

情報科学演習
第4回
写真の加工, 添付ファイルの送信

目次

1	本日の目標	1
2	2進法, 16進法, ASCIIコード	1
3	実習	3
3.1	Web browser でファイルを取り寄せる	3
3.2	写真の加工	4
3.3	添付ファイル (Attachment file) の送受信	5
3.4	添付ファイルの保存	6
3.5	レポート課題	6
3.6	補足 0: 後の講義の準備	7
3.7	補足 1: データ作成上の注意	7
3.8	補足 2: 講義のテキスト	7
3.9	補足 3: HTML メールの危険性	8
3.10	補足 4: 添付ファイルの実際	8

1 本日の目標

- ASCII(アスキー) 文字と 2 桁の 16 進数との対応 (ASCII コード) の仕組みを知る。
- 画像の切り出し, 拡大・縮小の操作を通じて, コンピュータにおける画像の扱いを知る。
- メールにおける添付ファイルの扱いについて知る。

2 2進法, 16進法, ASCIIコード

現在のコンピュータは電気の有無で, 1, 0 を表します。コンピュータの中では全てのデータは 0, 1 の列, 即ち 2 進法の数です。2 進法 1 桁を 1bit(ビット, binary digit の略) といいます。例えば, Nintendo 64 の 64 は 64 ビットに由来します。さらに 8bit を 1Byte(バイト) といいます。2 進法の世界では $1024 = 2^{10}$ 毎に単位が K(キロ), M(メガ), G(ギガ), T(テラ) の単位をつけます。これに対して, $10^3 = 1000$ 毎には, KiB(キビ), MiB(メビ), GiB(ギビ), Tib(テビ) というようになりました (コンピュータの世界だけです)。

しかし 2 進法では, 簡単な情報を表すのにも膨大な桁数が必要です。そこで, 2 進法を 4bit 毎に区切って, 16 進法を使う事が普通です。16 進法では 1Byte が 2 桁の数になります。16 進法では 10 進法の 10, 11, 12, 13, 14, 15 に文字 a, b, c, d, e, f (もしくは, A, B, C, D, E, F) を使います。

コンピュータの中では全てが 2 進法の数ですから, 文字も数に変換されています。この文字と数の対応を文字コードといいます¹。文字コードにはいくつかの種類がありますが, ここでは最も良く使われ, かつ簡単な ASCII(American Standard Code for Information Interchange) コードを紹介します。次の表で最初の 2 列は制御文字と呼ばれる物です。

ASCII コード表

下位 \ 上位	0(0000)	1(0001)	2(0010)	3(0011)	4(0100)	5(0101)	6(0110)	7(0111)
0(0000)	NULL	DLE	Space	0	@	P	'	p
1(0001)	SOH	DC1	!	1	A	Q	a	q
2(0010)	STX	DC2	"	2	B	R	b	r
3(0011)	ETX	DC3	#	3	C	S	c	r
4(0100)	EOT	DC4	\$	4	D	T	d	t
5(0101)	ENQ	NAC	%	5	E	U	e	u
6(0110)	ACK	SYN	&	6	F	V	f	v
7(0111)	BEL	ETB	'	7	G	W	g	w
8(1000)	BS	CAN	(8	H	X	h	x
9(1001)	HT	EM)	9	I	Y	i	y
A(1010)	LF	SUB	*	:	J	Z	j	z
B(1011)	VT	ESC	+	;	K	[k	{
C(1100)	FF	FS	,	<	L	\	l	
D(1101)	CR	GS	-	=	M]	m	}
E(1110)	SO	RS	.	>	N	^	n	~
F(1111)	SI	US	/	?	O	_	o	DEL

¹より細かくは, 利用可能な文字の集合である「文字セット」と, それらをどのように 2 進法の数に対応させるかという「エンコーディング」の 2 つの部分からなりますが, そのような話は複雑なので省略します。

表より、Ascii コードでは、大文字の O は 16 進で数 4F(10 進で 79, 2 進で 01001111) に対応します。通常は 16 進の数は前に、0x をつけて、0x4F のように表記します。

文字コードに対するいくつかの注意

ASCII コードは 7bit 上の表からわかるように、ASCII コードは 7bit です (最上位の bit は常に 0)。しかし今のコンピュータはデータを Byte 単位で処理を実行しますので、最上位のビットを 0 として、8 bit のコードになっています。

