Fundamentals of computer

コンピュータの基礎

Numeric representation
(数の表現)

Base 10
(decimal)
(10進数)

$$1975 = 1\times1000 + 9\times100 + 7\times10 + 5\times1$$
$$= 1\times10^3 + 9\times10^2 + 7\times10^1 + 5\times10^0$$

the 1000's place
(1000の位)

All data in computer are represented by 0 or 1 (binary) : bit (b)

Base 2
(binary)
(2進数)

$$(11011)_2 = 1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$$
$$= 1\times(16)_{10} + 1\times(8)_{10} + 0\times(4)_{10} + 1\times(2)_{10} + 1\times(1)_{10}$$
$$= (27)_{10}$$

Base *r*
(*r*進数)

$$N = a_n r^n + a_{n\text{-}1} r^{n\text{-}1} + \cdots\cdots + a_3 r^3 + a_2 r^2 + a_1 r^1 + a_0 r^0$$
$$= (a_n a_{n\text{-}1} \cdots\cdots a_3 a_2 a_1 a_0)_r$$

Numeric representation
(数の表現)

Base 8 (octal) (8進数) (01234567)

2 digits: $0 \sim 8^2 - 1 = 63$

00: $0\times8^1 + 0\times8^0 = 0$

53: $5\times8^1 + 3\times8^0 = 43$

77: $7\times8^1 + 7\times8^0 = 63$

Base 16 (hexadecimal) (16進数) (0123456789ABCDEF) = $(0 \sim 15)$

2 digits: $0 \sim 16^2 - 1 = 255$

00: $0\times16^1 + 0\times16^0 = 0$

9F: $9\times16^1 + 15\times16^0 = 159$

FF: $15\times16^1 + 15\times16^0 = 255$

(ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789+/) = $(0 \sim 63)$

Correspondence relations (対応関係)

Base 10	Base 2	Base 8	Base 16
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

Convert Base (基数の変換)

Base r to Base 10

$N_r = (a_n a_{n-1} \cdots\cdots a_3 a_2 a_1 a_0)_r$

$N_{10} = a_0 r^0 + a_1 r^1 + a_2 r^2 + a_3 r^3 + \cdots\cdots + a_{n-1} r^{n-1} + a_n r^n$

Ex. $1101_2 = 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 = 13_{10}$

Base 10 to Base r

$$N_{10} = (b_n b_{n-1} \cdots\cdots b_3 b_2 b_1 b_0)_{10} = (c_{n'} c_{n'-1} \cdots\cdots c_2 c_1 c_0)_r$$

$$= \mathbf{c_0} r^0 + c_1 r^1 + c_2 r^2 + \cdots + c_{n-1} r^{n-1} + c_n r^n$$

$$= \mathbf{c_0} + r(\mathbf{c_1} + c_2 r^1 + \cdots + c_{n-1} r^{n-2} + c_n r^{n-1})$$

$$= \mathbf{c_0} + r(\underbrace{\mathbf{c_1} + r(\underbrace{\mathbf{c_2} + c_3 r \cdots + c_{n-1} r^{n-3} + c_n r^{n-2}}_{N_{10}^{(2)}})}_{N_{10}^{(1)}}) =$$

(1) $N_{10}^{(0)} = N_{10} = N_{10}^{(1)} * r + \mathbf{c_0}$ where $0 \le c_0 < r$

(2) $N_{10}^{(1)} = N_{10}^{(2)} * r + \mathbf{c_1}$ where $0 \le c_1 < r$

… repeat until $N_{10}^{(n+1)} = 0$

$\Rightarrow N_r = (c_n c_{n-1} \cdots\cdots c_2 c_1 c_0)_r$

Ex. Base 10 to Base 8

$302_{10} = 8 \times 37 + 6$

$37_{10} = 8 \times 4 + 5$

$4_{10} = 8 \times 0 + 4$

$302_{10} = 456_8$

Python program: base.py

Program: base.py
Usage: python base.py value base_source base_target

Ex.
COMMAND:
python base.py FA 16 8
Convert FA in base 16 to base 8

OUTPUT:
Convert FA in base 16 to base 10
1st digit = 10: $+ 10 * 16^0 \Rightarrow + 10_{10} \Rightarrow 10_{10}$
2nd digit = 15: $+ 15 * 16^1 \Rightarrow + 240_{10} \Rightarrow 250_{10}$

Convert 250 in base 10 to base 8
$250_{10} = 31 * 8 + 2$: base_8 => 2
$31_{10} = 3 * 8 + 7$: base_8 => 72
$3_{10} = 0 * 8 + 3$: base_8 => 372_8 result

Units of data processed in computers
(コンピュータ内のデータ単位)

bit (b): binary: 0 or 1

In computer: 8 bits data is treated as a fundamental unit
byte (B): $0 \sim 2^8 - 1 = 255$

Prefixes in computer science
NOTE: Two different expressions, e.g., kB and KiB, have conventionally often used to mean 2^{10} B.

