

# Technical concepts

Comp 199

# Admin

- Given I just started:
  - If there is anyone with a letter please see me.
- Google article.
- On the challenges of an always on digital world:
- <http://www.cnn.com/2013/07/12/tech/social-media>
- <http://tv.msnbc.com/2013/08/13/pre-trial-hearings->
-

# What is a computer?

- At its heart – what is a computer?

# What is a computer?

- At its heart – what is a computer?
  - A great big calculator with fancy input/output.
  - All it does is manipulate numbers.
  - Interpretation of those numbers is everything.
    - 01100110 – what is this?

# What is a computer?

- At its heart – what is a computer?
  - A great big calculator with fancy input/output.
  - All it does is manipulate numbers.
  - Interpretation of those numbers is everything.
    - 01100110 – what is this?
    - Integer? : 102
    - The single character 'f' (lower case f)
    - The last byte of the floating point number 27.3?
  -

# Binary

- Computers (all electronic computers) store data using electricity
  - Two states electricity on/electricity off
  - Interpreted as 1 / 0 respectively.
  - Long term storage magnetic, but used same 1,0 system.
  - So computers count and do arithmetic in binary (base 2)
  - Why do we use decimal (base 10)?

# Base 2

- Every time you count to two in your current “place value” (multiply by two) you need a new “place-value”
  - In decimal every time you get to ten you need a new place value: 1, 10, 100, 1000 etc
  - In base two you have a new place value at
    - 2, 4, 8, 16, 32, 64 etc
    - So what integer is 00001010?
    - How about 01010101
  - I'm using 8 'bits' because an 8bit BYTE is the smallest unit of data most computers recognize

# A little bit of text in Binary

- What is the phrase “BSU Comp 199: Information Security” in binary?

—



# A little bit of text in Binary

- What is the phrase “BSU Comp 199: Information Security” in binary?

– 01000010 01010011 01010101 00100000  
01000011 01101111 01101101 01110000  
00100000 00110001 00111001 00111001  
00111010 00100000 01001001 01101110  
01100110 01101111 01110010 01101101  
01100001 01110100 01101001 01101111  
01101110 00100000 01010011 01100101  
01100011 01110101 01110010 01101001  
01110100 01111001

# Hexadecimal

- Fortunately no one ever uses pure binary any more
  - Except electrical engineers
  - Use hexadecimal instead. (Base 16)
  - Each of those 8 bit bytes are a 2 digit hex number.
  - New place for every 16
  - Represent hex number “10” (single digit in base 16) as 'A', 11 as “B” and so on till 15=F
  - Fortunately don't have to memorize all of hex, tools exist which will show you hex and its equivalent characters.

# A bit of text is Hex

- Same phrase: “BSU Comp 199: Information Security “ in Hex:
  - 42 53 55 20 43 6f 6d 70 20 31 39 39 3a 20 49  
6e 66 6f 72 6d 61 74 69 6f 6e 20 53 65 63 75  
72 69 74 79

# Character mappings

- Since most people don't read binary/hex and don't want to
  - Need a way to represent readable characters
  - Original Method: ASCII

# Ascii and Character Sets

- American Standard Code for Information Interchange
  - original universal character set for computing
  - developed in early 1960s
  - mandated for all computers sold in US by Johnson in 1968
  - remained dominant till mid-1990s
  - 7-bit character set
    - 127 total characters possible
    - see printed chart of ascii characters

# Ascii and the world

- US was largest consumer of computers in world in 1960s-1980s.
  - ASCII on all of them
  - many seminal programming languages developed in that time
  - what is the side effect?

# Ascii and the world

- US was largest consumer of computers in world in 1960s-1980s.
  - ASCII on all of them
  - many seminal programming languages developed in that time
  - what is the side effect?
    - English became the de facto language for programming for a long time.
    - and for computer use
  - issues?

# Computing and culture

- Today
  - unicode (UTF-8) is standards
    - superset of ASCII with 32,767 possible characters.
    - nearly all of worlds languages have characters represented.
  - All character sets map numbers stored in computer to an actual character.
  - Viewers allow hex data to be shown side by side with the character equivalent
  - (for text files)
  - Text vs Binary files. - list on board.



# Files

- So the computer is a massive calculator storing lots of ones and zeros.
  - But humans work with files
  - Files basically strings of bits and bytes at specific points on disk – often with special series of bytes at the beginning and end.

# File Identity

- How does the computer know how to interpret the file?