文字コードは何種類もある 上でも言いましたが、数字と文字の対応表は他にもあります。ASCII 以外で代表的なのが大型汎用計算機 (Main Frame, 銀行のオンラインシステムなどで使われている) で使われている EBCDIC と呼ばれる文字コードです。

日本語の扱い 日本語, 中国語, ハングルなどの文字数の多い言語では、文字コードはより複雑になります。日本語の文字コードの代表的な規格は、JIS コード 2 種類と、EBCDIC の 3 つです。

インターネットでは 通信に於けるデータ交換では、お互いの合意があればどのようなデータ交換も可能です。しかし、通信をする度に文字コードをいちいち決めていたのでは大変です。そこで、インターネット通信で標準的に用いられる文字コードが決められています。他の言語と一緒に送っても問題の出ない文字コードは、日本語については昔の JIS コードから派生した iso-2022-jp という規格と Unicode から派生した iso-10646 という規格です。

過去のしがらみ 以前、JIS X 0201 1976 という 8bit の文字コードが (ISO-646 に基づいて) 定義されました。これは、最初の 7 ビットの部分を Ascii コードの 2 カ所に変更を加え、残った部分にカタカナを定義しました。この規格は日本のパソコンで多く用いられてきました。これは 2000 年に新しい JIS 規格に吸収されましたが、その規格にソフトウェアが対応していないことが数多くあります。この規格で定義されているカナ文字が、半角カナ文字と呼ばれる物です。上にも述べたように、きちんとした規格通りに動作しないソフトウェアが沢山あるので、これで定義されている半角カナ文字を使った文書を他人に送ったり渡したりしてはいけません。迷惑行為です。

現在の Windows の日本語環境では、\ のコードに対して ¥ 記号を表示しています。これはこの過去のしがらみ (JIS X 0201 1976) のためで、UTF-8 がデファクトスタンダード (事実上の標準) となった現在では、そろそろ廃止すべきものです。

半角/全角 元々は印刷業界用語で、文字幅の意味です。文字コードには文字幅は定義されていず、数と文字の対応だけが定義されています。多くの環境で、1byte 文字が 2byte 文字の半分の幅で表示されたため、半角文字という俗称が生まれました。2000 年にはこの俗称を追認する形で JIS X 201 1976 カナ文字に half-width という名称がつけましたが、今では意味がありません。この授業では (昔の風習を用いて) 1byte 文字と言う事にします。

他の言語では ASCII コードが合衆国でしか通用しないのは明らかです。ヨーロッパ圏やトルコ語、タイ語用の規格として多くで用いられていたのは、ISO-8859-1~ISO-8859-16 と決められている物です。これらは最近、下の UTF-8 に置き換わって行っています。

この講義では この講義では、主に UTF-8 と呼ばれるコードを用います。これは上に述べた ISO-10646 とほぼ同じです。

昔、日本で売られたパソコンでは MS-Kanji と呼ばれる文字コード、およびこれから派生した文字コードが使われていました。これはマイクロソフトウェア・アソシエイツと三菱電機 (アスキー社も加わっていた?) が定めた日本語の文字コードを少し変形したものです。上で述べた、JIS X 0201 1976 の規格を 2byte にし、増えた部分に JIS で定められた文字セットを少し番号をずらして割り振ってできたものです。今でこそ、過去との互換性のために国際的に登録された文字コードとなっていますが、元となった JIS 規格の規格違反をしている、多数の方言がある、ASCII 以外のコードとの共存ができない等問題点の多い文字コードです。

なお、コンピュータは、上で述べた main frame のように、パーソナルでないものもあります。過去には、何種類かの日本語文字コードが、コンピュータシステム毎に存在していました。それらの中で、代表的なもののひとつが、EUC (Extended Unix Code) です。

文字コードはこれからも変化する 日本語に限っても文字コードはまだまだ不完全な物です。これから先も変化し続けると考えられます。その変化を追いかける専門家になる事は不要ですが、その変化に対して合理的な行動がとれるように心がけて下さい。

より詳しく知りたいなら Wikipedia(<http://ja.wikipedia.org/>) の文字コードの項は、割と良く書けている。本気で勉強するなら、一冊の専門書を読む必要があります。

3 実習

まずは、写真データの加工について、簡単に講義をします。

3.1 Web browser でファイルを取り寄せる

前回撮影した写真が、この講義の Web ページに置いてあります。写真の置き場所は、プライバシーを配慮して、講義のページからはたどれなくしてあります。また、琉球大学外部からは、アクセスを禁止してあります。

