Hexadecimal

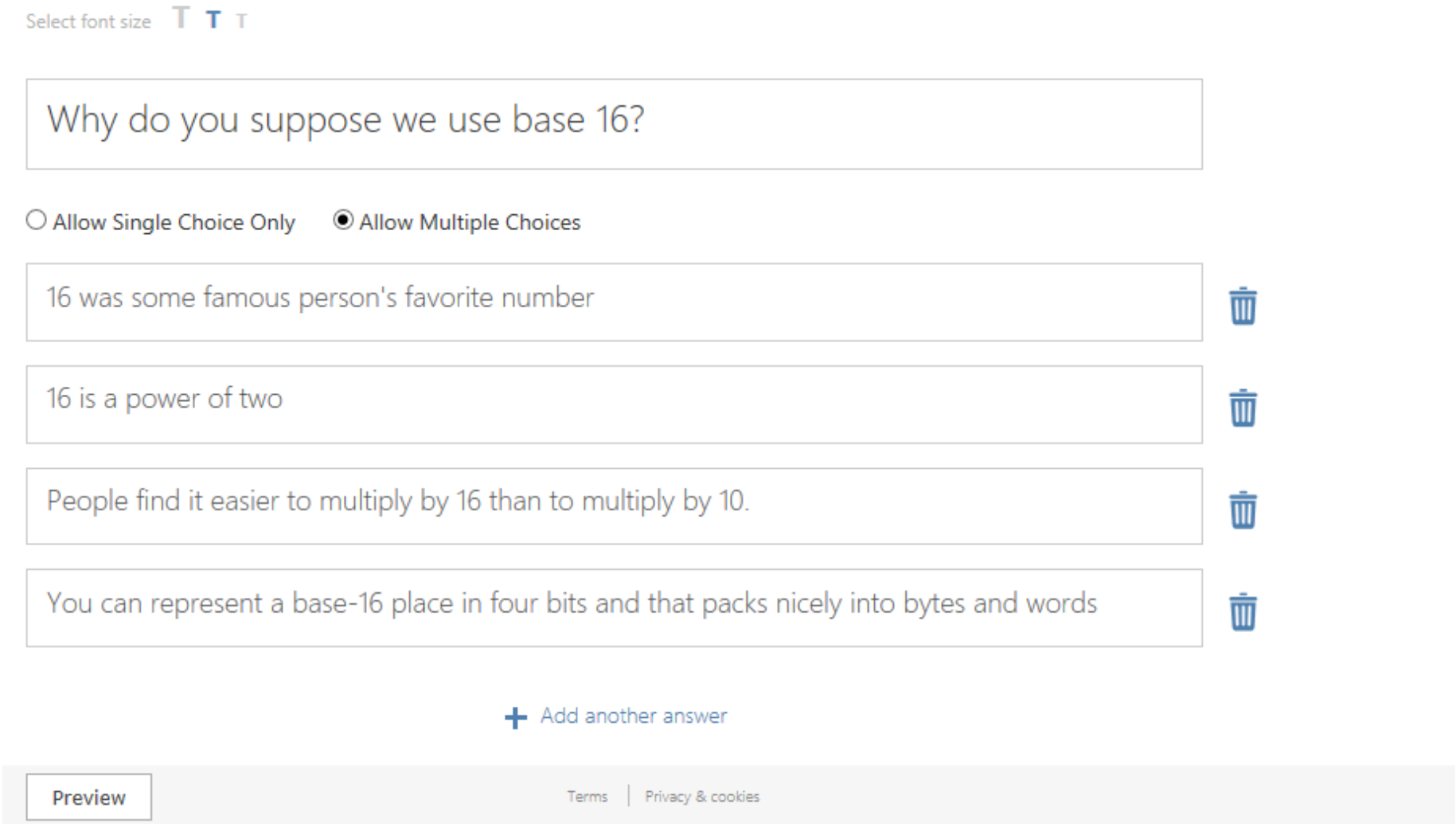
- Learning Objectives
  - Convert small binary numbers to/from decimal.
  - Convert binary to/from hexadecimal.
  - Convert (small) numbers from hex to/from decimal.
  - Recognize common hex numbers.
  - Look at a hex number and have a sense of where it is in the address space.
  - Add/subtract hexadecimal numbers.

Internal Representation

- Computers store all information as sets of binary digits (i.e., bit).
- Binary digits take on the value of either 0 or 1.
- 8 bits comprise a byte.
- For the past 20 years or so the basic word size of most computers was 32 bits; today newer machines have a word size of 64 bits.
- For most of this class, we will focus on 32 bit numbers.
- Looking at a string of 32 bits is somewhat overwhelming and not a great way of transmitting information.

