WWW ページが 見えるしくみ

子どもに教wantい 通信のしくみ

名古屋工業大学 菅原真司
Shinji Sugawara

まえがき

インターネット上で提供されるサービスの中で、一番の人気は何といっても世の中の様々な Web サイトを閲覧できる WWW（ワールドワイドウェブ）ではないでしょうか。少し意外ですが、このしくみはもともとスイスにある国際研究機関の CERN（欧州原子核研究機構：セルン）で、研究のための文献をヨーロッパに散らばる研究者間で共有するための文書管理システムとして作られました。今回は少しだけ WWW の秘密に迫ってみましょう。

ハイパーリンクと URL

WWW の特徴として真っ先に挙げられるのは、ハイパーリンクという考え方です。これは世界中に散らばっている文書を互に関連付けて、ある文書を読んでいるときに関連する別の文書が読みたくなったらそこから直接たどれるようなしくみです。まさに世界中の知識同士をつなげて、知識の蜘蛛の巣みたいにしてやろうというアイデアですね。WWW は World Wide Web が語源で、「世界中に広がった蜘蛛の巣」のような意味ですが、皆さんも、Web ページ上のある部分をクリックしたら別のサイトにジャンプするという経験を何度もされているでしょう。あれがハイパーリンクです。

さて、このしくみをつくるには、何が必要でしょうか。一番基本的なことは、世界中のどこからでも、特定の文書（文書に組み込まれた画像、音声情報なども含みます）を間違いなく指定することができる機能です。同じ指定をしても、時刻によって、場所によって、人によって、別の文書が取得されてしまうのでは困ります。そうならないために、WWW では文書の場所の特定と取得方法をセットにした書式を使います。これを URL（Uniform Resource Locator）といいます。これは一般には、後で説明する「スキーム名」と文書をしまっている場所の具体的な名前をの組合せから出来上がっています。

URL のしくみ

図 1 を見て下さい。これは典型的な URL で、本会のトップページを指定しています。ここで、最初の「http」というのがスキーム名で、この場合は HTTP という通信の取決めに従って文書にアクセスすることを表しています。

http:// www.ieice.org /jpn/index.html
スキーム名 オーソリティ パス

図 1 URL の構成

HTTP のような通信の取決めのことをプロトコルといいます。HTTP は、主に WWW 上の文書をコンピュータ間で転送するのに利用される代表的なプロトコルです。

スキーム名はコロン「:」で区切られ、その後にオーソリティと呼ばれる部分の開始を表す「//」が来ます。オーソリティというのは、多くの場合アクセスしようとしている文書を持つサーバ（情報を提供しているコンピュータ）のインターネット上での住所を含んだ名前になります。更に、もっと詳しくアクセスする文書の名前と場所を特定する必要がある場合は「/」で区切って書き込みます。これをパスと呼びます。

以上のことをまとめると、図 1 の URL は、「www.ieice.org というサーバの中にある jpn という場所の中の index.html という文書を、HTTP というプロトコルで転送してね」という意味になります。

HTTP の少しだけ詳しいしくみ

さて、それでは一体 HTTP とはどんな通信の取決めなのでしょうか。

WWW のしくみは、お客さん（クライアント：client）と、そのお客さんに奉仕する人（サーバ：server）の関係によく似ています。このようなシステムのことをクライアントサーバ型システムと呼んだりします。ユーザはインターネット上で情報を提供する WWW サーバに対して欲しい文書を要求し、サーバはその要求に応じて、該当する文書を転送します。まさにクライアントとサーバの関係ですね。実は HTTP はこのような動作を実現してくれる手順で、基本的な部分はとてもシンプルにできています。これを、図 2 を使って説明しましょう。

HTTP による WWW サーバとのやり取りは、前述の URL を、IE（Internet Explorer）や Safari、Firefox などのようなブラウザに指定した時点から始まります。このブラウザが事実上、WWW システムのクライアント（Web クライアント）に相当します。Web クライアントは、URL の中からスキーム名を見て HTTP での通信であることを確認すると、同じ URL 内に指定されている接続先サーバの名前を取り出します（手順①）。そのサーバの名前はインターネット上の住所も含んでいましたから、そのサーバへ接続して、転送して欲しい文書のリクエスト（HTTP リクエスト）を送信します（手順②）。このときに送られるメッセージが、図 3 になります。これは先の図 1 の URL に対応したものです。

