Beyond Base 10: Non-decimal Based Number Systems

- What exactly is decimal?
- How do other number systems work (binary, octal and hex)
- How to convert to and from non-decimal number systems to decimal
- Non-decimal based math

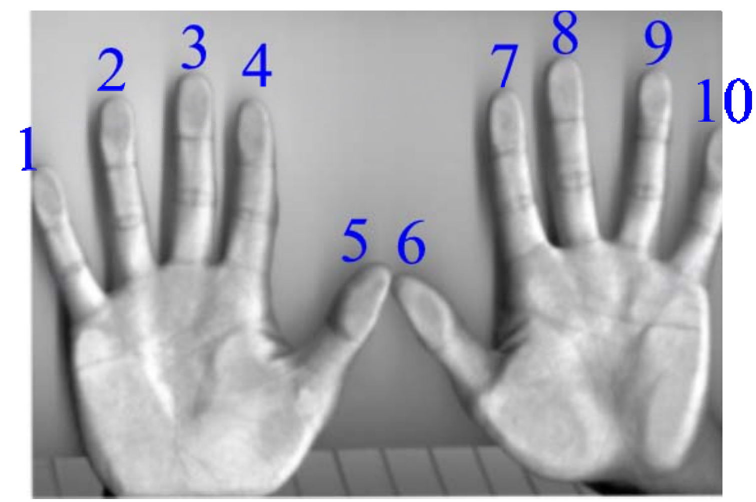
What Is Decimal?

Base 10

- 10 unique symbols are used to represent values

0
1
2
3
4
5
6
7
8
9
10
:

The number of digits is based on…the number of digits



The largest value that can be represented by a single decimal digit is 9 = base(10) - 1

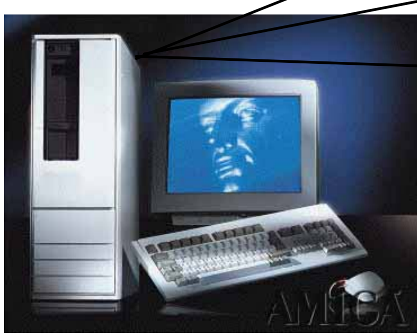
Binary

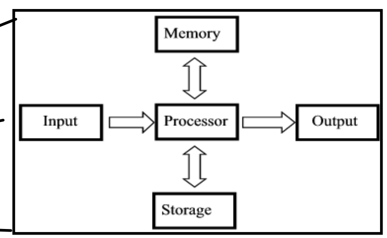
Base two

Employs two unique symbols (0 and 1)

Largest decimal value that can be represented by 1 binary digit = 1 = base(2) - 1

The only language that the computer understands (machine language)





0000 1001 0010 1111

1000 0000 1111 1010

0000 1111 0000 1111

Table Of Binary Values

Decimal value	Binary value	Decimal value	Binary value
0	0000	8	1000
1	0001	9	1001
2	0010	10	1010
3	0011	11	1011
4	0100	12	1100
5	0101	13	1101
6	0110	14	1110
7	0111	15	1111

Octal

Base eight

Employs eight unique symbols (0 - 7)

Largest decimal value that can be represented by 1 octal digit = 7 = base(8) - 1

Table Of Octal Values

Decimal value	Octal value	Decimal value	Octal value
0	0	8	10
1	1	9	11
2	2	10	12
3	3	11	13
4	4	12	14
5	5	13	15
6	6	14	16
7	7	15	17

Hexadecimal (hex)

Base sixteen

Employs sixteen unique symbols (0 – 9 followed by A - F)

Largest decimal value that can be represented by 1 hex digit = 15

Table of Hex Values

Decimal value	Hexadecimal value	Decimal value	Hexadecimal value
0	0	8	8
1	1	9	9
2	2	10	A
3	3	11	B
4	4	12	C
5	5	13	D
6	6	14	E
7	7	15	F

Summary (Decimal, Binary, Octal, Hex)

Decimal	Binary	Octal	Hex	Decimal	Binary	Octal	Hex
0	0000	0	0	8	1000	10	8
1	0001	1	1	9	1001	11	9
2	0010	2	2	10	1010	12	A
3	0011	3	3	11	1011	13	B
4	0100	4	4	12	1100	14	C
5	0101	5	5	13	1101	15	D
6	0110	6	6	14	1110	16	E
7	0111	7	7	15	1111	17	F

Arbitrary Number Bases

Base N

Employs N unique symbols

Largest decimal value that can be represented by 1 digit = Base (N) - 1

Why Bother With Binary?

