

学習情報ネットワーク論 (第3回)

室田真男
大学院社会理工学研究科 人間行動システム専攻
murota@hum.titech.ac.jp

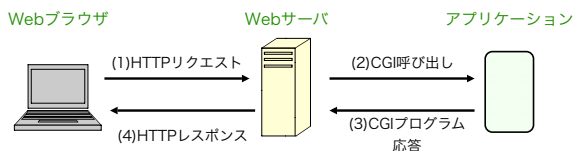
今日の内容 (2008年1月21日)

- World Wide Web
 - ▶ CGI (Common Gateway Interface)
 - ▶ Cookie
- 電子メール
 - ▶ メッセージフォーマット

2

CGIとは

- Common Gateway Interface
 - ▶ Webブラウザからの要求により、アプリケーションサーバ上のCGIプログラムを実行するためのインターフェイス



3

CGIにおけるデータ

- データの形
 - ▶ 変数名とその値の組
 - ▶ サーバへは符号化されて送られる
 - 名前=値&名前=値&名前=値...
- クライアントがサーバへ送る
 - ▶ GETメソッド
 - URIの一部として
 - ▶ POSTメソッド
 - サーバに送られるメッセージ本体のデータとして

4

GETによる方法

- 仕組み
 - ▶ 変数名と値の組がURIに追加される
 - 例: <http://example.org/cgi-bin/prog?keyword=Tokyo>
 - ▶ 環境変数に格納されてCGIプログラムへ
- 長所
 - ▶ URIとして残せる
 - ブックマークできる, ブラウザの再読込が有効
- 短所
 - ▶ 環境変数の長さに制限がある場合がある
 - ▶ 複雑なデータメッセージを送れない
 - ファイルのアップロードなども不可

5

POSTによる方法

- 仕組み
 - ▶ HTTP要求の後にデータとして転送される
 - ▶ データは標準入力へ渡される
 - ▶ データサイズはHTTPヘッダのContent-lengthで通知される
- 短所
 - ▶ URIとして残せない
 - ブックマークできない, ブラウザの再読込ができない
- 長所
 - ▶ 送れるデータの長さに制限がない
 - ▶ 複雑なデータメッセージを送れる
 - ファイルのアップロードなども可能

6

CGIにおけるデータ (2)

- サーバがCGIプログラムへ送る
 - ▶ 環境変数 (GET)
 - ▶ 標準入力 (POST)
- CGIプログラムがサーバへ送る
 - ▶ 標準出力

7

CGIサンプルプログラム

- GETの例
 - ▶ 省略
- POSTの例
 - ▶ 省略

```
#!/usr/bin/perl -w
use strict;
$! = 1;
my $data;
my $key;
my $val;

print "Content-type: text/plain\n\n";
print "Environment Variables and their Values\n\n";
print "-----\n\n";
while (($key,$val) = each %ENV) {
    print "$key = $val\n";
}
$val = $ENV{CONTENT_LENGTH};

if (defined($val)) { # for POST method
    print "\n";
    print "Data sent from browser to the server\n\n";
    print "-----\n\n";
    while (read(STDIN, $data, $val)) {
        print $data;
    }
}
```

8

クッキー (Cookie)

- Webサイトがユーザの利用履歴などを記録する方法のひとつ
- クッキーの要素
 - ▶ 応答メッセージ内のcookieヘッダ (Set-cookie)
 - ▶ 要求メッセージ内のcookieヘッダ (Cookie)
 - ▶ ユーザ側でブラウザが管理するクッキーファイル
 - ▶ Webサイト側がサーバの裏で保持するデータベース

9

クッキーの基本的な動作

- クッキーの基本的な動作
 - ▶ ブラウザは、同じサーバに対してリクエストを送るときは、Cookie:ヘッダにより記憶していたクッキー情報を送り返す
 - ▶ サーバはブラウザに対して、Set-cookie:ヘッダを使い、クッキー情報を送る
 - ▶ サーバは、Set-cookieヘッダを用いて情報の更新もできる
 - ▶ クッキーは設定された寿命になるまでブラウザ内で保管され、何回でもサーバに送られる
 - ▶ 寿命になると削除される
 - ▶ ひとつのクッキーあたり4096バイト程度利用できる

