデータを送るときに使われるPOSTです。

課題1

(1) 上記の例において、サーバから送られてきた文字列の中に、以下の文字列がある。何を表しているか。

Last-Modified: Thu, 02 Oct 2003 03:04:29 GMT

(2) server.exeを起動し80番ポートを開けて待機します。すなわちserver.exeはWebサーバのふりをすることになります。Edgeなどのブラウザを起動し、http://localhost/ へアクセスして下さい。localhostは自マシンを意味します。IEがWebサーバに送信した文字列をコピペして示しなさい。

　どのような情報が送られていますか？

　次にserver.exeにおいて、送信先ソケット番号をブラウザのソケット番号に設定し、以下の文字列を送りなさい。

HTTP/1.0 200 OK

Content-Length: 3

（空行）

abc

　結果として得られたブラウザの画面をコピペしなさい。

　操作をやり直す場合は、ブラウザの「再読み込み」のボタンをクリックして下さい。

(3)「薮哲郎のホームページ」の中にある「ブラウザのチェック」と書かれたリンクをクリックすると、「サーバが受け取る情報を表示します。例えば、HTTP\_REFERER = xxxxxx という表示は、= の左側が変数名、右側がその内容です。

　ログインしているパソコンのブラウザからクリックしなさい。また、自分のスマホからもアクセスしなさい。

　「パソコンからのアクセス」「スマホからのアクセス」のそれぞれの場合について、変数名を列挙し、各変数の内容が何を表しているか述べなさい。

　例えばYahooのサイトに行くと、パソコンから行くかスマホで行くかで表示が異なります。接続元のパソコンのOSによって表示するページの体裁を変えていると思われます。

## ネットワークを使うアプリについて

ブラウザか専用アプリか

　スマホからGmailを使うとき、Safariなどのブラウザではなく、Gmailというアプリを使っていると思います。同様に、Yahooのニュースを見るときもブラウザではなく「Yahooニュース」という専用アプリを使うことが多いと思います。以下のように使い分けていると思います。

* パソコンからアクセスするときはブラウザを使う
* スマホから使うときは専用アプリを使う

　スマホは画面が狭いので、ブラウザでは見づらい画面になってしまいます。サーバ側で、アクセス元がパソコンかスマホかを判別し、表示を変更すればいいと思われます。しかし、多くの場合、スマホには専用アプリが用意されています。その方が操作性が良いからでしょう。

httpsか独自プロトコルか

　「Gmail」「Yahoo天気」「Yahooニュース」などはブラウザから使うこともできます。すなわち、サーバとのやりとりはhttpsプロトコルを使っていると思われます。

　一方でLineはブラウザからはアクセスできず、パソコン・スマホ共に専用アプリが必要です。アプリとLineサーバとの間で、独自プロトコルによるデータの送受信が行われていると思われます。

ピア・ツー・ピアかサーバ経由か

　「Webサーバ」と「Webブラウザ」の間のやりとりは、2つのコンピューター間で直接行われます。これを、ピア・ツー・ピアと言います。一方で、2人だけのグループであってもLineでやりとりする場合は、各クライアント（アプリ）はLineサーバとやりとりします。クライアント同士はやりとりしません。

## smtp (simple mail transfer protocol)

　今ではメールの読み書きはWebブラウザを通して行うことが多いと思いますが[[3]](#footnote-3)、それ以前はThunderbird, Outlookなどのメーラ（メールを読み書きするアプリ）を使ってメールを読み書きしていました。図7.9のような感じになります。メールの配送はsmtpサーバが担当します。受け取るときはpop3サーバを使います。



図7.9　メーラを使ったメールの読み書き

　奈教のActive Mailのように、Webメールを使う場合は、ブラウザを使ってメールを読み書きします。この場合のしくみは図7.10のようになります。この図ではメール読み書き用Webサーバとメールサーバを別に描いていますが、同一のコンピューターかも知れません。メール読み書き用Webサーバの向こう側は、使い手には分かりません。



図.　Webメールのしくみ

　図7.10ではsmtpサーバは587番ポートを使っています。昔は、メールは暗号化せずに送っていました。暗号化しない場合は25番ポートを使います。smtp認証のしくみはhttp://www.kanadas.com/investigation-j/2007/04/smtp.htmlで分かりやすく説明されています。

