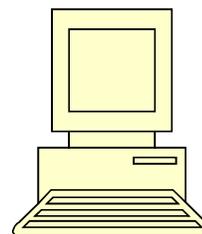
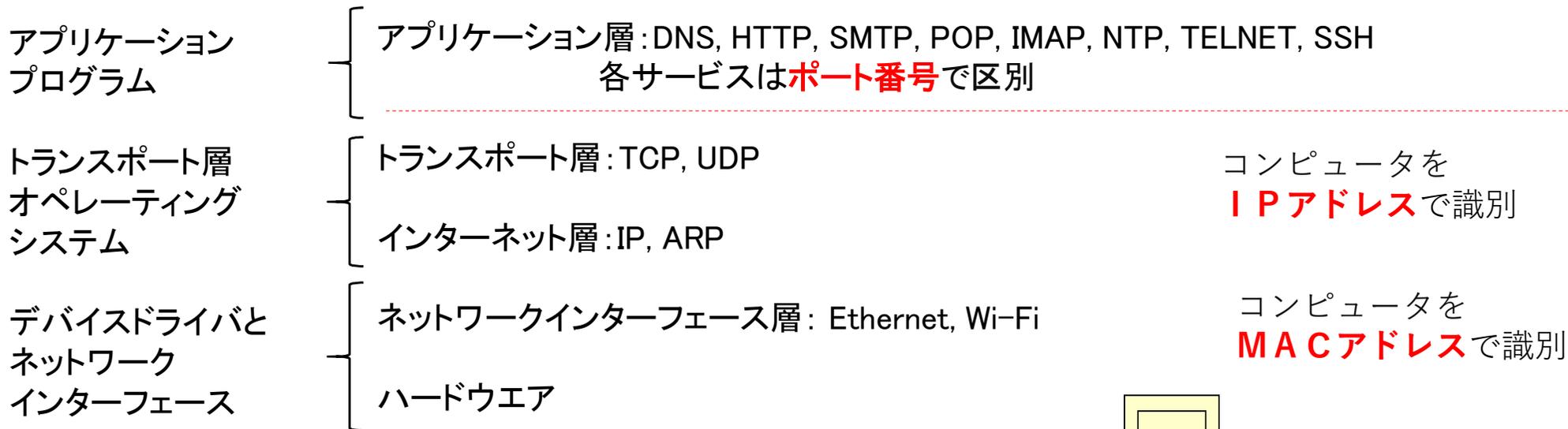


船舶海洋情報学

03. ポート番号とサービス・プロトコル



TCP-IPプロトコルの階層



133.5.136.77
通称メールサーバ
System.nams.kyushu-u.ac.jp
SMTPサーバ(郵便ポスト)**25番ポート**
POPサーバ(郵便受け)**110番ポート**

「IPアドレス」でコンピュータを区別
さらにサービスも区別せねばならぬ →

各コンピュータにサービス受付の「**ポート番号**」

TCPの代表的なポート番号とサービス

WWW (World Wide Web)

・80番 http World Wide Web HTTP
・443番 https http protocol over TLS/SSL 暗号化

電子メール

・25番 smtp Simple Mail Transfer Protocol 郵便ポスト=メールを出す
・109番 pop2 Post Office Protocol Ver.2 郵便受け
・110番 pop3 Post Office Protocol Ver.3 郵便受け
・995番 pop3s pop3 protocol over TLS/SSL 郵便受け(暗号化)
・143番 imap Internet Message Access protocol v2,v4 郵便受け
・220番 imap3 Internet Message Access protocol v3 郵便受け
・993番 imaps imap4 protocol over TLS/SSL 郵便受け(暗号化)

