COMPSCI 105: Lecture #3
Base Conversions
(The Second of Three Math-Heavy Lectures)

©2014-2020 Dr. William T. Verts

Base Conversions
• What are bases?
  – Labels for numbers using different sets of symbols.
• Why do we care?
  – We use base 10 (decimal) but it isn’t special,
  – Understand how computers work internally,
  – Used in color specifications for Web design,
  – Used in file permissions on Web servers (UNIX),
  – Used in many other places.

Ordinary Decimal Numbers
• Uses the positional notation developed by
  Arab and Hindu mathematicians of a thousand
  years ago.
• Requires the invention of “zero” to work (so
  that 12, 120, 102, 10.2, 100.02 are all distinct).
• Contrast with Roman numerals (I=1, V=5,
  X=10, L=50, C=100, D=500, M=1000).

Decimal Expansion of 123.456
= 1×100 + 2×10 + 3×1 +
  4×0.1 + 5×0.01 + 6×0.001
= 1×10^2 + 2×10^1 + 3×10^0 +
  4×10^{-1} + 5×10^{-2} + 6×10^{-3}

Decimal Expansion of 123
= 1×10^2 + 2×10^1 + 3×10^0
  Base (Radix)  Exponents  Coefficients
Definitions
• Base (Radix):
  – Number of distinct symbols used for counting
• Exponents:
  – Powers of the base for each digit,
  – Increase linearly going to the left, decrease to the right (... 3, 2, 1, 0, -1, -2, -3, ...),
  – Determines the weight/contribution of each digit
• Coefficients:
  – Digits of the number
  – Range is always from 0 to Base-1

What if the Base isn’t 10?
• So what? Math still works.
• Changing a base changes only the labeling of a number, not its value.
• Historical precedent:
  – Sumerians & Babylonians used base 60 (so do we in our clocks and angle measures!)
  – Mayans/Aztecs/Africans/some Europeans used base 20.

Converting from some arbitrary Base into Decimal (Base 10)
• Case #1: Base 4
  • Only four symbols used for counting: 0, 1, 2, 3,
  • No coefficient digit value can exceed 3,
  • Exponents are powers of 4,
  • Everything else is the same.

Case #1: Base 4
• What, then does 1203₄ mean? (Notice the subscript indicating the base)
• Using the same expansion technique:
  – 1×4³ + 2×4² + 0×4¹ + 3×4⁰ =
  – 1×64 + 2×16 + 0×4 + 3×1 =
  – 64 + 32 + 0 + 3 =
  – 99₁₀
• Therefore, 1203₄ is the same number as 99₁₀

What Happened?
• How did we get to Base 10 all of a sudden?
• When we wrote out the expansion of 1203₄ as 1×4³ + 2×4² + 0×4¹ + 3×4⁰, we wrote it down using base 10 (decimal) notation.
• The rest is just arithmetic!
  – Powers first,
  – Multiplications second,
  – Additions third.
Case #2: Base 2 (Binary)

• Only two symbols used for counting: 0, 1,
• No coefficient digit value can exceed 1,
• Exponents are powers of 2,
• Everything else is the same.

What, then does $110101_2$ mean?

Using the same expansion technique:

$-1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 =
-32 + 16 + 0 + 4 + 0 + 1 = -53_{10}$

Therefore, $110101_2$ is the same as $53_{10}$

Notice that multiplications are only by 0 or 1.

A Shortcut for Binary

• Write down the powers-of-two placeholder weights above each digit.
• Wherever there is a 1, add the weight to the total, ignore zeroes.

This Works for Fractions, too!

$32 \ 16 \ 8 \ 4 \ 2 \ 1 \ \frac{1}{4} \ \frac{1}{4}$

$110101.01$

Generalizations

• Conversion techniques presented so far work for converting numbers in bases 2, 3, 4, 5, 6, 7, 8, 9, 10 back into base 10.
• But what about bases greater than 10?
• We will need more symbols.
• By tradition, we use the letters from the Roman alphabet:
  $-0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, G, H, ...$
• Allows up to base 36 (10 digits, 26 letters).

Case #3: Base 16 (Hexadecimal)

• Sixteen symbols used for counting: 0-9 & A-F,
• A=10, B=11, C=12, D=13, E=14, F=15,
• No coefficient digit value can exceed F=15,
• Exponents are powers of 16,
• Everything else is the same.
Case #3: Base 16 (Hexadecimal)

- What, then does 1AE16 mean?
- Using the same expansion technique:
  - 1×16² + A×16¹ + E×16⁰ =
  - 1×256 + 10×16 + 14 =
  - 256 + 160 + 14 =
  - 430₁₀
- Therefore, 1AE₁₆ is the same as 430₁₀

Converting from Decimal (Base 10) into some arbitrary Base

Converting Integers From Base 10

- We now know how to convert to base 10...
- ...but what about converting from base 10 to another base?
- Can we know ahead of time how many digits in the target base we will need?
- Two Methods to know:
  - #1: Simple, understandable, lengthy computations
  - #2: Fast, efficient, hard to explain why it works

How Many Digits in Target Base?

- We need to compute logarithm in the target base B of decimal number N as: log₉(N).
- That is, what power of B gives us N?
- Unfortunately, logarithms in base B aren’t usually available, so we can compute it with...
  - ... logₐ(N) ÷ logₐ(B) for existing log in base A.
- Finally, take the ceiling of the result
  - Smallest integer no smaller than the result.

Example: 99₁₀ to Base 4

- Want: ⌊ log₉(99) ÷ log₉(4) ⌋
- If A=10 (common log):
  - ⌊ log₁₀(99) ÷ log₁₀(4) ⌋
  - ⌊ 1.99563519459755 ÷ 0.602059991327962 ⌋
  - ⌊ 3.3146783100398 ⌋
  - 4
- Base 4 answer needs 4 digits.