Hexadecimal

- Hexadecimal (hex) is base 16 representation.
  - We use the letters A-F as the extra digits, so we count:
    - 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10 …
  - Binary is base 2
  - Decimal is base 10
  - So, a decimal number has a one's place (10^0), a ten's place (10^1) and a hundred's place (10^2), …
  - Binary has a one's place (2^0), a two's place (2^1), a four's place (2^2), an eight's place (2^3), …
  - Finally, hex has a one's place (16^0), a sixteen's place (16^1), a two-hundred-fifty-six's place (16^2), …

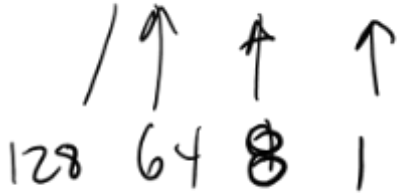

Select font size T T T
Why do you suppose we use base 16?
Allow Single Choice Only
Allow Multiple Choices
16 was some famous person's favorite number
16 is a power of two
People find it easier to multiply by 16 than to multiply by 10.
You can represent a base-16 place in four bits and that packs nicely into bytes and words
Add another answer
Preview
Terms
Privacy & cookies

Reading Binary

- You should become comfortable reading binary numbers up to 255.
  - It's easy and fun and will become second nature.
  - All you need to be able to do is add combinations of:
    - 1, 2, 4, 8, 16, 32, 64, and 128.
  - And you can learn to count to 63 on one hand.
- Just like with decimal numbers, the least significant digit (or least significant bit) is on the right, e.g., in the number 123, the rightmost digit (3) is in the one's place, etc.
- So, the binary number 0101 has a 1 in the 1's place and the 4's place, and therefore has the value 5.

A couple of binary examples;

- 1111 = 8 + 4 + 2 + 1 = 15
- 1000 = 8 + 0 + 0 + 0 = 8
- 1001 = 8 + 0 + 0 + 1 = 9
- 11001001 = 128 + 64 + 0 + 0 + 8 + 0 + 0 + 1 = 201



A couple of binary problems

- Pause the video for a moment and convert each of these binary values to decimal:
  - 11111111
  - 11110000
  - 00001111
  - 00110011
  - 01010101

A couple of binary problems

- Pause the video for a moment and convert each of these binary values to decimal:
  - 11111111 = 255
  - 11110000 = 240
  - 00001111 = 15
  - 00110011 = 51
  - 01010101 = 85

Converting from Binary to Hex

- Mapping binary to/from hex is easy!
- Binary to Hex
  - Group the bits in sets of four
  - Convert each set of 4 to a hex digit
  - Example:
    - 10001101010101000101
    - 1000 1101 0101 0100 0101 -- Group into sets of 4 bits
    - 8 D 5 4 5 -- Convert each set into a hex digit

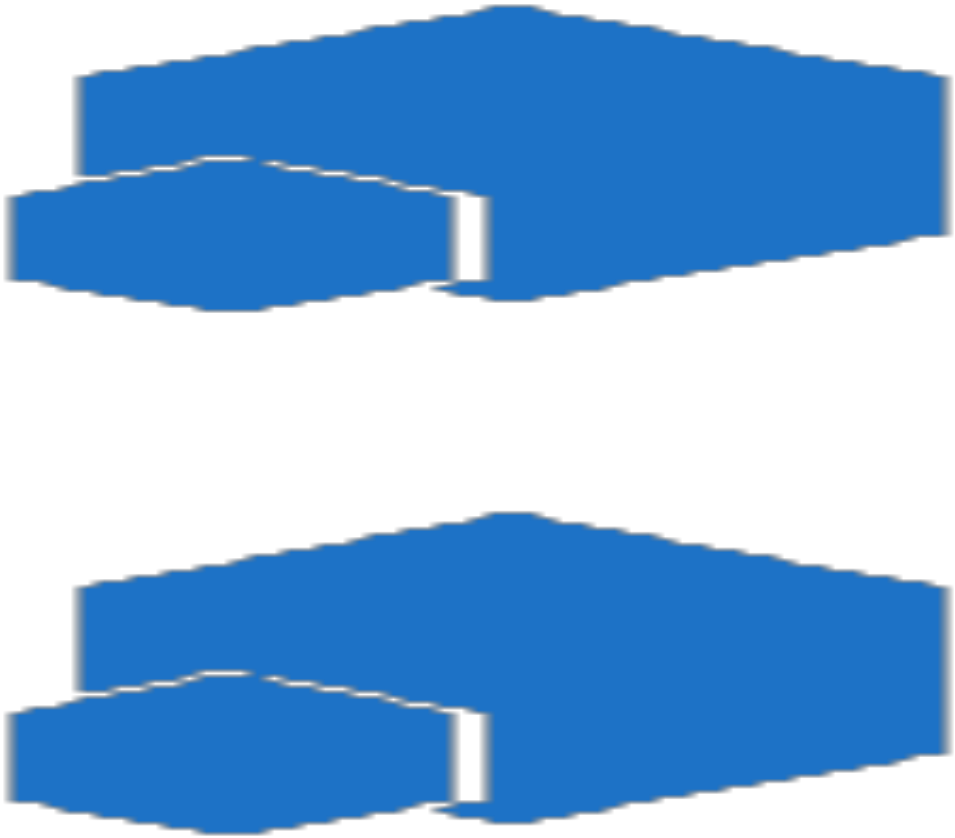
A = 10
B = 11
C = 12
D = 13

0x8D545

- Hex to Binary
  - Convert each hex digit into 4 bits
    - 0xDEADBEEF
    - 1101 1110 1010 1101 1011 1110 1110 1111