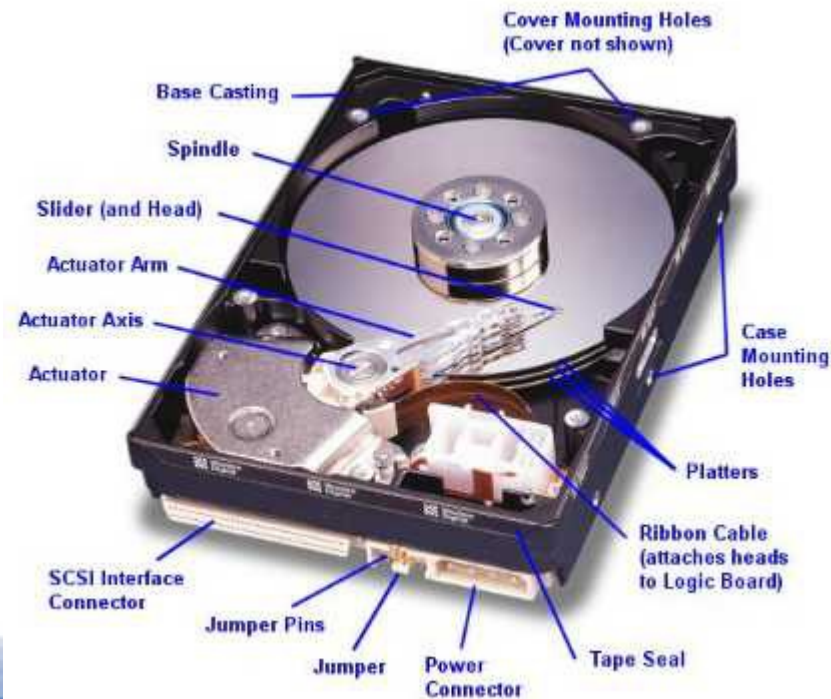
# File Identity

- How does the computer know how to interpret the file?
  - Method 1: using file extensions.
    - Windows hides file extensions by default
    - But relies on them completely.
    - Demo
  - Method1: File signature
    - Those special bytes at the beginning of each file form its “signature”
      - Tell anyone who asks what type of file it really is.
      - Demo using linux/mac command line.

# Storage 1: Traditional Hard Drive

- Stores data magnetically
  - See pict
  - Spark (static electric) does major damage to data
  - Magnetic charges still have residue even after overwrite
  - Moving parts lifespan ~3 years

- Image from <http://www.chicagojamscene.com>



# Storage 2: SSD

- Solid State Drives (flash drives)
  - Use electricity – but not lost when power turned off
  - Getting to be more and more common
  - No moving parts, ave lifespan 4ish years
    - Much faster than HDD
  - Downside for digital forensics
    - When the data is written over – it is gone

## Storage 3: Optical Storage

- Optical Storage: CDs DVDs Blu-Rays etc
- Thin layer of metal between two layers of hard plastic
  - Archival quality lasts for 25-50 years
  - Standard burn disks last about 10 years.

# Storage 4: Tape drives

- Tape backup drives
  - Slow, large capacity magnetic tape backup
  - Still used very commonly till about 4-5 years ago
    - Why?
  - What is used now?

# Memory vs Storage

- Distinction in terms: Memory vs Storage
  - Your book glosses over this
  - Memory: volatile memory – the working memory of the computer
    - Gone when power is off.
    - Like your short term memory/brain space
  - Storage: long term storage
    - Retained when power is off
    - Where your programs and data are saved
  - Out of memory error can only be solved by closing programs, not by freeing up “disk” space.



# Computing Environments

- Stand alone computer
  - All data on that device
  - Very few (that matter) left these days
- Networked computer
  - Data could be on device – or on network,
    - much more complicated
    - Much more common
- Mainframes
  - Large centralized server type computer – all data and processing centralized
  - Still common for some large business functions

# Computing Environments II: Cloud

- Cloud Computing
  - More and more common today:
  - Examples?

# Computing Environments II: Cloud

- Cloud Computing
  - More and more common today:
  - Examples?
    - iCloud
    - Google apps
- Two types that matter to us: (different from book)
  - Storage on cloud
  - Programs on cloud.
  - Common sense issues for digital forensics?

# Data types Jargon

- Three types of data
  - Active (the computer knows about it and is using it)
  - Latent (the computer freed this up, but hasn't overwritten it yet so it can be retrieved with correct tools)
  - Archived
    - Backed up on tape or other archival media.
    - Book warning about obsolete formats
      - Like what?

# File Systems Terms

- Three most common on desktops
  - FAT (File allocation table)
    - Old system used in DOS and updated over time
    - Little or no protection for files
    - Used for inter-operating system compatibility.
  - NTFS
    - Used in modern versions of windows
    - Non-case sensitive
    - More protections than FAT, less than some others.
  - HFS and friends
    - Used in modern versions of MacOS
  - EXT3: default used for linux.

# Storing files on Magnetic drives

- Walk through diagrams from book

# Reading