1. Firefox を起動する。
2. URI: <http://www.math.u-ryukyu.ac.jp/%7Esuga/joho/2023/photos/> を入力する。
3. 写真がクラス毎にありますので、自分のクラス (1組の人は、1.jpeg) をクリックするとその画像が表示されます。前回休んだ人は、flower.jpeg を使って下さい。
4. 写真自体を右クリックするとメニューが現れるので、「名前を付けてリンク先を保存」を選ぶ。
5. 出てくるウィンドウの上の方の欄に、ファイル名入力部分があるので、zentai.jpeg という名前で保存する。保存場所は、ダウンロードになっていると思います。「保存」を選ぶと、ダウンロードフォルダに写真データが保存されます。そうっていなかったら、保存ダイアログで、「ダウンロード」を選んで下さい。選び忘れたまま保存した人は、ホームフォルダ、あるいはその中にあるデスクトップと書かれたフォルダに保存されていると思います。

URI の記法で、%7E は、% の次の 2 文字は対応する ASCII コードの 16 進表記です。昔の多くの日本語キーボードでは、チルダ ~ の所に JIS X 0201 1976 のコード「`〱`」が刻印されており、そのことの混乱を避ける意味で使われます。

3.2 写真の加工

最近のデジタルカメラは高画質なので、Webに載せるには、データ量が大き過ぎます。そこで、アプリケーション「Gimp」で、写真の加工します。次の手順で、写真を Gimp を使って開きます。

- ファイルブラウザ (アクティビティで右に現れるキャビネットアイコン) を起動する。
- 左の一覧のダウンロードを左クリックする。
- ファイルの中に zentai.jpeg があるので、それを右クリックする。
- メニューがでて来るので、上から 2 つ目の「別のアプリケーションで開く」を選ぶ。
- 推奨アプリケーションの中にある GIMP(GNU Image Manipulation Program) を選ぶ。
- いろいろなメッセージが出ますが、「次へ」を 4 回選ぶと Gimp が起動する。
- 「Gimp 今日の技」というウィンドウが出ますが、「閉じる」を選ぶ。

写真を縮小する (flower.jpeg を使う人は、すでに縮小済みですので、この操作は飛ばして下さい。)

最近のデジタルカメラの写真は、データサイズが巨大です。印刷用や実験データの解析のためには好ましいのですが、単純に画像をみせるだけだと大き過ぎます。そこで、縮小してデータサイズを小さくします。

画面の上部のメニューから「画像」→「画像拡大縮小」を選びます。(Gimp で開いた画像自体を「右」クリックするとメニューが表れるので「画像」→「画像拡大縮小」を選ぶのも同じ) 拡大縮小のウィンドウが現れるので、画面サイズの「幅 (W)」の所に 1/8 の数値 (通常 750) をキーボードから入力して ENTER を押すと、高さも自動的に 1/8 になります。終わったら、右下の「拡大縮小 (S)」をクリックします。もう 1 度画像を「右」クリックして、「ファイル」→「別名で保存」を選び、「shukusho.jpeg」の名前で保存してあります。この際、「品質」という文字があるウィンドウが現れたら、スライダーをスライドして値を「100」にして「OK(O)」を選びます。(flower.jpeg の人は、縮小はしなくても良い。) こうすると表示画像が小さくなるので、画面下の % 表示がある部分 (表示画像の拡大、縮小率を表示している) の隣をクリックして、表示画像を大きくします。

自分の顔写真を切り取る クラス写真から自分の顔写真を切り取ります。写真は上で作って、画面に開いたままになっている shukusho.jpeg を利用します。flower.jpeg を使っている人は、バラの花を切り取って下さい。

1. Gimp のツールメニューから、選択ツール → 矩形選択選びます (図参照)。
2. 矩形選択ツールで画像の切り取りたい部分の対角線を、マウスでドラッグします。選択された部分が、点線で囲まれます。範囲が気に入らない人は、気に入る範囲になるまでやり直して下さい。間違って選択部分の画像が動いてしまった人は、編集メニューから (右クリックでメニューを出して「ファイル」→「編集」としても良い) 「フローティング選択領域移動のやり直し」を選ぶと元の位置に戻ります。
3. 選択部分が決まったら、編集メニューから「コピー」を選びます。
4. ファイルメニューから、画像の生成 → クリップボードからを選ぶと、コピーした部分の画像からなるウィンドウができます。

