

MIME and Base64 encoding

Math 348, Spring 2008

Many e-mail messages are now sent using the MIME format. ‘MIME’ stands for *Multi-purpose Internet Mail Extensions*. A typical e-mail message with an attachment will have the two lines

```
MIME-Version: 1.0
```

```
Content-Type: MULTIPART/MIXED; BOUNDARY="-1868689853-06"
```

somewhere in the header matter (the BOUNDARY is used to separate parts of the e-mail).

A typical message with a pdf attachment then looks like this:

```
---1868689853-06
```

```
Content-Type: TEXT/PLAIN; CHARSET=US-ASCII; format=flowed
```

```
This is readable text
```

```
---1868689853-06
```

```
Content-Type: APPLICATION/PDF; NAME=cycle2.pdf
```

```
Content-Transfer-Encoding: BASE64
```

```
JVBERi0xLjQKJcfsj6IKNSAwIG9iago8PC9MZW5ndGggNiAwIFlvdG9yZGVy
```

```
... Ao1JUVPRgo=
```

```
---1868689853-06--
```

The `format=flowed` segments are easy to read, as they consist of plain text characters. Of course, they are represented in ASCII in the computer file.

The part in `BASE64` is encoded as follows.

First the original binary message is split into blocks of 3 bytes (24 bits), with padding by 0’s on the final block. These blocks are divided into 6-bit segments (representing numbers 0–63 in binary, i.e., in base 64). They are then converted into characters (and then ASCII bytes) according to the following table:

base64	0	...	25	26	...	51	52	...	61	62	63
character	A	...	Z	a	...	z	0	...	9	+	/
ASCII	41	...	5A	61	...	7A	30	...	39	2B	2F

If there was i occurrences of ‘0’ added as padding ($i = 1, 2$), this will always result in i trailing 0’s being encoded as A’s. By convention, these final i ‘A’ are replaced by i ‘=’. This makes stripping off the padding mechanical. In the example above, there was one character padded.

EXAMPLE: The terminal ‘olJUVPRgo=’ in the attachment above represents the string 40/37/09/20/21/15/17/32/40/0 which in binary is

```
101000 100101 001001 010100 010101 001111 010001 100000 101000 000000 or
```

```
1010 00100101 00100101 01000101 01001111 01000110 00001010 00000000
```

This is the hexadecimal string *25 25 45 4F 46 0C plus a 00 for padding. Since ‘25’ is ASCII for ‘this string decodes to ‘%%EOF(cr)’.