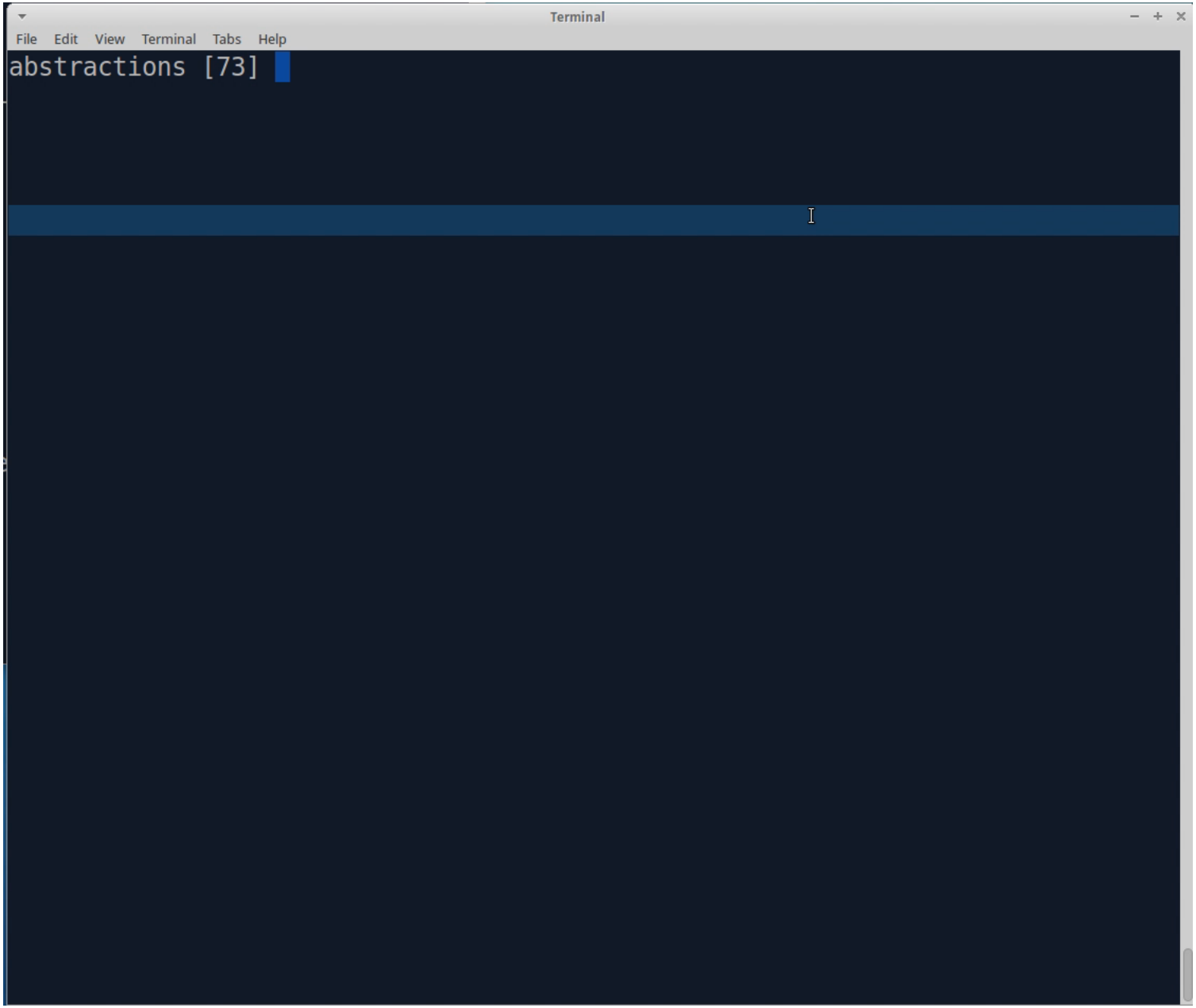
D E A D B E E F

A couple of sample problems



A handy tool

- The `bc` program, available on most Unix (i.e., Linux, OSX) machines, is a handy tool for converting to/from different bases.
- You can set the base from which you want to convert (input base): ibase = 2
- And the base to which you want to convert (output base): obase = 16
- And then `bc` will convert for you.
- Nonetheless, you'll want to be able to do small numbers yourself.


Terminal
File Edit View Terminal Tabs Help
abstractions [73]