Data Storage and Number Bases

4 Nov 2009
CMPT140
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

Data storage and I/O

- As programmers, you're already expert users of various datatypes and file I/O
- Now we peek under the hood to see what the compiler and the OS are really doing to implement these
- Every variable we declare takes up space in memory (RAM):
  - How much space does each variable need?
  - How is our data stored?

Binary numbers

- At the lowest level, all computer data are stored using logical bits: each bit can be either 0 or 1
  - High voltage (1) vs. low voltage (0)
  - Most memory chips use a big bank of tiny capacitors: has charge (1) vs. no charge (0)
- We use groups of bits to represent data (numbers, characters, strings, etc.):
  - e.g., this pattern of eight bits: 0 1 0 0 0 0 1 1
    - Could represent the decimal number 35
    - Or it might represent the character "#"
    - Or something else – depends on our interpretation

Number bases

- God gave us 10 fingers; so we often count in base 10:
  - "5927" interpreted as a decimal number:
    - 5 units of (10^3 = 1000)
    - 9 units of (10^2 = 100)
    - 2 units of (10^1 = 10)
    - 7 units of (10^0 = 1)
- Counting in binary is similar:
  - "0110" interpreted as a binary number:
    - 0 unit of (2^3 = 8)
    - 1 unit of (2^2 = 4)
    - 1 unit of (2^1 = 2)
    - 0 unit of (2^0 = 1)

Hexadecimal, octal

- Hexadecimal is base 16: we use 'A'..'F' to represent the "digits" ten, eleven, twelve, etc.
  - "BEEF" as a hexadecimal number:
    - B (11) units of (16^3 = 4096) => 45056
    - E (14) units of (16^2 = 256) => 3584
    - E (14) units of (16^1 = 16) => 224
    - F (15) units of (16^0 = 1) => 15
    - Total: BEEF (hex) => 48879 (dec)
- There's also octal, base 8:
  - only the digits 0..7 are used

Using bases in Python

- Python has special notation for expressing integer literals in hexadecimal and octal (no function is required!)
  - Hexadecimal: prefix “0x”

```
hexNum = 0xBEEF      # 48879
```

  - Octal: prefix “0”

```
octNum = 0115      # 1(8^2) + 1(8^1) + 5(8^0) = 77
```

- Convert into strings with hexadecimal/octal notation:

```
hexStr = hex(48879)      # '0xbeef'
octStr = oct(77)         # '0115'
```

Bits, bytes, nibbles, words

- One hexadecimal digit can be represented by four bits: one nibble
- Two nibbles (eight bits) is called a byte
  - One byte can be used to store one CHAR
- A group of bytes can be used to represent one datum: this is called a word
  - Most CPUs use 4-byte words (32 bits)
  - Newer CPUs can use 8-byte words (64 bits)
  - Word is the unit of data the CPU operates on

Accessing memory

- A computer's main memory (generally, RAM) stores everything it needs to do its current tasks
- A location within memory is uniquely identified by its address
  - Modern 32-bit OSes use 32-bit words to store memory addresses
  - So 2^{32} unique memory addrs (address space)
  - If each location stores one byte of data (*byte addressable*), then there is 2^{32} bytes = 4GB of addressable memory

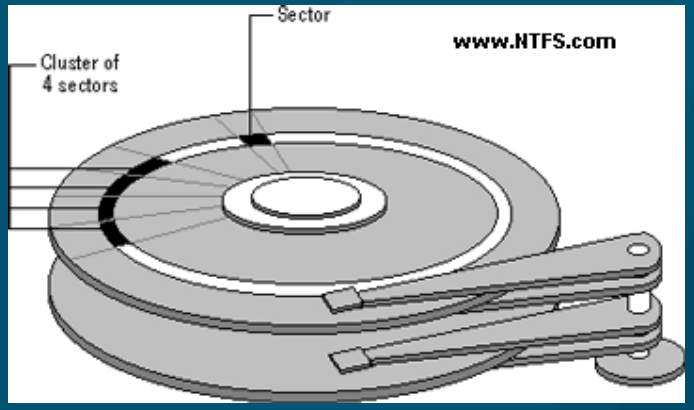
Units of measure

- SI abbreviations:
  - K = kilo = 1,000
  - M = mega = 1,000,000
  - G = giga = 1,000,000,000
- But when working with binary data:
  - KB = “kibibyte” = 1,024 bytes = 2^{10} bytes
  - MB = “mebibyte” = 1,024,576 = 2^{20} bytes
  - GB = “gibibyte” = 1,073,741,824 = 2^{30} bytes
  - But hard drive manufacturers use SI abbrevs

Units of measure, cont.

- Kilobytes vs. kilobits:
  - KB = kilobyte = 1,024 bytes = 8192 bits
  - Kb = kilobit = 1,024 bits
  - RAM chip manufacturers often use kilobits
- Also, in SI abbreviations,
  - M = mega = 10^6: e.g., megawatt = 10^6 watt
  - m = milli = 10^{-3}: e.g., milliwatt = 10^{-3} watt
- But not everyone is consistent, so be careful

Storage



- A page is a unit of memory used by a program
  - Often 4KB, but changeable
- A block is a unit of disk storage, often 512 bytes
- Hard disks are made up of platters, accessed by magnetic heads on movable arms
- Cylinders are concentric tracks across platters
- Hard drive geometry is traditionally expressed in C/H/S: cylinders / heads / sectors-per-track
  - E.g., 1024/255/63 (512B/block) = 8.4GB
  - Now use LBA: just use one 32-bit addr

For more information

- SI prefixes for binary multiples (e.g., mebibit)
  - http://physics.nist.gov/cuu/Units/binary.html
- The “Starman's Realm” on hexadecimal:
  - http://thestarman.pcministry.com/asm/6to64bits.htm
- Wikipedia entry on cylinder/head/sector:
  - http://en.wikipedia.org/wiki/Cylinder-head-sector