

INTERNET PROTOCOLS AND CLIENT-SERVER PROGRAMMING

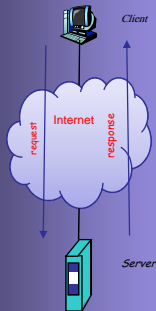
SWE344

Fall Semester 2008-2009 (081)

Module 10.3: Mail Protocols (Part 3)

Dr. El-Sayed El-Alfy

Computer Science Department
King Fahd University of Petroleum and Minerals
alfy@kfupm.edu.sa



Objectives

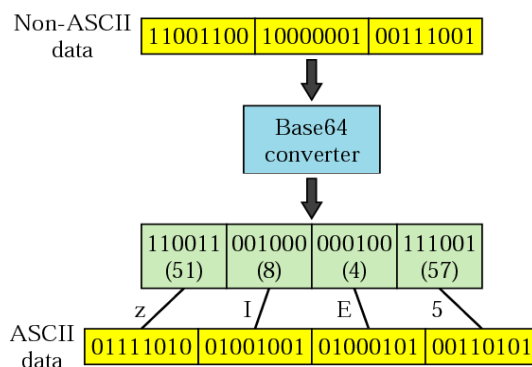
- ✦ Learn about the Multipurpose Internet Mail Extension, MIME, Defined in the following RFCs:
 - [RFC 2045]: <http://www.ietf.org/rfc/rfc2045.txt>
 - [RFC 2046]: <http://www.ietf.org/rfc/rfc2046.txt>
 - [RFC 2047]: <http://www.ietf.org/rfc/rfc2047.txt>
 - [RFC 2048]: <http://www.ietf.org/rfc/rfc2048.txt>
 - [RFC 2049]: <http://www.ietf.org/rfc/rfc2049.txt>
- ✦ Learn about two Transfer-Encoding Schemes used in MIME messages.
 - Base64 Encoding
 - Quoted-Printable Encoding

Introduction

- ✦ SMTP and Message Format Protocol were designed to construct mail and send it using only ASCII characters (7 bits).
 - This turned out to be very limiting since it cannot be used to send non ASCII data [128 – 255].
- ✦ Multipurpose Internet Mail Extension (MIME) was designed to extend the capability of SMTP mail to support sending non ASCII data, but without violating the previous protocols.
 - The mail is still sent using only ASCII data. Other types of data must first be encoded to ASCII using some encoding scheme.
 - The format of the mail remains the same (headers and body). Encoded attachments are just appended to the body.
 - MIME defined additional headers that are used to indicate the content-type, transfer encoding method and boundaries of each attachment so that it can be decoded and retrieved correctly at the receiving end.
 - The two most commonly used encoding schemes are: [Base64](#) and [Quoted-Printable](#).
 - MIME protocol is specified over five Rfcs: [2045](#), [2046](#), [2047](#), [2048](#), [2049](#)

Base64 Encoding

- ✦ Base64 is an encoding scheme that scans a stream of bytes and converts every 3 bytes (24 bits) into 4 blocks of 6 bits.
- ✦ The algorithm then uses its dictionary to convert each resulting block (decimal 0 – 63) into US-ASCII character.



Base64 Encoding ...

- Base64 dictionary defines the following mapping of the 64 possible values to ASCII characters:

6 bits codes	ASCII Characters
0 to 25	Upper case letters 'A' – 'Z'
26 to 51	Lower case letters 'a' – 'z'
52 to 61	Digits '0' – '9'
62	+
63	/

- What if the byte stream is not a multiple of 3?

- Case 1: If one byte is left: Pad two 0 bytes and split into group of 6 bits. Take two base64 characters and replace the last two with the '=' padding character.
- Example: '00000001' → '00000001', '00000000', '00000000'
→ '000000', '010000', '000000', '000000'
→ AQ==

Base64 Encoding ...

- What if the byte stream is not a multiple of 3?

