Handling Information

The Structure and Functioning of Computers and Networks

[an introduction]

Why are We "Getting Technical" Now?

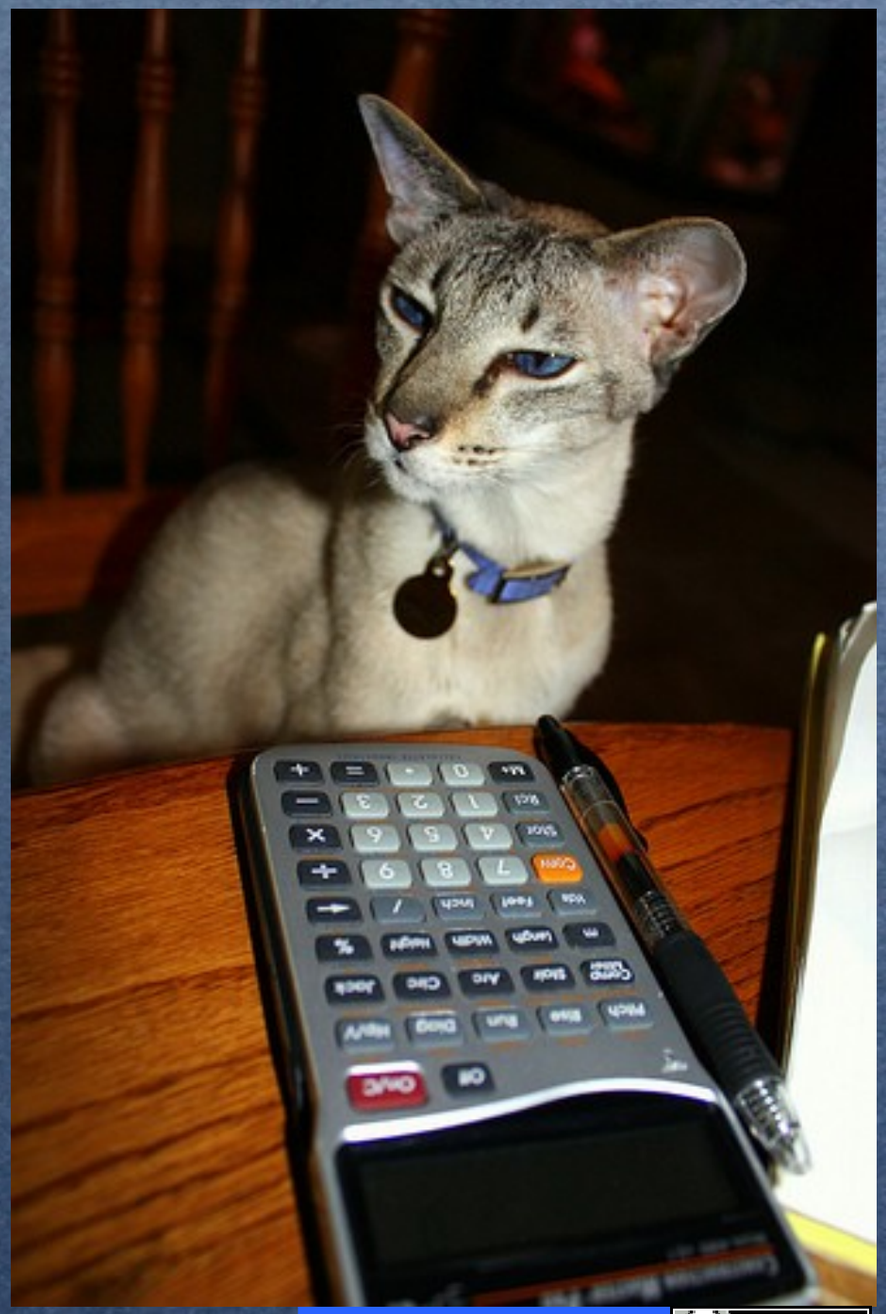
- Facing the IT revolution since about 1980, basic practices and rules of the game in information and knowledge delivery are transformed, and traditional practices are rendered obsolete
- In order to understand the new terrain we need to know some basics about IT, networks, and communication infrastructures
- This will be tough for some, boring for others, so please let us know
- A few questions, in an informal poll:
  - How many know what an API is?
  - The difference between bitmaps and vectors?
  - The concept of "abstraction layers"?
  - What a BIOS is?

