# 符号化

コンピュータリテラシ

2016年5月10日

建部修見

# 0と1の世界

- 計算機の全ての情報は0と1で表現
  - 命令
  - 文字コード
  - 画像
  - 音
  - 動画

# 2進数、8進数、16進数

| 2進数 | | 8進数 | | 16進数 |
|---|---|---|---|---|
| 0000 | 0 000 | 00 | 0000 | 0 |
| 0001 | 0 001 | 01 | 0001 | 1 |
| 0010 | 0 010 | 02 | 0010 | 2 |
| 0011 | 0 011 | 03 | 0011 | 3 |
| 0100 | 0 100 | 04 | 0100 | 4 |
| 0101 | 0 101 | 05 | 0101 | 5 |
| 0110 | 0 110 | 06 | 0110 | 6 |
| 0111 | 0 111 | 07 | 0111 | 7 |
| 1000 | 1 000 | 10 | 1000 | 8 |
| 1001 | 1 001 | 11 | 1001 | 9 |
| 1010 | 1 010 | 12 | 1010 | A |
| 1011 | 1 011 | 13 | 1011 | B |
| 1100 | 1 100 | 14 | 1100 | C |
| 1101 | 1 101 | 15 | 1101 | D |
| 1110 | 1 110 | 16 | 1110 | E |
| 1111 | 1 111 | 17 | 1111 | F |

# 2進数、8進数、16進数(2)

- 001100100001
- 001 100 100 001
- 1 4 4 1 = 01441
- 0011 0010 0001
- 3 2 1 = 0x321

# 文字コード

- 1バイト文字コード（いわゆる半角コード）
  - US-ASCII（7bit）
  - ISO646 – US-ASCIIの一部を変更した各国仕様
    - 円マーク、ウムラウトなど
  - 拡張ASCII（8bit）
    - 半角カナ
    - 西洋諸語（ラテン文字） ISO8859-1 (Latin-1)、東洋諸語（ラテン文字）、ロシア語（キリル文字）、ギリシア語など
    - コード表の切り替えが必要

# ASCII文字コード（US-ASCII）

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | NUL | DLE | SP | 0 | @ | P | ` | p |
| 1 | SOH | DC1 | ! | 1 | A | Q | a | q |
| 2 | STX | DC2 | “ | 2 | B | R | b | r |
| 3 | ETX | DC3 | # | 3 | C | S | c | s |
| 4 | EOT | DC4 | $ | 4 | D | T | d | t |
| 5 | ENQ | NAK | % | 5 | E | U | e | u |
| 6 | ACK | SYN | & | 6 | F | V | f | v |
| 7 | BEL | ETB | ‘ | 7 | G | W | g | w |
| 8 | BS | CAN | ( | 8 | H | X | h | x |
| 9 | HT | EM | ) | 9 | I | Y | i | y |
| A | NL | SUB | * | : | J | Z | j | z |
| B | VT | ESC | + | ; | K | [ | k | { |
| C | FF | FS | , | < | L | ¥* | l | \| |
| D | CR | GS | - | = | M | ] | m | } |
| E | SO | RS | . | > | N | ^ | n | ~ |
| F | SI | US | / | ? | O | _ | o | DEL |

制御文字　　は各国で変更可能な図形文字　* US-ASCIIではバックスラッシュ＼

# JISカナ（半角カナ）

| | 0 | 1 | 2(A) | 3(B) | 4(C) | 5(D) | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | | | | ー | タ | ミ | | |
| 1 | | | 。 | ア | チ | ム | | |
| 2 | | | 「 | イ | ツ | メ | | |
| 3 | | | 」 | ウ | テ | モ | | |
| 4 | | | 、 | エ | ト | ヤ | | |
| 5 | | | ・ | オ | ナ | ユ | | |
| 6 | | | ヲ | カ | ニ | ヨ | | |
| 7 | | | ァ | キ | ヌ | ラ | | |
| 8 | | | ィ | ク | ネ | リ | | |
| 9 | | | ゥ | ケ | ノ | ル | | |
| A | | | ェ | コ | ハ | レ | | |
| B | | | ォ | サ | ヒ | ロ | | |
| C | | | ャ | シ | フ | ワ | | |
| D | | | ュ | ス | ヘ | ン | | |
| E | | | ョ | セ | ホ | ゛ | | |
| F | | | ッ | ソ | マ | ゜ | | |