5. このファイルを保存します。

- (a) ファイルメニューから「名前をつけてエクスポート」を選びます。
- (b) ウィンドウが現れるので、ファイル名を `myface.jpeg` にしてエクスポートボタンを押します。
- (c) 品質のウィンドウが現れるので、スライダーを動かして 100% にして、「エクスポート」をクリックする。

写真の縮小と、自分の顔写真の切り取りが終わった人は、Gimp の画面を閉じて下さい。ウィンドウ右上の×印の所を押すと、ウィンドウが閉じます。変更を保存するかという質問が出たら、「保存しない」として下さい (ここで間違えて保存をして、`zentai.jpeg` の画像が乱れた場合は、時間のあるときに、上に述べた手順でクラス写真を縮小しなおしてください。)

3.3 添付ファイル (Attachment file) の送受信

e-mail はテキスト (文字の並び) を送受信するもので、画像データ等はそのままでは送れません。テキスト以外の内容を送受信するには、添付ファイルと言う仕組みを使います。ただし、テキストで内容が伝わる物を別形式で無理矢理添付ファイルにする人がいますが、これは良い事ではありません。必要なのは (特に e-mail では) 内容が伝わるかと言う事で、その体裁ではありません。e-mail を利用する際には、体裁を整えるのではなく、「伝えたい内容を的確に記述する」を意識して下さい。

ちなみに、「写メ」と略される「写メール」ですが、実際には、この添付ファイルの仕組みを利用したメールに他なりません。

Web Mail を使用して、自分宛に先程作成した顔写真のファイルを添付して送信します。

1. Firefox で琉球大学の Web Mail のサイト <https://webmail.cc.u-ryukyu.ac.jp/cgi-bin/index.cgi> にアクセスし、login する。
2. 上部のメニューから「作成」を選ぶ。
3. 「宛先」を自分 (`e+自分の学籍番号@eve.u-ryukyu.ac.jp`) にする
「件名」を Attachment test とし、「本文」に「添付ファイルの練習」と書く
4. 件名欄の下にあるファイル添付の「ファイル追加」をクリックする。
5. 添付ファイルを選ぶウィンドウになるので、先程切り出した自分の写真「`myface.jpeg`」を選んで「開く」をクリックします。(左の欄でダウンロードをクリックすると、右の欄に出て来る。人によっては下の方にあるのでスクロールする)
6. 「送信」ボタンをクリックする。
7. 送信が完了すると、受信トレイに移動するというボタンが出るので、クリックする。

これで、顔写真付きのメールが届いているはずです。基盤情報統括センターの Linux + WebMail の設定では、届いた画像をそのまま確認することができないようです。添付ファイルは、下で述べる「保存」をしてから確認するようになっています。

メールソフト (MUA) によっては、画像データがメールの表示欄にそのまま表示されるものもあります。

3.4 添付ファイルの保存

添付ファイルを開くのはとても危険な行為です。多くのコンピュータウイルスは、メールの添付ファイルを開くことで感染します。ただし、これは Windows の場合の話で、Linux では、ウイルスの感染の可能性は Windows ほど高くありません。それでも、知らない人からのメールはもちろん、知っている人からのメールであっても、本文が全く無いメールや、添付ファイルの内容を説明していないメールの添付ファイルは**開かず**に削除して下さい。必要のないファイルをメールに添付することは止めて下さい。上でも述べましたが、添付ファイルを利用せずに情報を伝える事を常に考えて下さい。どうしても必要な場合は、添付ファイルの内容を明記して送るのがマナーです。

添付ファイルの保存は、添付ファイル付のメールを開くと、添付ファイルの項目のところ「ダウンロード」がありますから、これをクリックして保存します。保存先はダウンロードフォルダになるとしますので、おそらく、同じ名前のファイルがあるという警告が出ると思います。

3.5 レポート課題

この講義の最初の課題です。下記の指示に従い、自己紹介その他を書き、ファイルを添付してメールで送って下さい。

提出期限: 5月31日(水)

レポートを受け取った場合は、その旨をメールで返信します。また提出状況は数日ごとに下記のページに掲示します。メールや Web ページで、提出状況を確認の上、未提出および内容に不備のある人は再送信して下さい。締め切りまでに提出がない場合は、未提出ということで評価します。

