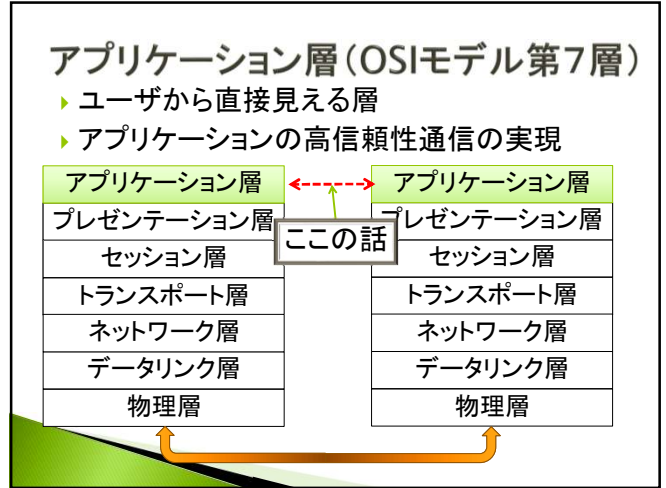




1



2

アプリケーション層の役割

- ▶ ユーザが直接使用するプロトコルの提供例)
 - DNS
 - **メール関連**
 - ・ SMTP
 - ・ POP
 - ・ IMAP
 - HTTP

3

電子メールの見方

Delivered-To: hanada@cc.saga-u.ac.jp

Received: from mx1.cc.saga-u.ac.jp (mx1.cc.saga-u.ac.jp [133.49.4.3])
 by spool0.cc.saga-u.ac.jp (Postfix) with ESMTTP id 2A6FA4B650
 for <hanada@cc.saga-u.ac.jp>; Mon, 9 Jun 2017 09:50:35 +0900 (JST)

(中継)

Received: from mail.is.saga-u.ac.jp (ns.is.saga-u.ac.jp [133.49.20.1])
 by mx0.cc.saga-u.ac.jp (MOS 4.4.6-GA)
 with ESMTTP id BCW84670;
 Mon, 9 Jun 2017 09:50:35 +0900

Received: by mail.is.saga-u.ac.jp (Postfix)
 id B6D1415294; Mon, 9 Jun 2017 09:50:35 +0900 (JST)

Received: from [133.49.20.19] (jimpic.is.saga-u.ac.jp [133.49.20.19])
 by mail.is.saga-u.ac.jp (Postfix) with ESMTTP
 id 40E8815233; Mon, 9 Jun 2017 09:50:35 +0900 (JST)

Message-ID: <5574E6DA.3020801@is.saga-u.ac.jp>

Date: Mon, 09 Jun 2017 09:50:34 +0900

From: 吉川 みき <furukawa@is.saga-u.ac.jp>

Reply-To: all@is.saga-u.ac.jp

Subject: [all 7262]【重要】平成29年度 本庄キャンパスの停電作業について

Content-Type: text/plain; charset=iso-2022-jp

Content-Transfer-Encoding: 7bit

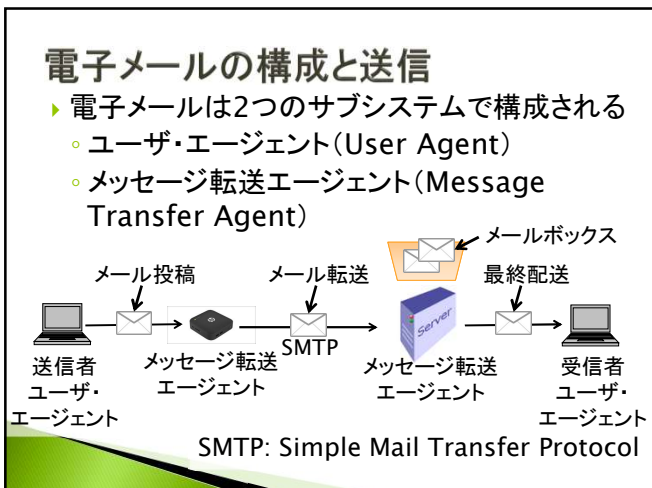
吉川です。

おはようございます。

下記のお知らせが届いています。(以下略)

- ▶ メールは「ヘッダ」と「本文」(と「添付ファイル」)からなる
- ▶ ヘッダは次の情報を含む
 - ユーザが指定
 - ・ 差出人
 - ・ あて先
 - ・ 送付時刻
 - ・ タイトル(Subject)
 - システムが利用
 - ・ 伝送ルート
 - ・ 内容の型と文字コード

4



5

ユーザ・エージェント

- ▶ いわゆる「メーラ」「メールリーダ」と呼ばれるソフトウェア
- ▶ ユーザ・エージェント(の多く)が持つ機能
 - メールサーバからのメールの受信と保存
 - メールサーバへのメール発信の依頼
 - ヘッダ情報の一覧表示
 - フォルダへの分類
 - パケーションエージェント機能
 - ・ 設定すると、受信したすべてのメールにあらかじめ設定したメールを返信する(不在設定等)
 - 暗号化/復号
 - 添付ファイルの添付/分離

