

1 主記憶装置の構成

- 主記憶装置 — プログラムやデータを記憶するための番地のつけられた記憶セル (語) の集合.
- ビット (*bit*, *b*) — データの最小単位, $2^{16} = 65,536 = 64K$, $2^{32} = 4,294,967,296 = 4G$
- 最上位ビット (*Most Significant Bit*, MSB), 最下位ビット (*Least Significant Bit*, LSB) — 1 語の左端・右端のビット
- 語長 — 記憶セルの大きさ, バイト (*byte*, *B*) — 1 文字を格納するための単位, 通常は 8 ビット
- パリティビット (*parity bit*) — 信頼性向上のために付加されたパリティチェック用ビット.
- オクテット (*octet*) — byte が必ずしも 8 ビットを表すとは限らないので, ネットワークの世界では 8 ビットをオクテットと呼ぶ.

2 数の表現

2.1 整数 (integer), 自然数

- 符号 + 絶対値表現 (*signed magnitude representation*) — 符号に 1 ビット割り当てる.
- 1 の補数表現 (*one's complement representation*) — 絶対値を 2 進数で表現し, 負の数であれば 0 と 1 を反転.
- 2 の補数表現 (*two's complement representation*) — 1 の補数表現に 1 を加える.
- n* 増し表現 (*n-biased representation*) — 正のみ

[練習問題] 上記の整数表現法を比較せよ.

2.2 実数 (real number) — IEEE, URR

$(-1)^s \times m \times b^e$ ただし, *m*: 仮数 (*mantissa*, *significand*), *b*: 基数 (*base*), *e*: 指数 (*exponent*).

- 固定小数点表示 (*fixed point number*) — 整数もこの表現法の 1 種
- 浮動小数点表示 (*floating point number*)
仮数の MSB が 1 である浮動小数点表現を正規化 (*normalized*) 浮動小数点表現と言う. (下線部は 2 進数)
 $15 = \underline{1.111} \times 2^{11}$, $-129 = -\underline{1.0000001} \times 2^{111}$,
 $0.125 = \underline{1.0} \times 2^{-11}$, $0.015625 = \underline{1.0} \times 2^{-110}$

浮動小数点表現の標準規格: IEEE 754 floating-point standard

32 ビット表現

1 ビット	8 ビット	23 ビット
符号 <i>s</i>	指数 <i>e</i>	仮数 <i>m</i>

は $(-1)^s \times (1 + m) \times 2^{(e-127)}$ を表現.

$$\left\{ \begin{array}{l} e = 255, m \neq 0: \text{非数}(Nan) \\ e = 255, m = 0: (-1)^s \times \infty \\ 0 < e < 255: (-1)^s \times 2^{e-127} \times \underline{1.m} \\ e = 0, m \neq 0: (-1)^s \times 2^{-126} \times \underline{0.m} \\ e = 0, m = 0: (-1)^s \times 0 \end{array} \right.$$

64 ビット表現

1 ビット	11 ビット	52 ビット
符号	指数	仮数

は $(-1)^s \times (1 + m) \times 2^{(e-1023)}$ を表現.

2.3 IEEE 754 での例外処理

- 桁あふれ (*overflow*): 演算結果が決められたビット数で表現できず.
- アンダーフロー (*underflow*): 値の絶対値が小さすぎて表現できず.

3 情報交換用符号

- ASCII 符号 (American Standard Code for Information Interchange) — 7 ビット符号
- EBCDIC 符号 (Extended Binary Coded Decimal Interchange Code) — 8 ビット符号
- JIS8 ビット符号 — ASCII 符号 + 半角かな文字
- JIS 漢文字符 — 2 バイト使用 JIS X 0208 ('83) ASCII 符号とは Shift-In/Shift-Out の escape sequence (ESC + 2 バイト) により区別.
- EUC-JP 符号 (Extended Unix Code) — 8 ビット. MSB セットで 2 バイト符号. ISO-2022. (G0: ASCII, G1: JIS X 0208-1983, JIS X 208-1990).
- ISO-2022-JP 符号 — MIME 文字セット. ESC (B ⇒ ASCII, ESC (J ⇒ JIS X 0201-1976 (ローマ字), ESC \$ @ ⇒ JIS X 0208-1978 (7 ビット × 2), ESC \$ B ⇒ JIS X 0208-1983 (7 ビット × 2).
- シフト JIS 漢文字符 — 2 バイト (MS-DOS 用)
- UNICODE — 2 バイト (Windows NT, ISO10646)

4 バイト順序 (Byte Order) 要注意

“big-end-first”

バイト番号			
0	1	2	3

例: IBM370, HP, SUN, PC/RT, MC, News, SGI

“little-end-first”

バイト番号			
3	2	1	0

例: Vax, DECstation, 80x86

4.1 ネットワークバイト順序 (Network Byte Order) Big Endian

htons()/htonl()/ntohs()/ntohl()

5 抽象データ型

- カプセル化 — データ構造と演算を一体化, クラス / インスタンス
- クラス — クラス / サブクラス
- 継承 (インヘリタンス)
 - オブジェクト指向 — 1,2,3

A 宿題 — 次回の講義前 (10:30 まで) に TA に 提出すること

A.1 問 1

- 1) 43267 を 2 進数, 7 進数で表現せよ.
- 2) 0.05078125 を 2 進数で表現せよ .
- 3) 43267 を浮動小数点 (2 進数) で表せ.
- 4) -0.75 を浮動小数点 (2 進数) で表せ.

A.2 問 2

次の IEEE 754 浮動小数点数は 10 進数でいくらか.

- 1) 01000011001101100000000000000000
- 2) 10000000011010000000000000000000

A.3 問 3

ビットパターンには固有の意味はない. それに対する命令によってその意味が決まる.

- 1) 計算機中の数と実世界の数との違いを簡単に述べなさい.
- 2) どのような意味を持つか, 例を 4 つ挙げなさい.

A.4 質問・コメントがあれば, お書き下さい.

B 今後の講義予定

1	4/9	講義	計算機内でのデータ表現
2	4/16	講義	アルゴリズムと計算量, 基本的なデータ構造
3	4/23	講義	基本的なデータ構造
4	4/30	試験 1	試験範囲は第 1 から第 3 回まで
5	5/7	講義	優先度つき待ち行列, ヒープ
6	5/14	講義	2 分探索木 (駒谷先生)
7	5/21	講義	平衡探索木
8	5/28	講義	ソーティング (整列)
9	6/4	試験 2	試験範囲は試験 1 以降
10	6/11	講義	ソーティング (整列)
11	6/18	講義	文字列照合アルゴリズム
12	6/25	休講	プログラミング課題: 文字列処理
13	7/2	講義	文字列照合アルゴリズム
14	7/9	講義	文字列処理の応用
15	7/23?	期末試験	試験範囲は試験 2 以降
16	8/1	13:30	プログラミング課題レポート締切

C 成績

試験 3 回, プログラミング課題レポートの合計に宿題 / 理解度テストを加味.