<http://www.math.u-ryukyu.ac.jp/~suga/joho/2023/reports/>

次の内容を守って、メールを提出して下さい。

- 課題登録を行っている琉大の自分のアカウントから送信すること。
- メールのはじめは、「情報科学演習レポート」とする。
- 宛先は johokagaku@math.u-ryukyu.ac.jp
- 1行目に学籍番号、名前(漢字と読み)を書く。
- 下記の1から4の各項目すべてについて、60字以上で記す。5も必ず書く。6については任意(書かなくても良い)。適当な文字数(35文字程度)で改行を入れること。
 - 1 自己紹介(出身高校と学科を入れること)
 - 2 数理科学科志望の理由、将来希望する職業とその理由
 - 3 現在までの大学の授業で感じたこと(共通教育、語学など専門以外の科目について)
 - 4 現在までの大学の授業で感じたこと(微分積分学等の専門基礎科目や専門科目について)
 - 5 高校時代に学習した情報科目の科目名(社会と情報、情報の科学のどちらか)。
 - 6 この授業に対しての感想や要望。

- 今週作った自分の顔写真のファイル (myface.jpeg) を添付する。授業で使用した写真ではなくデジカメ等で撮影した写真を、適当なサイズ (byte 数) に加工したものを添付しても良い。撮影の際に欠席した学生は、これらの方法で顔写真のファイルを作成し添付すること。なお、画質等を問題にはしないので、雑な写真で十分。どうしても顔写真のファイルが作成出来ない人は、(flower.jpeg の) 薔薇の写真でも良いが、その場合の評価は低くなる。

Web Mail ですから、自宅の PC, タブレット, 携帯電話などでも利用できます。どの端末でメールを送っているかは、分かりませんし、それを問いただすこともありません。

3.6 補足 0: 後の講義の準備

基礎ゼミのクラス写真に入っていない人は、今日の講義に従って自分のクラス写真を shukusho.jpeg という名前で保存して下さい。保存の際にファイルを置き換えるかという質問が出ると思いますが、置き換えて下さい。

3.7 補足 1: データ作成上の注意

コンピュータでデータを作成する (メールを送る) 際には、次の事に注意してください。コンピュータが行うのはデータ処理で、それに対する合理的な考え方をします。

1. ASCII コードにある文字はそのまま ASCII コード (半角文字) を使う。
2. 日本語コードにあるローマ数字は使わない。ASCII のアルファベットを使う。○付きの数字等は使わない。情報交換において飾りは不要であると考える。
3. 日本語コードにある 1 文字の単位 (km,kg 等) は使わない。素直にアルファベットを入力する (そもそも Kilo meter という 2 つの語を略して km なのだから、それを 1 文字にするという考え方が変)。
4. 日本語コードにある 1 文字の (株) とかも使わない。素直に括弧を入力する。
5. 見えない文字 (空白文字や改行) に注意する。空白は半角を使い (ヨーロッパ系言語やコマンドラインの) 語の区切りに用いる。
6. **重要: 空白 (特に全角の空白) で文字位置の調整をしない。** メールで有効なのは改行だけ。

要するに情報の内容の伝達が重要であり、それを飾るものは必要になった時点で別途考えるようにします。また、今見えているものが別の環境に移ったときに、同じように見える保証はありません。例えば、文字の大きさ、空白文字の幅、ウィンドウの大きさは、環境が異なれば違ったものになります。受け取った人は、あなたが作成した画面と同じ環境を利用していないことが、普通です。

3.8 補足 2: 講義のテキスト

先週も述べましたが、この講義のテキストは、

<http://www.math.u-ryukyu.ac.jp/~suga/joho/2023/>

にあります。各回の講義毎に、Web ページと PDF ファイルを置いていきます。何らかの授業で欠席した時には、次の授業までに講義の内容をこなしておいて下さい。実習室は、どこを使っても同じ環境です。また、休んだ際の資料入手も上のページから行って下さい。共通教育棟の実習室にはプリンタがあり、紙に出力する事も出来ます。(ただし、紙にプリントできるのは 1 人あたり年間 250 枚です)

3.9 補足 3: HTML メールの危険性

これから見せるのは、HTML 形式のメールを利用したフィッシングのサンプルです。私が作りました。

問題点は、クリック先が本来行くべきサイトとは、全く異なるところです。この例は、サンプルなので手を抜いてありますが、個人情報を設定するサイトを偽装するのは簡単です。このようにしてパスワードを奪うサイトの作成も簡単です。