6

ヘッダ情報

▶ 主なヘッダ情報(RFC5322)


ヘッダ名	意味	ユーザ設定
To:	あて先メールアドレス(主、必須)	○
Cc:	あて先メールアドレス(副)	○
Bcc:	ブラインド・カーボン・コピー用メールアドレス(非表示)	○
From:	メッセージ作成者	○
Sender:	実際の送信者のメールアドレス	×
Received:	経路上の各転送エージェントが加えた行	×
Return-Path:	発信元への経路識別に利用可能(エラー時に使用)	×
Date:	メッセージが送信された時刻(年月日と時間)	×
Reply-to:	返信先メールアドレス	○
Message-id:	このメッセージを参照するための一意な識別子	×
References:	関連する他のメッセージ識別子(返信、転送時に使用)	×
Subject:	1行表示の短いメッセージ概要	○

7

その他のヘッダ情報

▶ X-で始まるヘッダについて

- RFC5322が認めた非公式(個人的)ヘッダ一例(実際に佐大内で使われているもの)
 - ・ X-ML-Name: メーリングリスト名
 - ・ X-Mail-Count: メーリングリスト内の番号
 - ・ X-MLServer: メーリングリストが使っているソフトウェア名
 - ・ X-mailer: 発信人が使ったメーラソフトウェア名



8

アプリケーション層(2)

花田 英輔

9

MIME

(Multipurpose Internet Mail Extensions)

▶ 電子メールの多言語化等に対応

- ファイルの添付にも活用

▶ 新たなヘッダの追加

- MIME-Version: MIMEのバージョンを指定
- Content-Description: メッセージの中身の説明
- Content-id: 一意の識別子
- Content-Transfer-Encoding: 転送のためのメッセージ本体のコード識別(主に文字コード)
- Content-Type: コンテンツの型、形式
 - ・ 次に例示

10

Content-Typeの例

▶ コンテンツの型、形式、名称等を表す例)

Content-Type: text/plain; charset="UTF-8"
→ テキスト(平文、文字コードはUTF-8)

Content-Type: text/html; charset="UTF-8"
→ テキスト(HTML形式、文字コードはUTF-8)

Content-Type: application/pdf; name="00-xxx.pdf"
→ アプリケーションデータ(PDF形式、ファイル名は"00-xxx.pdf")

Content-Type: application/octet-stream; name="01-xxx.doc"
→ アプリケーションデータ(バイトデータ、ファイル名は"01-xxx.doc")

↑ コンテンツ・タイプ ↑ コンテンツ・サブタイプ

11

MIMEのタイプ/サブタイプの例

▶ 複数のRFCで規定された

タイプ	サブタイプ	意味
Text	plain, html, css	各種形式のテキスト
image	gif, jpeg, tiff	静止画像ファイル
audio	basic, mpeg, mp4	音声、音楽等のファイル
video	mpeg, mp3, quicktime	動画ファイル
model	vrml	3Dモデル
application	octet-stream, pdf, javascript, zip	アプリケーションソフトのデータファイル
message	http, rfc822	カプセル化されたメッセージ
multipart	mixed, alternative, parallel, digest	複数タイプの組み合わせ

12

メッセージ転送プロトコル

- ▶ メッセージ転送プロトコルの役目
 - メールの投稿
 - サーバ間でのメールの転送
- ▶ どちらもSMTPで実行可能
 - ただし
 - ・ SMTP自体は認証を行わない
 - ・ 端末とサーバ間はSMTP自体はコネクションを確立できない
 - ・ TCPやSSLコネクションを使う



13

メールの転送

- ▶ メールアドレスは”ユーザ名@メールサーバ名”で構成される
 - 相手先メールサーバ内のユーザ用メールボックスに配送する
- ▶ メールサーバの場所と経路を知る方法
 - DNSを利用する
 - DNS内のMXレコードを問い合わせることで、相手先メールサーバのIPアドレスを知ることが可能
- ▶ 転送設定とその動作
 - メールサーバに別アドレスを設定し、自動的に転送が可能 → 一旦届いたメールを再配信する

14

最終配送(到着したメールを読む)

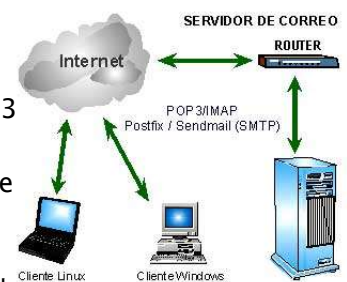
- ▶ メールの配送は相手先メールサーバのメールBOXに届けば終了
- ▶ ユーザはどのようにしてメールを読むのか？
 - 昔)
 - メールサーバにログインして読む(今も可能)
 - メールサーバから手元PC等に取り込む
 - Web画面を通してサーバを見に行く
 - ▶ 手元に取り込む方法/プロトコルは？