Representing information

- ASCII
- Unicode

Computer programs

ASCII (American Standard Code for Information Interchange

Decimal	Binary	ASCII
0 – 31	00000000 – 00011111	Invisible (control characters)
32 – 47	00100000 – 00101111	Punctuation, mathematical operations
48 - 57	00110000 – 00111001	Characters 0 - 9
58 – 64	00111010 – 01000000	Comparators and other miscellaneous characters : ; ? @
65 - 90	01000001 – 01011010	Alphabetic (upper case A - Z)
91 – 96	01011011 – 01100000	More miscellaneous characters [\ ] ^ _ '
97 – 122	01100001 – 01111010	Alphabetic (lower case a - z)
123 – 127	01111011 - 01111111	More miscellaneous characters { | } ~ DEL

ASCII (2)

Uses 7 bits to represent characters

Max number of possibilities = 2^7 = 128 characters that can be represented

e.g., 'A' is 65 in decimal or 01000001in binary. In memory it looks like this:

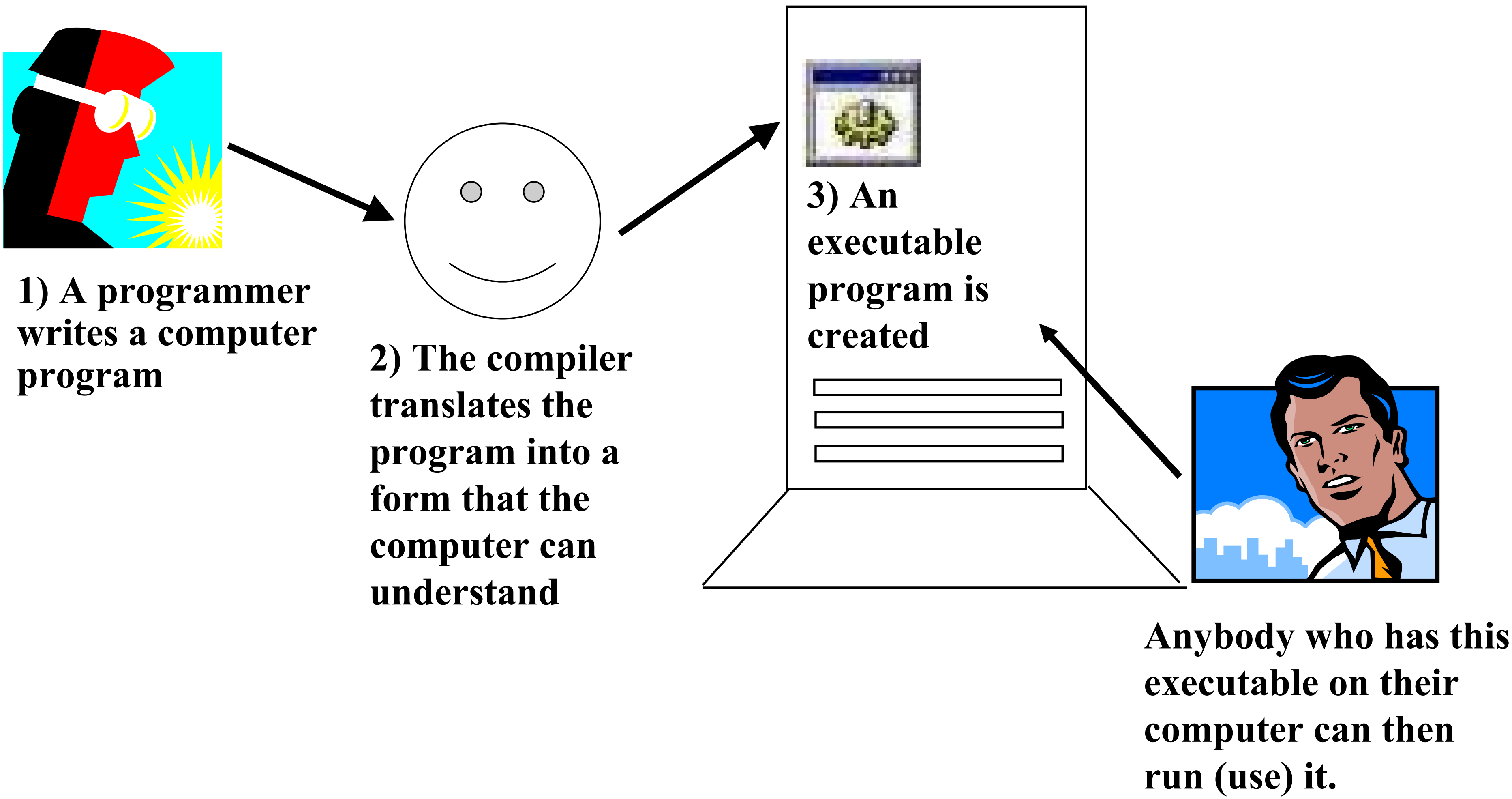
0	1	0	0	0	0	0	1

Unicode

Uses 16 bits (or more) to represent information

Max number of possibilities = 2^{16} = 65536 characters that can be represented (more if more bits are used)

Computer Programs



Uses Of Octal (PDP-11)

Machine language	Octal value
1010111000000	012700
1001010000101	011205

Example from Introduction to the PDP-11 and its Assembly Language by Frank T.