つまり、HTML 形式では、ある種の情報を簡単に隠すことができ、それを利用して犯罪を行います。なので、メールの利用に当たっては、次のことに注意してください。

- HTML メールは送らない。
- HTML メールを送るような人とは、あまり付き合わない。
- HTML メールの内容をクリックしない。

どうしても、HTML メールの中身をクリックしなければならないとなった場合、メールからクリックするのではなく、送り主のサイトにブラウザで訪れて、そこから辿るようにしてください。

今では、PC と携帯電話のメール環境を同期することも難しくありません。メールは、携帯電話ではなく、PC を利用した方が (設定に依存するのですが)、安全な対処がしやすいです。

3.10 補足 4: 添付ファイルの実際

次のページにある内容は、私宛に添付ファイルでウィルスが送られて来たメールのデータを少し書き換えたものです。

この内容は、実際のメールがどのようなになっているかを、全部表示してあります。最初の 10 行は、ヘッダ (header) と呼ばれる内容で、メールの返信先 (Return-Path)、配送経路、送り主 (From)、宛先 (To)、件名 (Subject) などが書かれています。メールソフト (MUA) は、これを見て件名や送り主を記述し、また返信の際の宛先を決定します。

消してありますが、差出人は理学部の教員になっていました。ただし、実際にその人がウィルスを送ったのではなく、別の人のコンピュータに生息していたウィルスが、そのコンピュータ内のファイルにあったメールアドレスを使って、差出人を偽って送ったものです。最後の 2 行 (以降は省略しました) からがウィルスデータです。posting.zip という名前の添付ファイルが添えられており、それを何も考えずに Windows 環境でダブルクリックすると、ウィルスに感染するという仕掛けになっています。知っている人からのメールでも、添付ファイルは危ない、と言う事は覚えておいて下さい。逆に言うと、添付ファイルを用いずに必要な情報を伝えるという努力は、常に心がけるべきです。

そもそもメール配送システムは、アスキーコード表にあるデータ以外の情報交換を想定していない形で作られました。従って実際には、下のメールデータを見ればわかる通り、添付ファイルも本文の(アスキーコードからなる)テキストデータになっています。現在では、メールソフトがそれを解釈して画像などに変換して表示しています。下の表示は、メールソフトの解釈をさせずに表示させたものです。

実は、日本語も(例えば、先週見た私から皆さんへのメールは)アスキーコードの表に含まれるデータに変換されてメールで送られています。その変換方式が、上で述べた ISO-2022-JP あるいは、UTF-8 という文字コードです²。これも、メールソフトがその逆変換を行って、日本語として表示しています。実際のメールデータを見るには、Web Mail ではできないようで、別の MUA を利用する必要があります。

```
Return-Path: <xxxxxxx@sci.u-ryukyu.ac.jp>
Received: from math.u-ryukyu.ac.jp (zzzzzz.zzzzz.zzzz.zz [xxx.xxx.xx.xx])
by yyyyyyy.math.u-ryukyu.ac.jp (8.10.1/8.10.1) with SMTP id i2G1fCK19560
for <www@math.u-ryukyu.ac.jp>; Tue, 16 Mar 2004 10:41:12 +0900 (JST)
Message-Id: <200403160141.i2G1fCK19560@yyyyyyy.math.u-ryukyu.ac.jp>
From: xxxxxx@sci.u-ryukyu.ac.jp
To: xxx@math.u-ryukyu.ac.jp
Subject: warning
Date: Tue, 16 Mar 2004 19:02:45 -0700
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="54807803"
```

stuff about you?

```
--54807803
Content-Type: application/x-zip-compressed; name="posting.zip"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="posting.zip"
```

```
UEsDBAoAAAAAAAAFUqCTBdbrAiAFYAAABWAAAPAAAAcG9zdGluZy5odG0uZXh1TVqQAAMAAAAE
AAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAgAAA
```

以下略

²初期のメール配送システムは、7bit(asciiのみ)で作られていました。それに合わせた(7 bit のコードを拡張する形での)文字コードが ISO-2022-JP です。現在では、メール配送システムは、8 bit のコードを利用できるようになったので、UTF-8 と呼ばれる 8 bit のコードを拡張した文字コードも送れるようになりました。