ここで、最初の GET というのは、メッセージのメソッドといわれる部分で、「文書を転送して下さい」という意味に

図2 HTTP プロトコル

GET /jpn/index.html HTTP/1.1
メソッド リソース名 バージョン

図3 URL の構成

なります。その後には、転送を要求している文書の名前と、その文書のサーバ上の格納場所を一度に表現した /jpn/index.html が並び、空白のあと、最後に、HTTP のバージョン情報が続きます。図3の例ではHTTPのバージョンは1.1になっていますね。ここまでがWebクライアントの動作です。

今度は上記のメッセージを受け取るWebサーバ側の動作を説明しましょう。メッセージを受信（図2手順③）した後、クライアントが要求してきた文書を準備（手順④）できれば、返信（HTTPレスポンス）はとても簡単です（手順⑤）。HTTPレスポンスの例を図4に示します。

まず1行目でHTTPのバージョンとステータスを送ります。ステータスとはこの場合、「進行状況」や「状態」のことで、サーバがその時点でどういう状態にあるかを意味します。その状態はコード化して表現（3けたの数字にしている）されており、ステータスコードと呼ばれます。ステータスコードの後ろには、その簡単な説明句が続きます。この例ではステータスコード200でリクエストが成功した（OK）ことを伝えています。ステータスコードの部分では、200以外には、例えば301 Moved Permanently（文書のアドレスが恒久的に移動している）、401 Unauthorized（認証が必要なので送れない）、404 Not Found（文書が見つからない）があります。

次に、2行目からヘッダの部分になり、文書のタイプや、応答日時などの細かな情報が複数行にわたって表示されます。そして最後にヘッダから1行空けてボディ部分として、クライアントの要求文書本体が続きます。

クライアントは、HTTPレスポンスを受け取り（図2手順⑥）、その内容を解釈します（手順⑦）。必要な文書がそろえばそれで手順は終了です。しかし、受け取った文書の中に更に別の文書を埋め込む必要があるような場合は継続して必要な文

図4 HTTP レスポンス

書を取りに行くため、最初の手順①に戻ります。例えばある文書を転送してもらったら、その中には画像が何枚も含まれていることになっていて、その画像データは別のファイルとしてサーバにしまっていることが分かったときなどが、これに該当します。

HTTP の機能拡張

一つの文書に多くの画像が別ファイルとして埋め込まれているような場合、古いバージョンのHTTPでは、上記の例のように、一つのファイルを転送してもらう度に手順①に戻り、最初からサーバへ接続し直す必要がありました。しかし、これではとても効率が悪いため、1回の接続で、幾つもの文書ファイルを続けて取り出せるように、機能の拡張が行われました。この機能を、キープアライブ（keep alive）と呼び、HTTP/1.1から利用できるようになりました。また、同バージョンでは、リソース圧縮といって、gzipやcompressなどのファイル圧縮方式を使って文書のサイズを小さくして送信することで、通信帯域を節約することも可能になりました。

お す び

さて、WWWページが見えるしくみについて、主に通信プロトコルの面から説明しましたが、いかがでしたでしょうか？

できることならば、ページの記述に使用されるHTMLという言葉のことや、ショッピングサイトを利用したときに、以前に買い物かごに入れた商品をなぜかずっと覚えていてくれる不思議な機能に関係する「クッキー」という技術のからくりなどについてもお話しできるともっと楽しかったかもしれません。でも、残念ながら紙面が尽きてしまいました。またいつかどこかでお話したいと思います。

文 献

- [1] ネットワークマガジン編集部編、すっきりわかった！Web技術、株式会社アスキー、2005。
- [2] 藤本 壱、これだけは知っておきたいWebアプリケーションの常識、技術評論社、2009。