コンピュータの時刻合わせ
ただしバイナリデータでやりとり

・123番 ntp Network Time Protocol 通常はUDPで通信

遠隔ログイン

・23番 telnet 遠隔ログインProtocol
・22番 ssh SSH遠隔ログインProtocol 暗号化

ネームサービス

・53番 DNS Domain Name Server

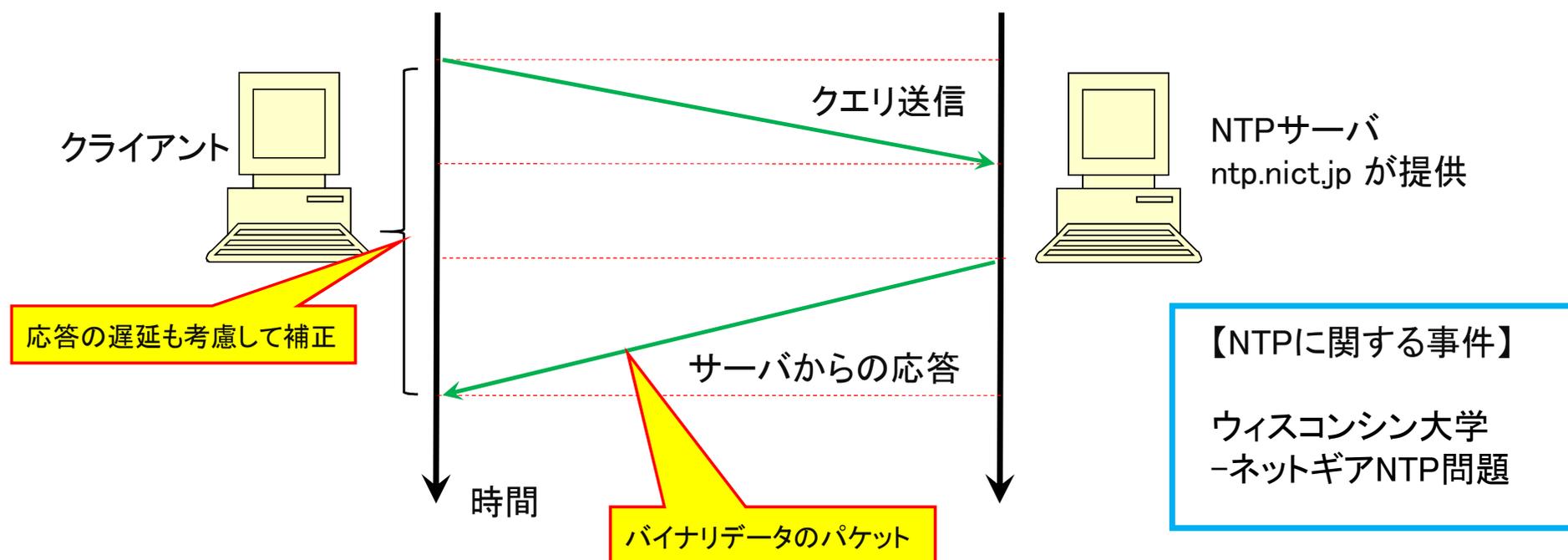
その他:ファイル転送プロトコル ftp など

NTP (Network Time Protocol)

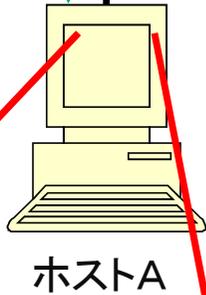
ホストの時計を正確な時刻へ自動的に合わせるための通信プロトコル(UDPまたはTCP123番ポート)

複数のホストでファイルを操作するとき、時刻情報が不正確では深刻な混乱が発生
ネットワークへ接続されているホストの時計を正しい時刻へ同期する

NTPサーバはDNSと類似した階層構造をとることが可能だが、階層の深さは最大15まで



遠隔ログイン(TELNET と SSH)

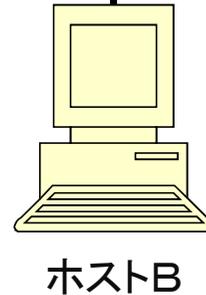


ホストA

TeraTermなど多数のソフトがある

TELNETクライアント:
TELNETサーバの23番ポートへTCP接続

ホストAを操作しているのに、
あたかもホストBの前でコンソールを
操作しているかのようにホストBを
利用できる



ホストB

TELNETサーバ
TCP23番ポート
他のホストからの接続を受付
(アカウント・パスワード管理)
それ以外は通常のコンソール
(DOS窓・コマンドプロンプト)
と同じ動作



```
delete
quit
Connection closed.
C:\Home\Gen\public_html\kajivara\lab\Medu\船船海洋情報学\telnet_java>java telnet
localhost 5677
hostname = localhost
TCP-IP port number = 5677
Connected to localhost port 5677
stdin ready
receiver ready
Type "quit" when you want to quit this program.
list
リストを表示します
delete
ABC!
quit
Connection closed.
C:\Home\Gen\public_html\kajivara\lab\Medu\船船海洋情報学\telnet_java>cd ..
C:\Home\Gen\public_html\kajivara\lab\Medu\船船海洋情報学>cd telnet
指定されたパスが見つかりません。
```

```
delete
quit
Connection closed.
C:\Home\Gen\public_html\kajivara\lab\Medu\船船海洋情報学\telnet_java>java telnet
localhost 5677
hostname = localhost
TCP-IP port number = 5677
Connected to localhost port 5677
stdin ready
receiver ready
Type "quit" when you want to quit this program.
list
リストを表示します
delete
ABC!
quit
Connection closed.
C:\Home\Gen\public_html\kajivara\lab\Medu\船船海洋情報学\telnet_java>cd ..
C:\Home\Gen\public_html\kajivara\lab\Medu\船船海洋情報学>cd telnet
指定されたパスが見つかりません。
```