1 kB = 2^{10} B	**= 1,024 B**	**= 1 KiB** *(kibibyte)*
1 MB = 1024 kB	**= 1,048,576 B**	**= 1 MiB** *(mebibyte)*
1 TB = 1024 GB = 1024^2 MB = 1024^3 kB = 1024^4 B		**= 1 GiB** *(gibibyte)*

The current SI conventional strictly means:
1 kB = 10^3 B = 1,000 B
1 KiB = 2^{10} B = 1,024 B

Numeric representation: Integer (整数型)

Integer type:

Historically, integer sizes in C were related to the CPU word size, but actual sizes are implementation-dependent.
歴史的には、C言語の整数型サイズはCPUのワードサイズと関係していたが、実際のサイズは処理系依存である。

16bits for 16bit CPU

unsigned int (符号無し整数型) $0 \sim 2^{16} - 1 = 65{,}535$
signed int (符号付き整数型) $-32{,}768 - +32{,}767$

32bits for 32bit CPU

unsigned int (符号無し整数型) $0 \sim 4{,}294{,}967{,}295$
signed int (符号付き整数型) $-2{,}147{,}483{,}648 \sim +2{,}147{,}483{,}647$

In C language (typical examples; implementation-dependent):

short int : at least 16 bits, often 16 bits
int : often 32 bits
long int : at least 32 bits, often 32 or 64 bits
long long int : at least 64 bits, often 64 bits

Numeric representation of real values: Floating point

(非整数実数の表現: 浮動小数点型)

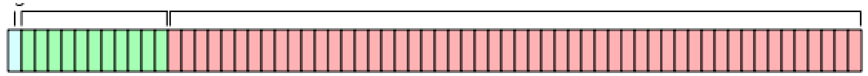
Floating point type: Minimum 32bit (except half precision)

The range of available value depends on computer architectures, programming language etc.

Fixed, hidden bit

$$-1.011101_2 \times 2^{-0101_2}$$

Sign (符号)　Fraction (仮数部)　Exponent (指数部)



C language (C言語)

float　　　: 32 bits　3.4E-38 – 3.4E+38

double　　: 64 bits　1.7E-308 – 1.7E+308

long double: 64 bits

Fortran

Single precision　　(単精度)　FP (REAL)　　: 32 bits

Double precision　　(倍精度)　FP (DOUBLE)　: 64 bits, 16 digits (桁) in decimal

Quadruple precision (4倍精度) FP (REAL*16)　: 128 bits

Definition of IEEE 754 (binary32, binary64):

Sign　　　: 1 bit

Exponent: 8 bits (REAL, –128 ~ +127)　　11 bits (DOUBLE, –1024 ~ +1023)

Fraction　: 23+1(hidden bit) bits (REAL)　52+1 bits (DOUBLE)

8,388,608: 7 digits　　4,503,599,627,370,495: 16 digits

Required variable sizes: Integer types

unsigned int (16 bits): 65,536

16bit CPU can handle only 64 kB of memory

(アドレスバスが16bitだと、64 kBのメモリーしか扱えない)

unsigned int (32 bits): 4,294,967,295

32bit CPU can handle 4 GB memory

(アドレスバスが32bitだと、4 GBのメモリーを扱える)

GDP of Japan: ~5 trillion US$ = 500,000,000,000,000 JYen

(requires 16 digits)

cf. unsigned long long int (64 bit): ~1.8E+19 (18 digits)

The ratio of the circumference of a circle (円周率):

Significant figure: 50 trillion digits (as of Jan, 2020)

Need to use multi-fold calculation (多倍長計算)

Implemented based on software

Required sizes: FP types for quantum calculations

1s orbital energy level:

H atom : 13.6 eV

heavy atoms: >> keV

Energies related to physical properties

Thermal energy at room temperature: 26 meV

Magnetism: several meV

Quantum simulations of physical properties require the precision for the meV – MeV range (over 9 digits precision)

Definition of standard FP: IEEE 754

Fraction: 23 bit	(single)	8,388,608	7 digits
Fraction: 52bit	(double)	4,503,599,627,370,495	16 digits

Required sizes: FP types for semiconductor simulation

Boltzmann factor: $\exp(-E_g / k_B T)$

$\boldsymbol{E_g = 1.1\ \text{eV}}$

$k_B T = \quad 0.026\ \text{eV}\ (T = 300\ \text{K}) \quad \Rightarrow \exp(-42) \sim \mathbf{10^{-19}}$

$\boldsymbol{E_g = 4.0\ \text{eV}}$

$k_B T = \quad 0.026\ \text{eV}\ (T = 300\ \text{K}) \quad \Rightarrow \exp(-154) \quad \sim \mathbf{10^{-67}}$

$k_B T = 0.00026\ \text{eV}\ (T = 3\ \text{K}) \quad \Rightarrow \exp(-15400) \sim \mathbf{10^{-6688}}$

Double precision (64bit): Fraction: 16 digits
Exponent: $-1024 \sim +1023$ ($2^{-1024} \sim 10^{-308}$)

Quad precision : 128 bit

Octuple precision (8倍精度): 256 bit