# 日本語

- （符号化）文字集合
- JIS X 0208
  - ASCIIの図形文字は94文字で、その範囲を2バイト用いて（94×94＝8836文字）定義
  - 1バイト目を区、2バイト目を点とよぶ
  - 1～8区 各種記号、ひらがな、カタカナなど
  - 16～47区 第一水準漢字
  - 48～84区 第二水準漢字
- JIS X 0212
  - 16～77区 補助漢字

# 日本語（２）

- 符号化方式（エンコーディング）
- JISコード（ISO-2022-JP）
  - 7bitコード（メールで用いられる）
  - エスケープで文字コードを変更
    - ESC $ B (1B 0x24 0x42)で漢字
  - JISカナは21～5F
  - 漢字、補助漢字は1バイト目、2バイト目ともに21～7Eで符号化
- シフトJISコード（Shift_JIS）
  - 8bitコード
  - JISカナ A1～DFで1バイト
  - 漢字は2バイト（1バイト目81～9F, E0～EF、2バイト目40～7E, 80～FC）
  - 補助漢字は表現できない

# 日本語（３）

- 日本語EUC（EUC-JP）
  - 日本語UNIXで利用
  - JISカナは2バイトで1バイト目は8E、2バイト目はA1～DF
  - 漢字は2バイトで、1バイト目、2バイト目はA1～FE
  - 補助漢字は3バイトで1バイト目は8F、2バイト目、3バイト目はA1～FE

# 誤り検知と誤り訂正

- 誤り検知
  - パリティビット
    - 1bit追加して1の数を偶数（奇数）にする
  - チェックサム
  - Cyclic Redundancy Code (CRC) 巡回冗長符号
  - Secure hash 暗号学的ハッシュ関数
- 誤り訂正
  - (3,1) repetition code 反復符号
    - 0 を 000, 1 を 111 に符号化する
    - 1bitの誤りは検出、訂正可能
  - など

# Base64符号化

- バイナリデータ（8bit）をテキスト（64文字）に符号化
- メールのMIMEなどで使われる
- 64文字を利用 A-Za-z0-9+/

| テキスト | M | a | n |
|---|---|---|---|
| ASCII | 77 (0x4d) | 97 (0x61) | 110 (0x6e) |
| Bit pattern | 0 1 0 0 1 1 0 1 | 0 1 1 0 0 0 0 1 | 0 1 1 0 1 1 1 0 |
| Index | 19 | 22 | 5 | 46 |
| Base64 | T | W | F | u |

ManはTWFuに変換される

# Base64符号化（2）

- 24bitを4文字に符号化
- 最後の端数は0でパディングされ、=に符号化

| テキスト | M | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ASCII | 77 (0x4d) | | | | | | | | 0 | | | | | | | | 0 | | | | | | | |
| Bit pattern | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Index | 19 | | | | | | | | 16 | | | | | | | | 0 | | | | | | | |
| Base64 | T | | | | | | | | Q | | | | | | | | = | | | | | | | |