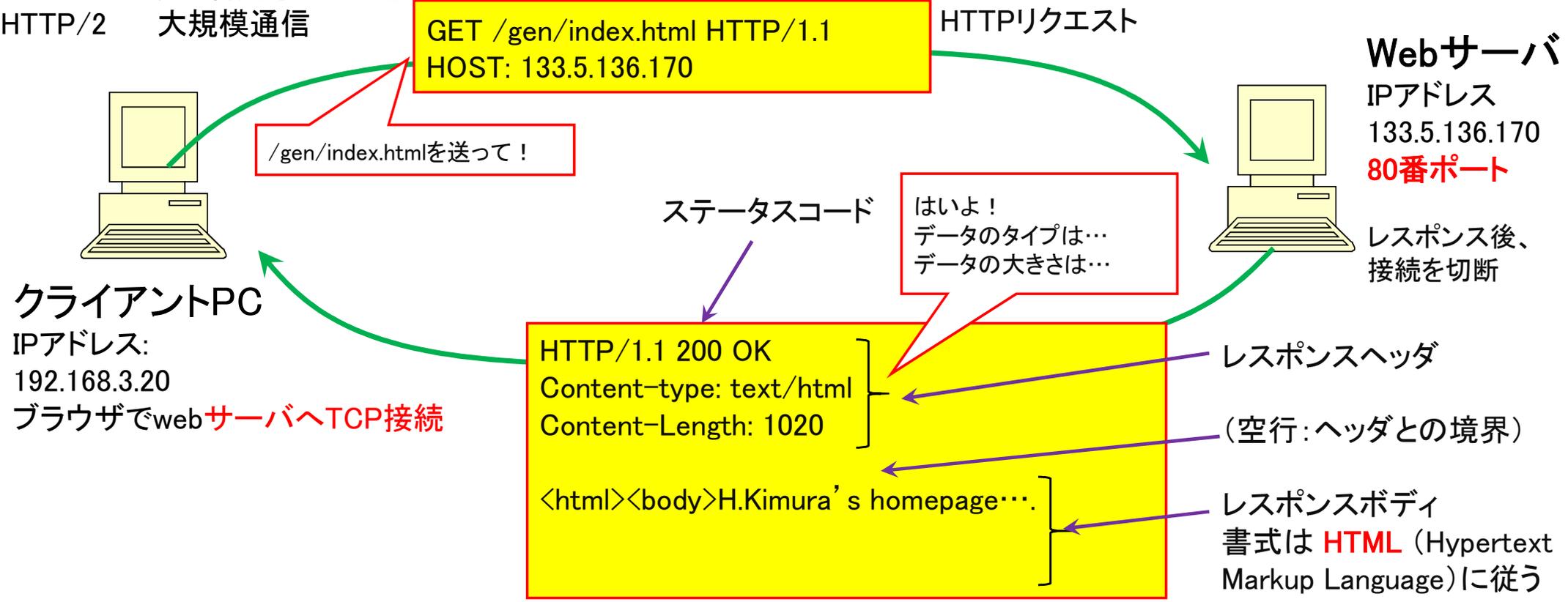
TELNETは、アカウントやパスワードも含めた全てのパケットが
暗号化されずにそのままネットワークを流れていくため、**セキュリティ上危険な通信**

SSHは、これらの**通信を暗号化**

World Wide Web (WWW)

HTTP (Hypertext Transfer Protocol)

HTMLなどのコンテンツの送受信に用いられるリクエスト-レスポンス型の通信プロトコル
ポート番号80をデフォルトとして使用
HTTP/1.1 広く利用されている
HTTP/2 大規模通信



HTTPリクエスト(メソッド)の一覧

メソッド リソース名 バージョン

```
GET /gen/index.html HTTP/1.1  
HOST: 133.5.136.170
```

← HTTPヘッダ

GET	指定したリソースを取得する。日本語などの文字列は%xxのように16進数の文字コードで表現 URLの最後に?に続けて文字データをサーバに送ることができる
HEAD	指定したリソースのヘッダ情報だけを取得する
POST	指定したリソースに対してデータを送信する。データは「要求本文」にセットして送信する バイナリデータ(文字以外)も
PUT	指定したリソースに対してデータを送信して置き換える
DELETE	指定したリソースを削除する
CONNECT	プロキシにおけるトンネリング処理を行う
OPTIONS	利用可能なオプション(メソッド)の一覧を返す
TRACE	サーバの動作の診断をするための機能