　25番ポートを使う場合のメールクライアントとメールサーバ間の通信例を表7.2に示します。このメールの送信プロトコルをsmtp (simple mail transfer protocol) と言います。サーバからの応答250はack (acknowledge) を表します。smtpの命令表を表7.3に示します。

表7.2　メールの送信時のやりとり

|  |  |
| --- | --- |
| メーラ | メールサーバ |
| HELO machine-nameMAIL FROM:<xxx@nara-edu.ac.jp>RCPT TO:<yyy@nara-edu.ac.jp>DATAメールの1行目メールの2行目 . (ピリオドが終端を表す)QUIT | 220 mail2.nara-edu.ac.jp ESMTP(以下略)250 mail2.nara-edu.ac.jp250 2.1.0 Ok250 2.1.5 Ok354 End data with <CR><LF>.<CR><LF>250 2.0.0 Ok: queued as F0A0F22FCBD221 2.0.0 Bye |

表7.3　SMTP命令表

|  |  |
| --- | --- |
| 命令 | 意味 |
| HELO | サーバに自分自身の正体を知らせる |
| MAIL FROM: | 送信側のメールボックスを知らせる。宛先不明などの時のエラーメッセージはこのアドレスへ返される。アドレスは < > で囲む |
| RCPT TO: | 宛先のアドレスを指定する。アドレスは < > で囲む |
| DATA | メッセージを送る準備が出来たことをサーバに知らせる。返答コードが354のときデータを送信することが出来る。 |
| QUIT | 通信を終了する |

　本実習では表7.2のやりとりを手作業で行います。

　HELO, MAIL FROMなどの命令を小文字で入力してもエラーにはなりませんが、MAIL FROM: と RCPT TO:　のコロンの直前に空白を入れるとエラーになるので注意して下さい。

　手紙に例えると、封筒に書いた宛先と差出人が「RCPT TOとMAIL FROM」に該当します。封筒の中に入った便箋に書いた内容が「DATAからピリオドまでの部分」に該当します。便箋に書いた内容に相当する部分を「本文」と呼びます。

　e-mailの本文は「このような形式にしなさい」というルールがあります。そのルールはRFC822と名付けられています。それによるとe-mailの本文は「ヘッダブロック」と「メッセージブロック」の2つのブロックからなり、2つのブロックは空行（CR LF）で区切るというルールがあります。

　ヘッダブロックの中には差出人を表すフィールド（行）や件名を表すフィールドなどを書きます。ただし、本校のメールサーバの場合、ヘッダブロックを省略してもメールを受け付けてくれます。本文の例を示します。

From: Tetsuro Yabu <yabu@略.ac.jp>

To: (Tetsuro Yabu) yabu@略.ac.jp

Subject: Test mail

Reply-To: hanako@mydomain.com

X-My-header: dokuji no jyouhou

Test mail.

last line.

　最初の5行がヘッダブロックで、最後の2行がメッセージブロックです。この例では、ヘッダブロックに5個のフィールドがあります。1個目のフィールドの「From:」の部分をヘッダと言います。ヘッダは「:（コロン）」までの文字列で、1桁目から始まります。行頭にスペースを入れてはいけません。スペースかタブから始まる行は継続行とみなされます。ヘッダの大文字と小文字は区別されません。アドレスを表すフィールドにおいては、(　　) で囲んだ部分はコメントと見なされます。また、< > で囲んだ部分がある場合、その中がアドレスで、それ以外はコメントと見なされます。

　From: の後に差出人のアドレス、To: の後に宛先のアドレス、Subject: の後に件名を書きます。

　受信したメールに対して「返信」の操作をした場合、ヘッダブロックの中にReply-To: というヘッダが存在する場合はそのアドレスを返信先アドレスに設定し、無い場合はFrom: アドレスを返信先アドレスに設定します。上のメールに対して返信しようとすると、hanako@mydomain.com が返信先アドレスに設定されます。メーリングリストを運用する場合などにReply-To: フィールドは使われます。

　希に開封確認を求めるメールがあります。Disposition-Notification-To: というヘッダがある場合、メーラは指定されたアドレスへ開封確認を送って良いかどうかユーザーに問い合わせます。X- で始まるヘッダは個々のメーラが独自に設定したヘッダです。