- $n$バイトが$4\lceil n/3 \rceil$バイトとなり、約33%のオーバヘッド

# Base64符号化の例

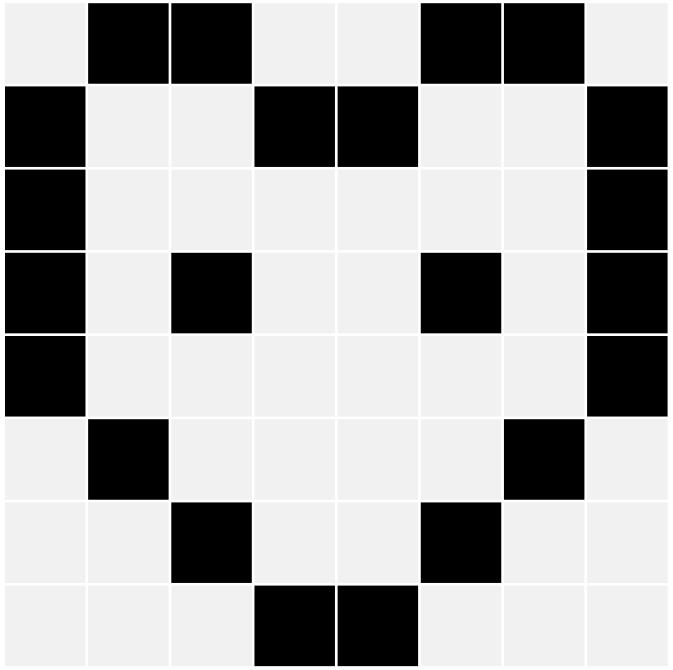
```
Content-Type: image/png; name="google_logo.png"
Content-Disposition: INLINE
Content-Transfer-Encoding: base64
Content-ID: <google_logo>

iVBORw0KGgoAAAANSUhEUgAAAXgAAACACAYAAAABK3JmAAAgOklEQVR42uzde4hUVRwH8KO7z5S5S
iqQ3lWb56oE6M+uymnPPmVlTWrVaVvrDxOqPAlGLqD8kGtl77vooDUkkpT+MzHKLorKQ3VUjCjI0
SKnENe2hOY8e7r1n3Jfu7fxEEMGsmZ29c3fm+4EfM4zr7Ir3fPdw5nfPYQAAAAAAAAPkjZOYWIdUc
YakXuHReF5a9yzDVd4Zln9CvOcK0u/Wjq1/v0a9l9PO0rsPCVF/qx+3CchqFVIujq9SU2Rt6hzAA
ACiMyKqO0cLKPCVMtUNI+zcK73yVYdldOvC/4pazOmo6RnCzG2AAANB/uHl6LLfUi7oOUhB7VYap
TnNpb+WNalZ9k1vGAACg72j2LCznUR2yeyhsC1/2z/pxRbSxfRQDAIDs1bzce5UO9ueyWX7xdlbv
KCHV+pq4cz0DAE8Ftz+yLfRunZtV7ahbyKCw6uNuhQ725YalkrmFr/dBzy1nZW3cHc4AAAEPlxdt
VLXcUkeyCFgfBb39iyEz8xkAIODhovuluolL9R4F5UAvbqp3sD4PgIAHzZB2nWGpPykci6Wo957a
KxkAIOBLUSTuDhXS2eJZ8Hof8ue4VM8z1x3EAAABXypmmWdu1QF4gIKw2ItbahvujAVAwJeEiMyE
hLRPUfiVShmm+iIS//saBgAI+GJlWE6U2go9CtU/9Pfbp+sDWgqinnVhOWu5pdbp55vow1D9fK8w
neO0nNLvP5N0vo7H3cEMABDwxcaQ6kHa46VfwtO0ew1pf8OlY9H30UF+A8tC1breYdHGTNiQzjIu
nQ9pc7I83/3aQdscMABAwBdhf/vs/gh3Lu1vueUspTZLlke0Zi5kZq6wVBPtPolwB0DAw2Vws72a
Qi6fs3UKXqMxM415ILJW3SikY9KGYwh3AAQ8XMAb2sfntcddOh/xBuduVgCz4qevpTV82lce4Q6A
...
/ZqLdU4IIUR1R/qZsZ/T3J7/OUTDfsI6YD+2c7a2tV28xgkhhIiWpsXFUXjb1nj0kb3uGkxS1mQR
LpA0tSTul4R+GPAhb10IIRJQYple3Du6uaM42eLnM1IdxYdMBuHBVEdhOr9j1qqkfYUQQgjxT3tw
QAIAAAAg6P/rdgQqAAAAAAAAABMBOJQ6SEPUEbKAAAAAElFTkSuQmCC
```

# Quoted-printable

- 8bitデータを表示可能なASCII文字に符号化
- 表示可能なASCII文字はそのまま
- それ以外は=XY（XYは16進2桁）で表現
  - =はエスケープキャラクタ
  - =は=3Dと符号化
- 行末の=はsoft line break。復号すると消える
  - 長い文を改行

# 画像

- ビットマップ画像（ラスター画像）



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 66 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 99 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 81 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | A5 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 81 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 42 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 24 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 18 |

- 24bit（RGB各8bit）color 640×480 pixel
  - 640×480×3（byte）＝921,600 byte＝900KiB

# 画像圧縮

- 可逆圧縮（lossless compression）
  - GIF（256色まで）
  - PNG
    - GIFより圧縮率が高いことが多い
    - 文字や線画はサイズも小さくなり適する
- 非可逆圧縮（lossy compression）
  - JPEG
    - サイズが小さくなり写真などの圧縮に適する
    - 文字や線画は圧縮ノイズが目立つ