Goals of This Module

- How computers and networks are structured and how they operate: critical cost issues for deployments
- How those structures inadvertently create "bottlenecks" that can be exploited by the greedy or power-hungry
- The importance of technological standards in terms of serving users and focusing innovation
  - "open" (or expert) standards vs. proprietary standards
  - [*not* the same as "open source," which we also address]
- A brief view of emerging possibilities in computing and networks
  - "cognitive communities"
  - emergent machine intelligence: computers "thinking" on their own
  - a systematic creation of a virtual world parallel to the "real world"

Defining Digital

- A world of “toggles”: differences in kind
  - yes/no and the spin-outs from truth tables
- Compare to analog: differences in degree
  - Sound
  - Language
  - Images
  - How the brain “fills in” “missing information”
- How robust? How scalable? How replicable?
  - Compare LPs to CDs
  - Pattern recognition



Can your PC identify this guy?

Ad/disadvantages of digitality

Precise, reproducible, well-defined

vs.

Non-linear, elusive, busy, poor at generalities & interpretation;
have to sample and reconstruct to approximate continuity

Computers and Brains: A Spurious Comparison?

- Analogous, or separate but equal?
- The failed promises of “artificial intelligence”
  - The “Turing test”
  - Agenda adaptation to “intelligent agents”
- Next-generations computing better?—“fuzzy,” quanta, parallel processing, multiple modes…

The Current "Laws"

- Moore's Law on transistor density
- Metcalf's Law on network effects
- [Frost's Law on forces of habit ;-} --but the real issue of legacy systems and practices, but (we hope) not people]

Hardware & Software

- Hardware: CPUs, memory, drives, peripheral devices (I/O)
- Software: Operating systems, applications, “middleware;” IAC, etc.
  - Application Programming Interfaces (APIs) and process-communication protocols
  - Machine language and source code
- Sometimes the distinction is blurred: ROMs used in old game machines

Basic Computer Architecture: Abstraction Layers

Distributed Processing Systems
[Grid systems, Beowolf, server farms, etc.]

Middleware
[Java, XML-family, Web Services, .NET, etc.]

Applications
[e-mail, word-processing, browsers, Kaaza…]

APIs

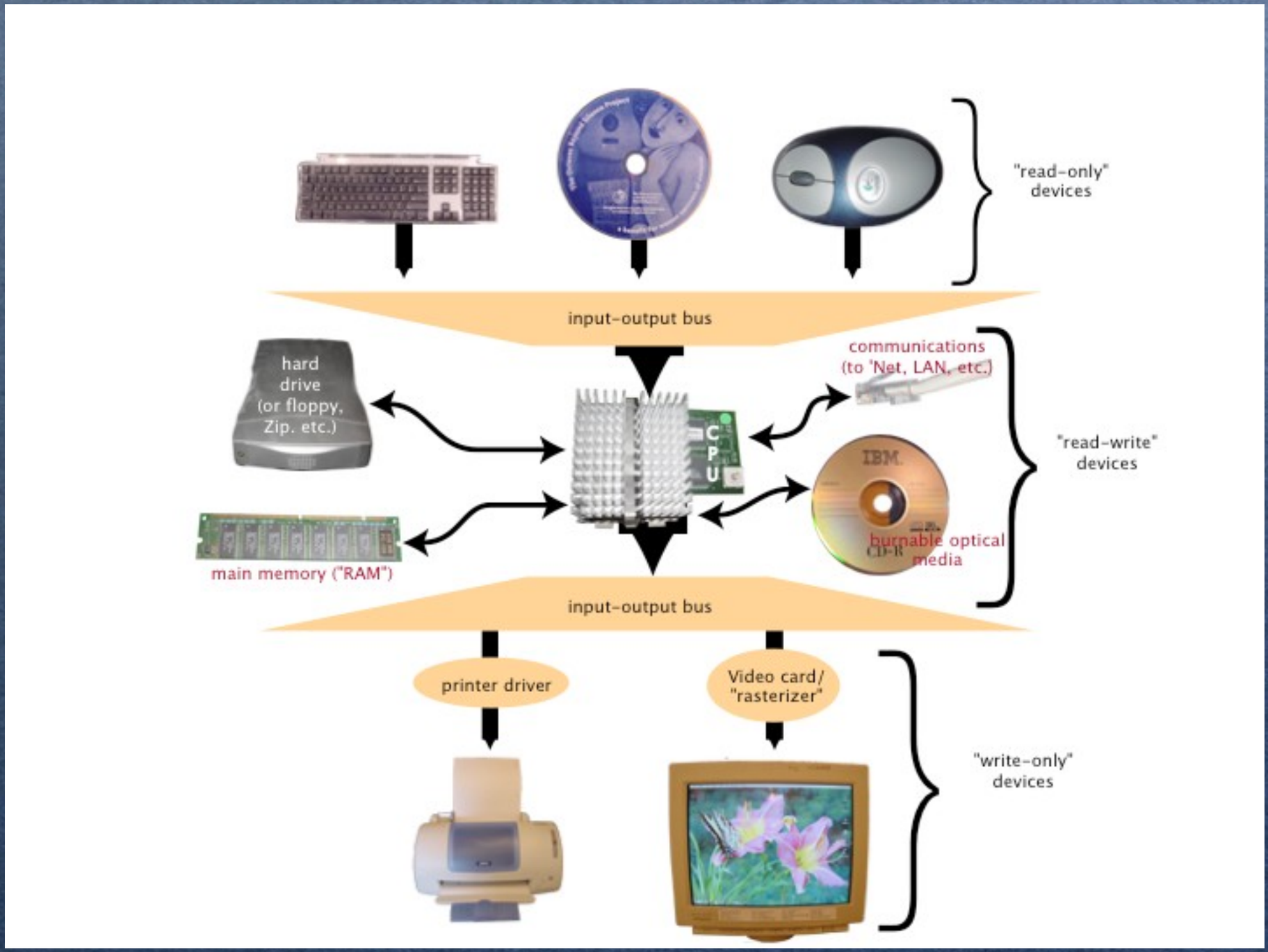
"Patched-in" communications layer [legacy]