10

クッキー情報

- クッキー情報
 - ▶ Set-cookie: 名前=値 [; domain=ドメイン] [; path=パス] [; expires=日時] [; secure]

名前=値	クッキーの情報の本体 複数のクッキーを利用する場合は名前を変える
domain	そのクッキーが利用するドメイン名
path	複数のCGIでクッキーを共有するときに用いる クッキーの有効期限
expires	指定しないとブラウザを終了するまで有効 「曜日, DD-月名-YYYY HH:MM:SS GMT」
secure	SSLなど安全な接続をしているブラウザに送られる

11

クッキーの用途例

- ユーザ認証の簡略化
 - ▶ ID,パスワードなどをクッキー情報としてユーザーに配信
- ショッピング等での応用
 - ▶ 商品をカゴに入れることにクッキーをユーザへ送信。購入指示時にその全てをサーバに送り返してもらい集計
- ユーザのアクセスタイムの把握
 - ▶ アクセス時に時間をクッキーとして送信。退出時にそれを送り返してもらうよう設計
- ユーザの嗜好を把握
 - ▶ ユーザがアクセスしたページをクッキーとして保存
- サーチエンジンでの利用
 - ▶ キーワード、検索したページ数、現在表示中のページ数などをクッキーによって保存

12

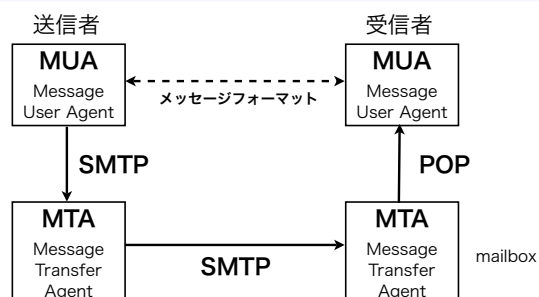
電子メール

- 構成要素

- ▶ メッセージフォーマット
- ▶ **MTA : Message Transfer Agent**
 - メールの配送を行う
- ▶ **MUA : Message User Agent**
 - ユーザインタフェースを担当
 - MTAからメールを受け取ってユーザに渡す
 - ユーザが送ったメールをMTAに転送

13

電子メール配送の仕組み



14

メッセージフォーマット

- メールのメッセージフォーマット

- ▶ 「ヘッダ部」と「ボディ部」
- ▶ CR (0x0D) LF (0x0A)のみの行で区切られる

15

ヘッダ部

- 複数行の「ヘッダフィールド」で構成される
- 「ヘッダフィールド」 =
- 「フィールド名」 + ":" + 「フィールドボディ」
- 各ヘッダフィールドの最後はCRLF
- 記述順序は問わない
- 長いヘッダフィールドはフィールドボディを複数行に分割可
 - ▶ 2行目以降の先頭には、スペースあるいはタブ(TAB)を挿入

16

ヘッダフィールド名の例

Date	メールの発信時刻	Message-ID	メールのメッセージID
From	メールの発信者	Received	MTAによって追加される転送記録
Sender	実際の送信者	Reply-To	返事の宛先を指定する場合のアドレス
To	メールの宛先	In-Reply-To	返事の元になるメールのメッセージID
Cc	カーボンコピーによるメールの宛先	Return-Path	エラーメールなどの送付先
Bcc	ブラインドカーボンコピー(表示されない)	X-*	任意の名前
Subject	メールの表題		

17

ボディ部

- 7ビットUS-ASCII
- 行末はCRLF
- 一行は1000バイト以内
- 当初は構造をもたなかった
- MIMEの導入により構造化も可能

18

インターネットメールにおける日本語

- ISO-2022-JPによる符号化方式
 - ▶ エスケープシーケンスによる文字セットの切り替え
 - ESC (B US-ASCII文字
 - ESC \$B JIS X 0208-1983 (新JIS漢字)
 - ▶ 行の始めはUS-ASCII文字で始まる
 - ▶ 行の終わりはUS-ASCII文字に切り替える
 - ▶ 例
 - 西9号館
 - 1b 24 42 403e 1b 28 42 39 1b 24 42 3966 345b 1b 28 42
 - ESC \$ B 西 ESC (B 9 ESC \$ B 号 館 ESC (B