# データ圧縮

- Run-length encoding（連長圧縮）
  - 連続したデータを長さで表現
  - BBBBBWWWWWWWWWWBBB→B5W9B3
  - BとWしかない場合は593
  - 白黒画像などで利用
  - データが連続していないとデータが大きくなる
  - ABCD→A1B1C1D1

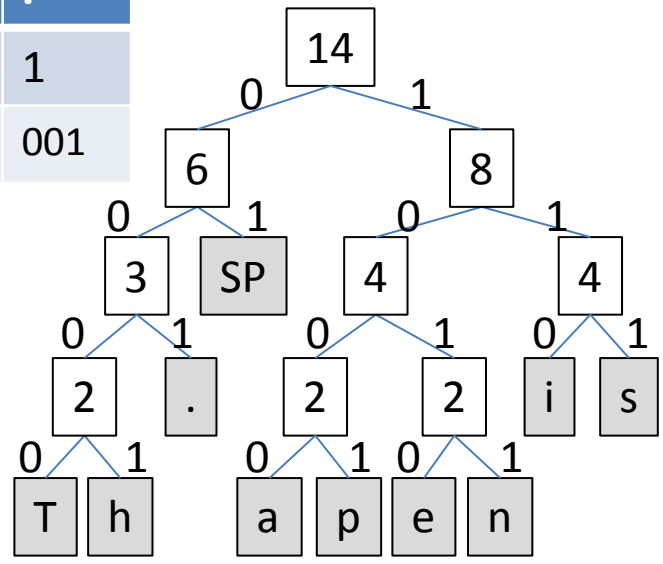
# データ圧縮（2）

- ハフマン符号（1952 David A. Huffman）
  - 文字の頻度により最適な符号化を行う
  - This is a pen.
  - 14文字×8bit＝112bit
  - 14文字×4bit＝56bit（10種の文字は4bitで表現可能）

| | SP | i | s | T | h | a | p | e | n | . |
|---|---|---|---|---|---|---|---|---|---|---|
| 頻度 | 3 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| HC | 01 | 110 | 111 | 0000 | 0001 | 1000 | 1001 | 1010 | 1011 | 001 |

  - ハフマン符号では45bit

ハフマン木



# ベクタイメージ

- 作業座標におけるベクトルを用いたポリゴン（多角形）で表現
  - 色、形、曲線、太さなどを指定
  - 解像度によらない。ズームなど可能
- Scalable Vector Graphics (SVG)
  - World Wide Web Consortium (W3C)の標準

```
<svg>
  <circle cx="75" cy="75" r="75" fill="orange" />
  <circle cx="50" cy="50" r="20" fill="gray" />
  <circle cx="100" cy="50" r="20" fill="gray" />
  <rect x="50" y="100" width="50" height="20" fill="gray" stroke-width="4" stroke="red" />
</svg>
```

# 演習（1）

- 以下のコマンドでファイルを取得しよう

```
$ wget http://www.hpcs.cs.tsukuba.ac.jp/~tatebe/ascii.dat
```

  - このファイルは0～127が1バイトずつ書き込まれた128バイトのファイルである
- od -c ascii.datを実行してASCIIコードを確認しよう
- od -a ascii.datを実行してASCIIコードを確認しよう

# 演習（2）

- 任意のテキストファイルをcompress, gzip, bzip2のそれぞれで圧縮してみよう。サイズがどのようになるか調べよう
- 元に戻して元のテキストファイルに戻るか確かめよう。二つのテキストファイルの差分をトルコマンドにdiffコマンドがある

# 演習（3）

- JPEG画像をPNG画像に変換するとデータサーズはどのようになるか調べよう。適当な画像がない場合は以下を用いよう

  https://www.coins.tsukuba.ac.jp/wp/wp-content/uploads/ho_promotion01.jpg

  なお、convertコマンドで形式の変換ができる

  $ convert foo.jpg foo.png

# オプション演習（１）

- echo -n あ > a.txtを実行して、「あ」だけを含むファイルを作成してみよう
- nkfコマンドを用いて、JISコード、シフトJISコード、EUC-JPに変換してみよう

  $ nkf -j a.txt > a.jis

  $ nkf -s a.txt > a.sjis

  $ nkf -e a.txt > a.euc
- それぞれのファイルサイズを調べよう
- od -t xC a.jisなどとするとa.jisの内容を16進で表示できる。どのようになっているか調べよう。なお「あ」は4区2点である。

# オプション演習（２）

- http://www.w3schools.com/svg/を参照してSVGを学んでみよう