- Case 2: If two are left: Pad one all-zero bytes and split into group of 6 bits. Take three base64 characters and replace the last with '='.
- Example: '00000010', '00000001' → '00000010', '00000001', '00000000'
→ '000000', '100000', '000100', '000000'
→ AgE=

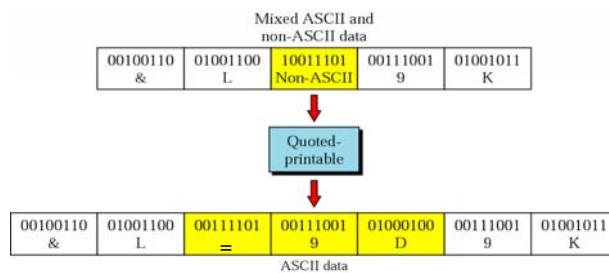
- Base64 encoding requires that a line should not contain more than 76 base64 characters.

- Thus, <CRLF> is added after every 76 characters.

```
ACAABwAjbkoGRwYnBiAAJwZEBiUGLgZiBikGOGAgAEoGQgZIBkQGIAAqBjkGJwZEBkkGewBIBkQG
SAYgADQGJwYhBiAAMQYoBkMGIABEBiwGOQZEBiAAJwZEBkYgJwYzBiAAIwZFBikGIABIBicGLQYv
BikGIABIBkQGJwYgAEoGMgYnBkQGSZAGBiAARQYuBioGRAZBBkoGRgYgACUGRAYnBiAARQZGBiAA
MQYtBkUGIAAxBigGQwZ9AAwGIABCBicGRAYgACcGRAZFBkEGMwYxBkgGRgY6ACAAIwZKBiAARAZI
ICA0NDaWICAgICAgICAgICAgICAgLzEyLzE0Mjbl3CAADBAAAAIAAAeAAAAABgAAAFRpdGxl
AAM=
```

Quoted-Printable Encoding

- ✦ A problem with base64 encoding is that it increases the amount of data by 33% (4 bytes for every 3).
- ✦ If most of the data consist of ASCII (usually the case with mail body) it is more efficient to use the Quoted-Printable encoding.
- ✦ Quoted-Printable works as follows:
 - If a character is ASCII, send it as it is. Exception to this rule is the '=' character, which is treated using the following rule.
 - If a character is non-ASCII, send it as three characters. First character is the equals sign (=). Next two characters are the hex of the byte.



KFUPM: Dr. El-Alfy © 2005 Rev. 2008

7

MIME Headers

- ✦ The MIME protocol provides additional headers that are included in the message body to identify the different parts a mime-message.
- ✦ There are six important headers introduced to identify MIME messages, namely:
 - MIME-Version
 - Content-Type
 - Content-Transfer-Encoding
 - Content-Disposition
 - Content-ID
 - Content-Description

KFUPM: Dr. El-Alfy © 2005 Rev. 2008

8

MIME Headers ...

MIME-Version header:

- ✦ This is a required header indicating that a message is composed using the MIME protocol.
- ✦ **MIME-Version: 1.0** is the only currently defined MIME-Version header.
- ✦ The MIME-Version header is a top-level header and does not usually appear in body parts.
 - An exception is if the body part is an encapsulated message of **content-type: message/rfc822**, which might have its own MIME-Version header.

MIME Headers ...

Content-Type header:

- ✦ Content-Type is used to specify the media (data) type and subtype in the body of a message and to fully specify the representation of such data.
- ✦ There are seven main types defined, namely: **text**, **image**, **audio**, **video**, **application**, **multipart** and **message**.
 - A number of sub-types are also defined under each of these categories.
- ✦ The simple form of the header is: **Content-type: type/subtype**
Example:
 - Content-type: image/gif
 - Content-type: text/plain; charset="iso-8859-1"
 - Content-type: text/html; charset="iso-8859-1"
 - Content-type: application/msword

MIME Headers ...

Content-Type header: **Multipart/Mixed**

- ✦ A MIME message may contain attachments. Such a message must have a **multipart/mixed** Content-Type header at the top.
- ✦ The format is:
`Content-type: multipart/mixed; boundary="uniqueBoundary"`
- ✦ The body of a multipart/mixed message is divided into blocks (one for each attachment).
- ✦ Two dash characters prefixed to the boundary, marks the beginning of each block (also called body part).
`--uniqueBoundary`
- ✦ Each body part has its headers (content-type, Transfer-Encoding, etc.) followed by a blank line and its body.
- ✦ The end of the multipart document is indicated by:
`-- uniqueBoundary --`

MIME Headers ...