19

MIME

- Multipurpose Internet Mail Extensions
 - ▶ US-ASCIIテキスト文字以外の言語文字列やイメージ、音声、ビデオなどマルチメディアデータが扱える
- 設計上考えなければいけないこと
 - ▶ 上位互換性
 - ▶ 下位互換性
- メッセージフォーマット
 - ▶ 本文で扱えるのはUS-ASCIIのみ
 - ▶ 一行1000文字以内

20

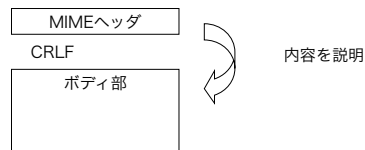
上位・下位互換性

- 上位互換性
 - ▶ 上位（後発）のソフトウェアなどの製品が、下位（先発）の製品のデータを扱えること
- 下位互換性
 - ▶ 上位の逆
 - ▶ 一般的には、制限のついた下位互換になることが多い
- インターネットではとても重要な概念

21

MIMEメッセージの構造

- ボディ部のデータ内容をヘッダフィールドで規定



- 主なヘッダフィールド

MIME-Version	MIMEバージョン
Content-Type	コンテンツタイプ/サブタイプ
Content-Transfer-Encoding	コンテンツ転送/符号化
Content-ID	コンテンツ識別子
Content-Description	コンテンツ記述
Content-Transfer-Encoding	コンテンツ転送/符号化

22

主なMIMEヘッダ

- MIME-Version
 - ▶ MIMEのバージョンを記述、現在のバージョンは1.0
 - ▶ 記述例
MIME-Version: 1.0
- Content-Type
 - ▶ 受信側のMUAが、メッセージを識別/処理するために必要なデータや符号化の種類を記述
 - ▶ 書式と記述例
Content-Type:=type "/" subtype * [";" parameter]
Content-Type: text/plain; charset=iso-2022-jp

23

よく使われるContent-Type

text/plain; charset="..."	通常のテキスト (デフォルトcharsetはUS-ASCII)
text/html	HTMLテキスト
application/octet-stream	一般的バイナリデータ
application/pdf	PDFデータ
application/msword	MS-Wordデータ
application/vnd.ms-excel	MS-Excelデータ
application/vnd.ms-powerpoint	MS-PowerPointデータ
application/xml	XMLデータ
message/rfc822	メールの転送
img/jpeg	JPEGデータ
img/png	PNGデータ
multipart/mixed	様々なタイプの混合

24

multipart/mixedの例

```
Content-Type: multipart/mixed; boundary="--Next_Part--"

--Next_Part--
Content-Type: text/plain; charset=iso-2022-jp
Content-Transfer-Encoding: 7bit

JPEGの画像を添付します。

--Next_Part--
Content-Type: image/jpeg; name="example.jpeg"
Content-Transfer-Encoding: base64

R0lGODlhggIPAvUAAAAAAAAA/+AQICAgL+6u8bCw8vHyM3K
ys/MzNHNzdTQ0NbS0tjU1NjY1dzc2t7d2+Hf3OPf3uDg3uDh
3uLj3/jqzvrqzfnszv3szf3tzvnu1frt1f3wz//z1v3w2fzw
2v702f/12+Pk4OPk4eTk4+fk4ufp5Ovo5+ns5+3t5+vs6u/v
607v703w6vHw6/Dx7vL17/Tz

--Next_Part--
```

25

Base64 Encoding

- 3バイトのバイナリデータを、4つの6ビットデータに分割
- 各6ビットデータを印字可能なUS-ASCII文字に置換
- 変換元のデータが3バイトに満たないときは、結果が4文字になるようpadding文字として"="を追加
- 0xC17305の変換例

0xC17305 → 11000001 01110011 00000101

↓ ↓ ↓ ↓

110000 010111 001100 000101

(48) (23) (12) (05)

w X M F

wXMF ←

26

6bit値	Base64	6bit値	Base64	6bit値	Base64	6bit値	Base64
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v	(pad)	=
14	O	31	f	48	w		
15	P	32	g	49	x		
16	Q	33	h	50	y		

27