Operating system (Unix, MacOSX, Windows)
and hardware device drivers

Basic Booting Layer: BIOS
(basic input-output system)

hardware addressing

Basic Computing Hardware



Problems of Standards

- Component vs. monolithic systems
- Proprietary vs. open
  - DOS/Wintel and Apple
  - Unix, Linux, and open-source
    - historical irony of the IBM PC
- Perils of improper timing in standard-setting
- Proprietary standards and implicit monopolies
- Conflicts in purposes
  - “network” machines vs. stand-alones
  - Cost and diffusion issues
  - Divergent business models: Xerox/Wang/Apple approach vs. Dell

Breaking News on Standards!

- In the third week of September 2005, the State (Commonwealth?) of Massachusetts issued a new policy: all software used by state government must read and write to an open, non-proprietary format
- This means:
  - Massachusetts affirms the OASIS standard set for open document format standards
  - Massachusetts will soon be no longer "locked in" to Microsoft's proprietary formats, freeing it to use less costly software
  - Of course, Microsoft is livid…
- FYI, remember that there's a difference between "open standards" and "open source"

Hardware I: The CPU

- Carrier waves and Hz ratings
- Bus widths (in bits) [bits vs. Bytes]
- Registers, caches and memory available to processors
- Single- vs. multiprocessors
- Pipelines and predictions

Hardware II: [active] Memory

- RAM vs ROM
  - Loading to RAM vs. reading from ROM (PCs vs. game consoles)
- Earlier types of memory: ferrite donuts
- Memory costs
- Memory (and bus) speed as a constraint
- Virtual memory

Hardware III: Addressing

- Logical vs. physical addresses
- Locality annihilated—to a point
- Memory and storage mapping: directories, etc.

Hardware IV: Storage

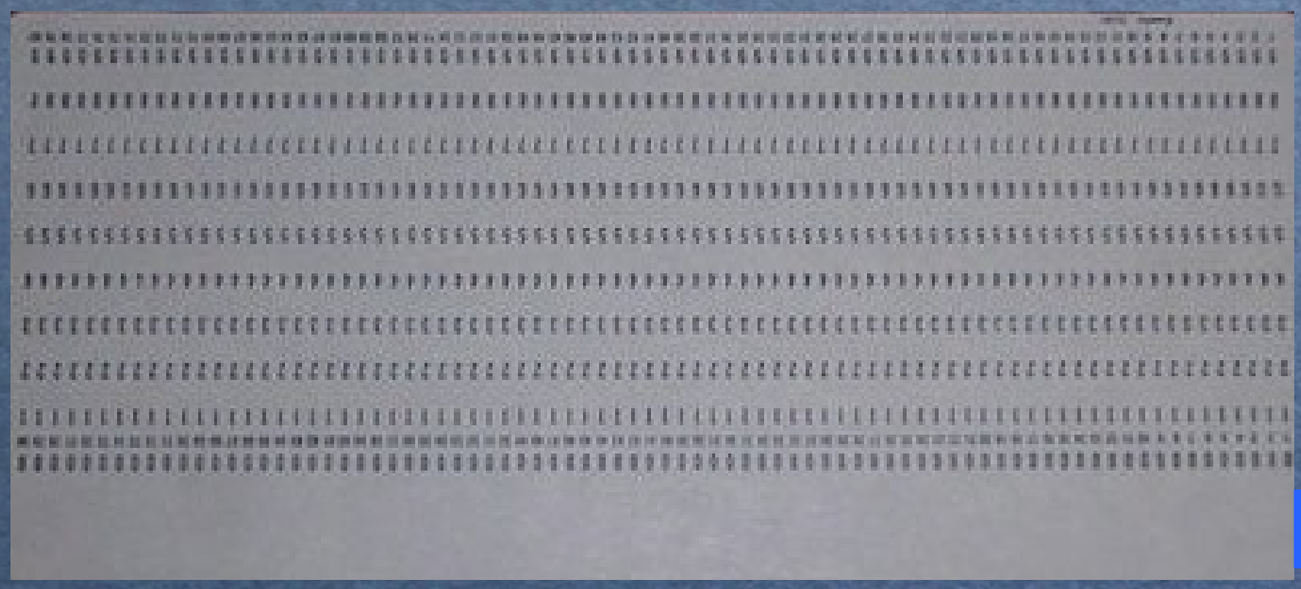
- Types:
  - Tape, floppy, M-O, laser-based disks, RAM disks
- Speed & purposes
  - Immediate, short-term, and long-term
  - Cost constraints
  - [More on this with data preservation]