Content-Type header: **Multipart/Alternative**

- ✦ A MIME message or message part may have alternative parts, and the MUA is expected to select one of the alternatives for display (e.g. html and text).
 - In such case, the **multipart/alternative** is specified at the top of the message or the message part.
- ✦ The format is:
`Content-type: multipart/alternative; boundary="anotherBoundary"`
- ✦ A multipart/alternative message is divided into blocks (one for each alternative).
 - `--anotherBoundary` marks the beginning of each alternative.
 - Each alternative part has its headers, followed by a blank line followed by its body.
 - The end of the alternative parts is indicated by: `-- anotherBoundary --`

MIME Headers ...

Content-Transfer-Encoding header:

- ✦ The Content-Transfer-Encoding header describes what encoding is used for a particular part of the message body.
 - e.g.: `Content-Transfer-Encoding: base64`
- ✦ If a part does not have a Content-Transfer-Encoding header, the content transfer encoding of the part is assumed to be ASCII.

MIME Headers ...

Content-Disposition header:

- ✦ Content-Disposition header is used to provide information about how to present a message or a body part.
 - The options are **inline** or **attachment**.
- ✦ If '**inline**' is specified, the MUA is expected to display the part automatically together with the main body of the message.
- ✦ If '**attachment**' is specified, the MUA should not automatically display the part.
- ✦ Content-Disposition header can also be used to specify information about the attachment, such as its filename, last modification date, etc.
- ✦ Example:
`Content-Disposition: attachment; filename="saudiflag.gif"`

MIME Headers ...

Content-ID header:

- ✦ Content-ID headers are unique values that identify body parts, individually or as groups.
- ✦ Content-ID is used to distinguish body parts and allow cross-referencing between body parts.

Content-Description header:

- ✦ This is used to add descriptive text to non-textual body parts.

Example 1 ...

```
Received: from khuzama.ccse.kfupm.edu.sa (localhost [127.0.0.1])
  by khuzama.ccse.kfupm.edu.sa (8.11.0/8.9.3) with ESMTP id h4J8QuH074
  for <bmghandi@ccse.kfupm.edu.sa>; Mon, 19 May 2003 11:26:56 +0300
  (Saudi Standard Time)
//deleted
Received: from soldier.ccse.kfupm.edu.sa (196.1.64.147) by
  ccsevs.ccse.kfupm.edu.sa via csmmap
  id 28329; Mon, 19 May 2003 08:29:47 +0000 (UTC)
Received: from icsbmghandi (ics-bmghandi.pc.ccse.kfupm.edu.sa [196.1.65.143])
  (authenticated bmghandi (0 bits))
  by soldier.ccse.kfupm.edu.sa (8.11.1/8.11.1) with ESMTP id h4J8P6D00647
  for <bmghandi@ccse.kfupm.edu.sa>; Mon, 19 May 2003 11:25:06 +0300
  (Saudi Standard Time)
Message-ID: <002f01c31de0$57ce6610$8f4101c4@icsbmghandi>
From: "Bashir Mohammed Ghandi" <bmghandi@ccse.kfupm.edu.sa>
To: "Bashir Mohammed Ghandi" <bmghandi@ccse.kfupm.edu.sa>
Subject: Testing Attachement I
Date: Mon, 19 May 2003 11:26:27 +0300
MIME-Version: 1.0
X-Priority: 3
X-MSMail-Priority: Normal
```

MIME version header

Example 1 ...

X-Mailer: Microsoft Outlook Express 6.00.2800.1106
 X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2800.1106
 Content-Type: multipart/mixed;
 boundary="NextPart_000_002B_01C31DF9.7CF57870"
 Content-Length: 137705

Multipart/Mixed

This is a multi-part message in MIME format.

--NextPart_000_002B_01C31DF9.7CF57870
 Content-Type: multipart/alternative;
 boundary="NextPart_001_002C_01C31DF9.7CF57870"

Start of part 1

--NextPart_001_002C_01C31DF9.7CF57870
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

Part 1 has nested
alternative parts

Salaam,

This is testing attachement.
 Please ignore.

Beginning of Nested Part 1
of main Part 1, which is of
type plain text

Example 1 ...