Example: 99₁₀ to Base 4

- Want: ⌊ log₈₉(99) ÷ log₈₉(4) ⌋
- If A=ₑ (natural log):
  - ⌊ logₑ(99) ÷ logₑ(4) ⌋
  - ⌊ 4.59511985013459 ÷ 1.38629436119891 ⌋
  - ⌊ 3.3146783100398 ⌋
  - 4
- Base 4 answer still needs 4 digits.
Conversion Method #1

• Write out “enough” powers of the target base to cover the job, but no more,
• Distribute values from base 10 number to largest digit (without exceeding maximum value in target base),
• Repeat with next smallest digit, continue,
• After filling out rightmost digit there should be nothing left.

99\textsubscript{10} to Base 4 (should be 1203\textsubscript{4})

• Write out “enough” powers of 4, compute:
  \[ \ldots \quad 4^5 \quad 4^4 \quad 4^3 \quad 4^2 \quad 4^1 \quad 4^0 \]
  \[ \ldots \quad 4096 \quad 1024 \quad 256 \quad 64 \quad 16 \quad 4 \quad 1 \]
  \[ \text{Don’t need anything bigger than 64 (four digits)} \]
• Distribute:
  – How many 64 are in 99? \( 1 \), with 35 left over,
  – How many 16 are in 35? \( 2 \), with 3 left over,
  – How many 4 are in 3? \( 0 \), with 3 left over,
  – How many 1 are in 3? \( 3 \), with 0 left over.

430\textsubscript{10} to Base 16 (should be 1AE\textsubscript{16})

• Write out “enough” powers of 16, compute:
  \[ \ldots \quad 16^4 \quad 16^3 \quad 16^2 \quad 16^1 \quad 16^0 \]
  \[ \ldots \quad 65536 \quad 4096 \quad 256 \quad 16 \quad 1 \]
  \[ \text{Don’t need anything bigger than 256 (three digits)} \]
• Distribute:
  – How many 256 are in 430? \( 1 \), with 174 left over,
  – How many 16 are in 174? \( 10 \), with 14 left over,
  – How many 1 are in 14? \( 14 \), with 0 left over.
  – Replace digits > 9 with letters: \( 1AE \)

Conversion Method #2

• Divide decimal number by the target base, save remainder,
• Repeat process with quotient,
• Stop when quotient is zero.
• Sequence of remainders is answer in right-to-left order.

99\textsubscript{10} to Base 4 (should be 1203\textsubscript{4})

• \( 99 \div 4 = 24 \text{ R } 3 \),
• \( 24 \div 4 = 6 \text{ R } 0 \),
• \( 6 \div 4 = 1 \text{ R } 2 \),
• \( 1 \div 4 = 0 \text{ R } 1 \). \quad \text{(Quotient=0 means stop!)}
• Result = 1203\textsubscript{4}.

53\textsubscript{10} to Base 2 (should be 110101\textsubscript{2})

• \( 53 \div 2 = 26 \text{ R } 1 \),
• \( 26 \div 2 = 13 \text{ R } 0 \),
• \( 13 \div 2 = 6 \text{ R } 1 \),
• \( 6 \div 2 = 3 \text{ R } 0 \),
• \( 3 \div 2 = 1 \text{ R } 1 \),
• \( 1 \div 2 = 0 \text{ R } 1 \). \quad \text{(Quotient=0 means stop!)}
• Result = 110101\textsubscript{2}. 
430\textsubscript{10} to Base 16 (should be 1AE\textsubscript{16})

- 430 ÷ 16 = 26 R 14, (replace 14 with E),
- 26 ÷ 16 = 1 R 10, (replace 10 with A),
- 1 ÷ 16 = 0 R 1, (Quotient=0 means stop!)
- Result = 1AE\textsubscript{16}.

Special Rules for Converting Between Base 2 (Binary) and a base which is a Power of Two (Base 4, Base 8, Base 16, etc.)

Back to Bytes

- Decimal values 0...255,
- Binary values 00000000...11111111,
- Hexadecimal values 00...FF.
  - \( F \times 16^1 + F \times 16^0 = \)
  - \( 15 \times 16 + 15 \times 1 = \)
  - \( 240 + 15 = \)
  - \( 255 \)
- One Byte is exactly two hexadecimal digits!
- Therefore, each hex digit is exactly 4 bits.

Base 2 to Base \(2^N\) for any \(N\)

To convert a binary number to any base which is a power of two:
- Start from the right...
- ...and partition the binary number into packets of \(N\) bits per packet,
- If leftmost packet contains fewer than \(N\) bits, pad on left with 0s, and then...
- ...convert each packet as a separate problem.

Base 2 to Base 16 (Hexadecimal)

- Binary number \(10110101101111\),
- Partition into groups of 4 bits (\(2^4 = 16\))
  - 0010 1101 0110 1111
- Convert each packet separately:
  - 0010 = 2, 1101 = 13 = D, 0110 = 6, 1111 = 15 = F
- Hexadecimal result is \(2D6F\)
- Which would you rather try to remember?
- They are the same value!

Base 2 to Base 8 (Octal)

- Binary number \(10110101101111\),
- Partition into groups of 3 bits (\(2^3 = 8\))
  - 010 110 101 101 111
- Convert each packet separately:
  - 010 = 2, 110 = 6, 101 = 5, 101 = 5, 111 = 7
- Octal result is \(26557\)
- Which would you rather try to remember?
- They are the same value!