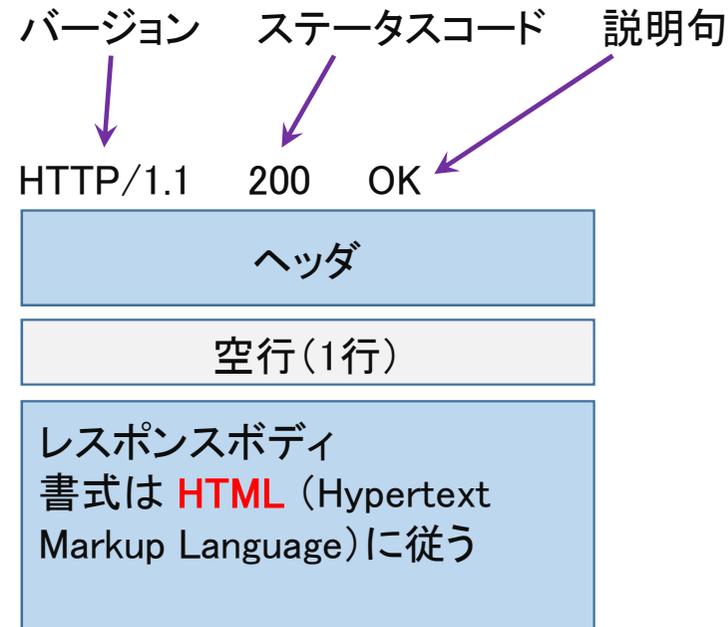
HTTPヘッダフィールドの一覧

- Accept 受信可能なメディアタイプを指定する。例:textやHTML、XML、画像形式など
- Accept-Encoding 受信可能なエンコーディングの種類(圧縮方法のcompressやgzip、deflateなど)を指定する
- Accept-Language 受信可能な言語を指定する。例:日本語版のWebブラウザならJa-JPなど
- Date コンテンツの作成日時を表す
- Expires コンテンツの有効期限を表す
- Last-Modified コンテンツの最終変更日時を表す
- Content-Type 送受信されるコンテンツの種類(メディアタイプ)を表す
- Content-Length コンテンツのデータ長
- If-Modified-Since 指定した日時以降に変更があれば、要求を実行する
- Transfer-Encoding データを分割して送信する場合の符号化方式などを指定する
- Range 送受信されるデータの範囲を表す。データ全体を渡すのではなく、部分的に送受信したい場合に利用する
- Location URLの移動先(リダイレクト先)などを表す
- Referer 参照元ページのURL情報を表す
- Origin 最初のページのURL情報を表す
- Host 接続しようとしているホストの名前を表す
- Server HTTPのサーバの情報を表す。例:「Microsoft-IIS/8.5」など
- User-Agent ユーザーエージェント(HTTPのクライアント)の名称やバージョンなどを表す
- Cookie HTTP Cookieのデータを表す
- Set-Cookie HTTP Cookieをセットさせるための指示を表す
- WWW-Authenticate 認証が必要な場合にサーバ側から返される「realm(レルム)」情報を表す
- Authorization 認証に関する情報(認証方法やユーザー名、パスワードなどの情報)を表す
- X-~ サーバやクライアントの実装に依存するような独自のヘッダフィールドは「X-~」という名前で定義して利用される

ウェブサーバとブラウザ間で状態を管理するしくみ

例) ショッピングサイトにおけるカートやログイン状態の管理

HTTPレスポンス



HTTPステータスコードの一覧

200	OK
202	Accepted(受理)
400	Bad Request(不正な要求)
401	Unauthorized(そのURLは認証されておらずアクセスできない)
403	Forbidden(アクセスは禁止されている)。ユーザー認証されればアクセスできる(可能性がある)
404	Page Not Found(指定されたリソースが見つからない、もしくはアクセスできない)。 応答本文として、エラー時に表示すべきHTMLページの内容が含まれていることがある
405	Method Not Allowed(禁止されているメソッド、もしくはサポートされていないメソッド)
408	Request Timeout(要求全体を受け取る前にタイムアウトした)
500	Internal Server Error(サーバの内部エラー)

SMTP (Send Mail Transfer Protocol)

- ・telnetというプログラム

(任意のIPアドレスの任意のポート番号へ**TCP接続**して文字列で通信を行うプログラム)
を起動し、IPアドレス 133.5.136.77 のポート番号25番へ接続する:

> Java telnet 133.5.136.77 25

- ・以下のように入力してあいさつ

HELO (こちらのホストマシン名:分からなければ適当あるいは省略可)
正常に送れば250という数字が返ってくる

- ・次にメール送り主である自分のアドレスを指定

MAIL FROM: <kimura@system.nams.kyusyu-u.ac.jp>
正常に送れば250という数字が返ってくる

- ・宛先のアドレスを指定

RCPT TO: <kimura.hajime.480@m.kyushu-u.ac.jp>

- ・本文を入力:

DATA ← まずこれ1行入力して改行し、英数字のみでメッセージ、
終了するにはピリオド1文字だけ入力して改行

- ・QUIT と入力して終了

サーバーと文字列を
やりとりするだけ!

TCP-IPクライアント／サーバによるお手軽IoTの例

「ネットワーク家電」や「IoT」等と呼ばれるものの正体はこのようなハードウェアとVPNなどのネットワーク技術、認証・暗号化などのセキュリティ技術、通信フォーマットなどを共通化する規格などの組合せから成る



クライアントPC
IPアドレス:
ブラウザおよびプログラムで
サーバへTCP接続

PICNIC (秋月電子通商)
ネットワークI/Oボード

TCP-IPでネットワークに接続し、ネットワーク上の別のPCのブラウザやプログラムにより上記I/Oボードの入力や出力を制御



I/Oサーバ
IPアドレス
192.168.0.190
HTTPポート: 80番
LCDポート: 10000番
パラレルポート: 10001番
シリアルポート: 10002番



リレー
モータドライバ
ブザー
リモコン
...

光センサ
磁気センサ
温度センサ
距離センサ
...

TCP-IPによるクライアント／サーバシステムの構築

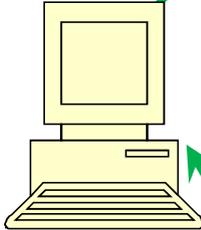
アプリケーションにおけるプログラムの世界とTCP/IPの世界を結ぶ特別な出入り口
= **ソケット(Socket)**

TCP通信を行うためのプログラミング言語の関数

クライアント側： IPアドレスとポート番号を指示して相手のホストに接続
サーバ側： ポート番号を設定して相手ホストからの接続を受け付ける
相手と接続したら、**ファイルを読み書きするのと同じ要領でデータを送受信**

TCP通信プログラムが簡単に作れる

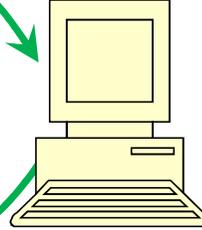
クライアントPC
自作プログラム



または
Telnetクライアント

クエリ文字列

サーバ
自作プログラム



レスポンス文字列

同一のポート番号に対し、同時に複数のホストからの接続を許可する場合(1対多)と、1つのホストとの接続中は他のホストの接続をブロックする場合(1対1)とがある

まとめ

(1) ポート番号でサービスやプロトコルを区別

DNS(53), NTP(123), HTTP(80), SMTP(25), POP, IMAP, TELNET(23), SSH(22)
このくらいは知っておこう

(2) TCP-IP接続によるクライアント-サーバシステム

(3) プログラム言語からのTCP-IPの利用: **ソケット(socket)**

レポート課題 : ●TELNETによりTCP接続してサーバに“HELLO”と文字列を送ると“Hi! I am ? ? ? ? .”と返答し、“Bye!”と文字列を送ると接続を切るサーバのプログラムを作り、telnet接続して動作を確認せよ。
? ? ? ?の部分にはレポート作成者の学籍番号と名前を出力するよう作成せよ。
●プログラム言語は問わない(Python か Java がお勧め C++でも良い)。
インターネット上に公開されている既存のプログラムソースコードに変更を加えれば良い。
「ソケット通信 プログラム Python」などのキーワードで検索せよ

【提出方法】

ソースコードおよびPython以外の言語の場合はコンパイル済みのバイナリをzipにまとめて圧縮し、

下記の課題提出用フォルダへ、課題の番号と提出者が分かるようにファイル名を以下のようにしてアップロードせよ
第3回2TE19xxxZ名前.zip

<https://share.iii.kyushu-u.ac.jp/public/hROwAAqIPI5ATI4BUXJtIJeJbMLJzVszfitNI89GHcMK>

上記フォルダへのリンクは下記ホームページから

<http://sysplan.nams.kyushu-u.ac.jp/gen/edu/NavalInformationProcessing/2019/index.html>