Regards,
 Bashir

--NextPart_001_002C_01C31DF9.7CF57870
 Content-Type: text/html; charset="iso-8859-1"
 Content-Transfer-Encoding: quoted-printable

Beginning of Nested Part 2
of main Part 1, which is of
type html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
<HTML><HEAD>
<META http-equiv=3DContent-Type content=3D"text/html; charset=3Diso-8859-1">
<META content=3D"MSHTML 6.00.2800.1141" name=3DGENERATOR>
<STYLE></STYLE>
</HEAD>
<BODY bgColor=3D#ffffff>
<DIV><FONT face=3DArial size=3D2>Salaam,</FONT></DIV>
<DIV><FONT face=3DArial size=3D2></FONT>&nbsp;</DIV>
<DIV><FONT face=3DArial size=3D2>This is testing =
attachement.</FONT></DIV>
<DIV><FONT face=3DArial size=3D2>Please ignore.</FONT></DIV>
<DIV><FONT face=3DArial size=3D2></FONT>&nbsp;</DIV>
```

Example 1 ...

```
<DIV><FONT face=3DArial size=3D2>Regards,</FONT></DIV>
<DIV><FONT face=3DArial size=3D2>Bashir</FONT></DIV></BODY></HTML>
```

```
--NextPart_001_002C_01C31DF9.7CF57870--
```

End of nested
alternative parts

```
--NextPart_000_002B_01C31DF9.7CF57870
```

```
Content-Type: application/msword;
      name="KeyboardShortcuts.doc"
```

```
Content-Transfer-Encoding: base64
```

```
Content-Disposition: attachment;
```

```
      filename="KeyboardShortcuts.doc"
```

Beginning of main
part 2

```
0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAAAACAAAAoQAAAAAAAAAA
EAAAowAAAAEAAAD+////AAAAJ8AAACgAAAA////////////////////////////////////////
//deleted
AAAAAAAAogEAAAAAACiAQAAAAAAKIBAAAAAAAAAogEABQAAAAAAAAAAAAAAAALYBAAAAAAfiwA
AAAAAAB+LAAAAAAAH4sAAAAAAAFiwAACwAAACgLAADAIYAALYBAAAAAAAEYAAO4AADCLgAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAA=
```

Example 1 ...

```
--NextPart_000_002B_01C31DF9.7CF57870
```

```
Content-Type: image/gif; name=" graph1.gif"
```

```
Content-Transfer-Encoding: base64
```

```
Content-Disposition: attachment;
```

```
      filename="graph1.gif"
```

Beginning of part 3

```
R01GOD1hlgbYALMAABSSgBjWgBrUgBrYwBzUgB7YwhjYwhrYwh7YxBaUiFrYlKlKXOzt739//3
9///ywAAAAAlgbYAAAE/ldISau9OOvNu/9gKI5kaZ5oqq5s675wLM90bd94ru987//AoHBILBqP
//deleted
YC4EYZGnaLSBDQNYxi9QO+Id3DXMmFCVqEQBBjk0VsnHjbOc42yM5s7kQGKABndhiOS5mVm243hE
XwMtNcO8LtjKVs4Bdz/EMWS4UA2wjbSkJ03pSlv60pjOtKY3zelOe/rToA61qEdN6lKb+tScjgAA
ADs=
```

```
--NextPart_000_002B_01C31DF9.7CF57870--
```

End of Multi-parts

Resources

- ✚ MSDN Library
 - <http://msdn.microsoft.com/en-us/default.aspx>
- ✚ [RFC 2045]: <http://www.ietf.org/rfc/rfc2045.txt>
- ✚ [RFC 2046]: <http://www.ietf.org/rfc/rfc2046.txt>
- ✚ [RFC 2047]: <http://www.ietf.org/rfc/rfc2047.txt>
- ✚ [RFC 2048]: <http://www.ietf.org/rfc/rfc2048.txt>
- ✚ [RFC 2049]: <http://www.ietf.org/rfc/rfc2049.txt>
- ✚ Books
 - Richard Blum, C# Network Programming. Sybex 2002.
- ✚ Lecture notes of previous offerings of SWE344 and ICS343
- ✚ Some other web sites and books; check the course website at
 - <http://faculty.kfupm.edu.sa/ics/alfy/files/teaching/swe344/index.htm>