15

最終配送プロトコル

- ▶ 主なもの

- POP3
 - ・ Post Office Protocol version3
- IMAP
 - ・ Internet Message Access Protocol
- Webメール
 - ・ Webブラウザを通じたサービス



16

POP3とIMAPの違い

- ▶ POP3
 - メールボックスからメッセージを持ってくる機能のみ
 - POP3は原則的にサーバにメールを残さない
 - ・ 残す設定が可能なものもある
 - IMAPに比べると機能は少ない
- ▶ IMAP
 - メールサーバはIMAPサーバを実行している必要がある
 - 転送、削除、フォルダの作成/削除、メッセージのフォルダへの振り分けなどが可能

17

アプリケーション層(3)

花田 英輔

18

プロトコルとしてのHTTP

▶ RFC2616で規定

- HTTP自体はアプリケーション層のプロトコルであるが、利用拡大に伴いトランスポート層的な使い方も出始めている
- 例)
 - ・ダウンロード、ファイル転送での利用
 - ・入力フォーム等によるデータベースアクセス
 - SOAP(Simple Object Access Protocol)の利用
 - ・家電製品の遠隔操作
- HTTPのバージョンも上がっている
- ・最新は5.0(=HTML5.0を用いた記述)



19

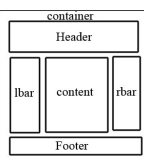
HTTPができること(取り扱える情報)

項目	HTML 1.0	HTML 2.0	HTML 3.0	HTML 4.0	HTML 5.0
ハイパーリンク、画像、リスト	○	○	○	○	○
アクティブマップ、フォーム		○	○	○	○
数式、ツールバー、表			○	○	○
アクセシビリティ機能				○	○
オブジェクト埋込み				○	○
スタイルシート				○	○
スク립ティング				○	○
ビデオと音声					○
インライン・ベクトル・グラフィックス					○
XML表現					○
バックグラウンド・スレッド					○
ブラウザ記憶					○
描画キャンバス					○

20

HTTPによる要求と応答

- ▶ ユーザはブラウザを通してサーバに対し要求する
- ▶ サーバは要求に対して応答する
- ▶ ならば
 - セッションは維持し続けるべきか？
- ▶ しかし
 - ユーザはいつ要求するかわからない
 - ユーザは要求終了を明示しない
 - 画面に埋め込まれている画像等の要求手法は？

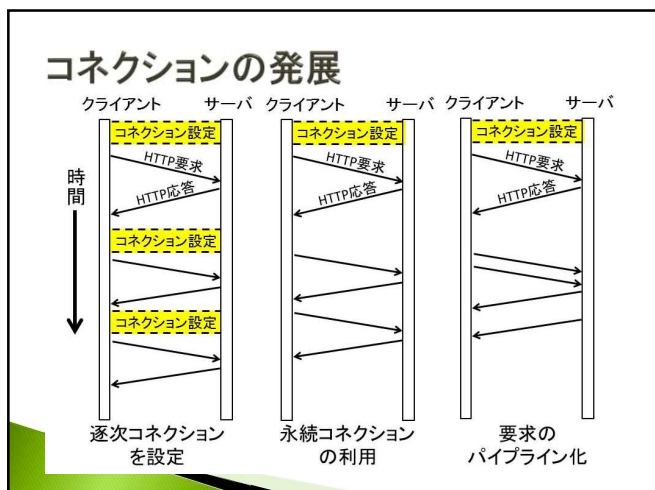


21

コネクションの発展

- ▶ 初期のHTTP(HTTP1.0)
 - ほとんどHTMLテキストのみだったので、1回ごとのコネクションで済んだ
- ▶ HTTP1.1以降
 - 埋め込み画像などが増えたため、永続コネクション(persistent connection)や並行コネクション(parallel connection)が生まれた

22



23

HTTP応答におけるコード

- ▶ HTTP応答はメソッド(method)をサポート
 - ページ操作に関する記述
 - 特に応答時のコードが有用

コード	意味	例
1xx	情報	100:サーバはクライアント要求の処理に同意
2xx	成功	200:処理成功、204:コンテンツ不存在
3xx	転送	301:ページは移動、304:キャッシュ済ページ
4xx	クライアント誤り	403:参照禁止ページ、404:ページ不存在
5xx	サーバ誤り	500:サーバ内部誤り、503:一時的障害

24

今回の課題

1. 電子メールを出したがエラーとなって戻ってくる場合の、原因として考えられること(複数ある)と、各原因におけるエラー内容として付け加えられるメッセージを調べて記せ
2. 本日の感想
 - ▶ 締切: 1月22日(月) 18:00
 - ▶ 以前の課題をこれから出しても採点します
 - ▶ 本講義に関する情報は(この講義資料も)次のWebpageに掲載するので、時々参照すること
<http://www.ai.is.saga-u.ac.jp/~hanada/DCT/>