Uses Of Hexadecimal (680X0)

Machine language	Hexadecimal value
11000010000000000000000101011110	6100015E
1000000000001	1001

Converting Between Different Number Systems

Binary to/from octal

Binary to/from hexadecimal

Octal to/from hexadecimal

Decimal to any base

Any base to decimal

Binary To Octal

3 binary digits equals one octal digit (remember $2^3=8$)

Form groups of three starting at the decimal

•For the integer portion start grouping at the decimal and go left

•For the fractional portion start grouping at the decimal and go right

e.g. $101\ 100_2 = ???_8$

$5\ \ 4_8$.

Octal To Binary

1 octal digit equals = 3 binary digits

Split into groups of three starting at the decimal

- For the integer portion start splitting at the decimal and go left

- For the fractional portion start splitting at the decimal and go right

e.g. $12.5_8 = ???_2$

$001\ 010\ .\ 101_2$

Binary To Hexadecimal

4 binary digits equals one hexadecimal digit (remember 2^4=16)

Form groups of four at the decimal

•For the integer portion start grouping at the decimal and go left

•For the fractional portion start grouping at the decimal and go right

e.g., $1000.0100_2 = ???_{16}$

$8\ .\ 4_{16}$

Hexadecimal To Binary

1 hex digit equals = 4 binary digits

Split into groups of four starting at the decimal

•For the integer portion start splitting at the decimal and go left

•For the fractional portion start splitting at the decimal and go right

e.g., $A.3_{16} = ???_2$

1010.0011_2

Octal To Hexadecimal

Convert to binary first!

e.g., 25_8 to $???_{16}$

010101_2

Regroup in groups of 4

$1\ 5_{16}$

Hexadecimal To Octal

e.g., 15_{16} to $???_8$

$0001\ 0101_2$

Regroup in groups of 3

$2\ 5_8$

Decimal To Any Base

1) Split up the integer and the fractional portions

2) For the integer portion, keep dividing by the target base until the remainder is less than the target base

3) For the fractional portion, keep multiplying by the target base until either the resulting product equals zero (or you have the desired number of places of precision)

Decimal To Any Base (2)

e.g., 9_{10} to $???_2$

$1\ 0\ 0\ 1_{\cdot 2}$

9 / 2: q = 4 r = 1

4 / 2: q = 2 r = 0

2 / 2: q = 1 r = 0

1

Stop dividing! (quotient less than target base)

Any Base To Decimal

Multiply each digit by the base raised to some exponent[1] and sum the resulting products.

	3	**2**	**1**	**0**	**-1**	**-2**	**-3** ← **Position of digits**
i.e.	d7	d6	d5	d4.	d3	d2	$d1_b$ ← **Number to be converted**

Base = b

Value in decimal = (digit7*b^3) + (digit6*b^2) + (digit5*b^1) + (digit4*b^0) + (digit3*b^{-1}) + (digit2*b^{-2}) + (digit1*b^{-3})

1 The value of this exponent will be determined by the position of the digit.

Any Base To Decimal (2)

e.g., 12_8 to $???_{10}$

1 0 ← **Position of the digits**

1 2 ← Number to be converted

Base = 8

Value in decimal $= (1*8^1) + (2*8^0)$

$= (1*8) + (2*1)$

$= 8 + 2$

$= 10_{10}$

Addition In Binary: Five Cases

Case 1: sum = 0, no carry out

$$\begin{array}{r} 0 \\ +\ \underline{0} \\ 0 \end{array}$$

Case 2: sum = 1, no carry out

$$\begin{array}{r} 0 \\ +\ \underline{1} \\ 1 \end{array}$$

Case 3: sum = 1, no carry out

$$\begin{array}{r} 1 \\ +\ \underline{0} \\ 1 \end{array}$$

Case 4: sum 0, carry out = 1

$$\begin{array}{rr} {\scriptstyle 1} & \\ & 1 \\ + & \underline{1} \\ 1 & 0 \end{array}$$

1 + 1 = 2 (in decimal)
= 10 (in binary)

Addition In Binary: Five Cases (2)

Case 5: Sum = 1, Carry out = 1

$$
\begin{array}{r}
{}^{1} \\
1 \\
1 \\
+\ \underline{1} \\
1\ 1
\end{array}
$$

$1 + 1 + 1 = 3$ (in decimal)
$= 11$ (in binary)

Subtraction In Binary (4 cases)

Case 1:

$$\begin{array}{r} 0 \\ - \underline{0} \\ 0 \end{array}$$

Case 2:

$$\begin{array}{r} 1 \\ - \underline{1} \\ 0 \end{array}$$

Case 3:

$$\begin{array}{r} 1 \\ - \underline{0} \\ 1 \end{array}$$

Case 4:

$$\begin{array}{rr} \overset{0}{\cancel{1}} & \overset{2}{\cancel{0}} \\ - & \underline{1} \\ & 1 \end{array}$$

The amount that you borrow equals the base

- Decimal: Borrow 10
- Binary: Borrow 2

Summary

What is meant by a number base

How do binary, octal and hex based number systems work and what role they play in the computer.

Converting between non-decimal number systems and decimal

Non-decimal based math (binary addition and subtraction)