Hardware V: Input Devices

- A/D converters
  - Sound
  - CCDs: scanners, cameras
  - Perils of sampling and problems of pixellization
  - Voice-recognition (and making it robust!)
- Direct-input devices
  - Punch-cards (for both data & commands)
  - Paper tape
  - Mice, keyboards

How Much Easier and Faster it all is Now…

Did you ever wonder (probably not!) how many punch cards



would be needed to store a 3-minute, 128 bps .mp3 music file?

Give up?

Try 36,864 (twenty+ cartons, at about 10 pounds each),
and your card-reader would have to process 205 cards per second!

Hardware VI: Output Devices

- Display: paper/[ticker!] tape to monochrome, to color
  - Resolution and the problem of bit-mapping
  - Ripping defined
- Printing: vectors and bitmaps [lineprinters/LPS]
- Burners, D/A processors, sound & video
  - Issues of encoding, encryption, and compression
  - Hardware algorithms

Software I: Basic Architecture

- Step 1: operating systems vs. applications
  - ["Traditional" PC-era] distinction; current example: Windows™ as an operating system, MS Word™ as an application
  - It blurs!
    - Mainframes (1950s-80s): complete systems/apps
    - 1969-present: Unix "services" used by apps, supplied by OS
    - 1984: Mac Toolbox—"widgets" used by apps, supplied by OS
- Reality is Step 2: Layers and abstractions
  - Typical: kernel, extension, drivers in Unix
  - Emerging
    - layers (both local and through networks) with coherently addressable APIs
    - networked, cross-platform, distributed applications: "Grid"

Software II: Types of Applications

- Words, texts, and characters
- Pictures, frames, and sounds
- Typographical and page-layout
- Databases, statistics, spreadsheets
- Place-based systems & others
- Network, distance-linking, & collaboration applications

- In a networked world, means and modalities of exchange: STANDARDS

Software III: Strategic Positions

- Controlling the APIs or layers: bottlenecking (Microsoft)
- In networked computers, issues of security
  - What is an “executable”? (problems with macros)
  - What’s an open port, an open relay?
- Proprietary vs. “open-source”
  - Bureaucracy, organization, and innovation
  - Irony: more openness means more security(?)
- (More on this when we cover info economics & business)

Computing Meets Communications: The Internet & Beyond

- Comparing and contrasting POTS and packets
  - When women were switches…
  - A data network able to withstand nuclear war(!)
  - DARPA, Metcalfe, and packet-switching
- Wires and fibers, LANS and WANS
  - Rings and Appletalk, to client-server, to swarms
  - "Thin clients," WiFi, Bluetooth, and 3G phones; security issues

From the Internet to the Web

- Bitnet, telnet, NSFNet, ftp: backboning with TCP/IP, routing
- Archie, Veronica, and Gopher and the smart Net
- T. Berners-Lee and the Web [of knowledge]
  - The logic of hyperlinking (what's 404?)
    - To other documents—an infomation-knowledge matrix?
    - Elegant simplicity of Hyper Text Markup Language
    - Live/executable documents (new "dashboards," GUIs, OSs? -- Microsoft and Netscape)
  - Knowledge as a matrix, problems of warranting
  - Distributed computing and cognition; evolving systems
  - Is the Net becoming a new "life form"?

The New Information Environment

- Distributed knowledge and fact overload
  - Data mining and knowledge locating: off-loading inference as well as deduction to the IT system
  - Google and the power of the search
  - The semantic Web
- Web Services & middleware
- Illusions of empowerment and mirrors of virtuality
- Cybercommunities, cyberliberation and cyberghettos
- Public, private, personal, and performative space on the Web.