課題6

　各小問においてアンダーラインが引かれた部分に答えなさい。

(1) 表7.2の手順に従って、「汎用通信プログラム」を用いて自分自身のアドレスにメールを送りなさい。奈良教育大学の学生用のメールサーバのホスト名とポート番号は

student.nara-edu.ac.jp　　25

です。アドレスは

xxxx@student.nara-edu.ac.jp

です。HELOコマンドの後に名乗るホスト名は、本来なら現在操作しているコンピューターのホスト名を入れるべきですが、本校のメールサーバの場合、何を入れても受け付けてくれるようです。MAIL FROMコマンドの後のアドレスはエラーが発生したときにエラー発生を通知するメールを送る時の送り先などに使われます。本校のメールサーバの場合、学内から出す場合は何を入れても受け付けてくれるようですが、一般のプロバイダではFrom: フィールドと異なるアドレスに設定するとメール配送を拒否されると思われます。

　本文として test mail という一行だけのメール（すなわち、ヘッダブロックは存在しない）を自分自身に送って下さい。

　Active Mailで確認して下さい。Active Mailは送られてきたメールをそのまま表示するのではなく、読みやすい形に整形して表示します。送られてきたメールをそのまま表示するには、右端の「▼操作を選択」→「ソース表示」です。

　今回送ったメールはヘッダブロックなしのメールでしたが、「ソース表示」をすると、ヘッダブロックがあり、多数のフィールドがあります。このヘッダブロックはメールサーバが自動的に付加したものです。メールサーバによって自動的に付加されたヘッダにはどのようなものがありますか？　ヘッダ名（: より左側）を全て書きなさい。このうち、メールの配送経路を表すヘッダ名はどれですか？

(2) この小問以降は図7.11のmail.exeを用います。「宛先」「メールサーバ名」「差出人」「送信内容」を記入して、「メール送信」ボタンを押して下さい。宛先は自分自身にします。



図7.11　mail.exeの外観

　ヘッダブロックを以下のように書いて下さい。

　　From: aaa <aa@bb.cc>

この通りでなくても良い

　　To: bbb <xx@yy.zz>

　　Subject: ccc

　差出人を表す From: と宛先を表すTo: にわざと間違った名前とアドレスを書いています。本来は From: の後には MAIL FROM: で指定したアドレス、To: の後にはRCPT TO: で指定したアドレスを書くべきです。この場合、受信したメールの「件名」「送信者」はどうなりますか？

＜注意＞

　奈良教育大学のメールサーバはヘッダブロックの From: と送信時の MAIL FROM: が異なっていてもメールを受け付けてくれますが、一般のプロバイダではそのようなメールは拒否されます。

(3) ヘッダブロックの中に

　　Reply-To: nazo no hito <nazo@himitu.com>

というフィールドを含めて自分自身にメールを送って下さい。そのメールに返信しようとすると、返信先のアドレスはどこになりますか。また、そのフィールドを入れない場合、返信先のアドレスはどこになりますか。

　先述のように、手紙に例えると、サーバとやりとりするときのMAIL FROM: や RCPT TO: は封筒に書いた宛先と差出人です。メールの配送はそれに従って行われます。

　DATA以下の部分は封筒中の便箋です。便箋中にFrom: や To: を書きますが、便箋中のFrom: と封筒に書かれたMAIL FROM: が一致していなくてもメールは配送されます。

　そして受取人が入手するのは封筒の中身（便箋）のみです。封筒は受取人には届きません。便箋に書いてある「差出人、受取人」は「真の差出人、真の受取人」と異なっている場合があります。

　怪しいメールが来たときは、メールのヘッダを表示して、配送経路、From: To: を確認して下さい。

1. 192.168はクラスC（サブネットマスクが255.255.255.0）、172.16～172.31はクラスB（サブネットマスクが255.255.0.0）のプライベートネットに使われます。 [↑](#footnote-ref-1)
2. ここでは単純なHTTP/1.0プロトコルを用いました。通常のブラウザはHTTP/1.1プロトコルを用います。HTTP/1.1はHOSTヘッダなどが必須となり、タイプ量が増えます。 [↑](#footnote-ref-2)
3. Webメールはどこででも読めるというメリットがあります。Webメールを使う場合、ユーザーが接続するのはWebサーバであり、その向こうがどうなっているかは、ユーザーには見えません。このあたりについては、私もよく分かりません。 [↑](#footnote-ref-3)