HOW

Computers Work

CM1030

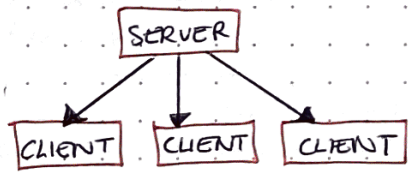
ABSTRACTION

An abstraction is where you highlight key information without going into all the detail. Or taking something to a higher level.

A notional machine is an abstraction of a computer.

How does a webpage work?

- ★ A server is where the information for the webpage is stored.
- ★ A client accesses that information either via us or via software eg browsers.



http://blahblah.com/computing/search=computer

the language — the webpage — the query

These days it is very rare for a webpage to be static, it is much more likely that the pages are created as we go. Usually off the back of a database.

Websites are typically modular with lots of different components working together eg video, pictures, flash plugins etc.

IP - Internet protocol

HTTP - hyper text transfer protocol

COOKIES - small amounts of data that keep track of the state we are in eg are we logged in? What is our browser history?

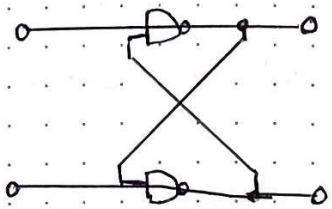
API - application programming interface. Kind of like a data access point that you can plug into. eg embedded google maps

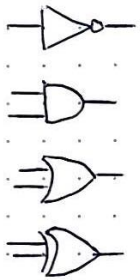
MODULARITY - having lots of different modules that you can plug into each other to build up the application eg printer drivers that can be used by the OS.

Data Representation

At its core the language a computer uses is binary. 1/0, on/off, true/false with a single digit being known as a BIT.

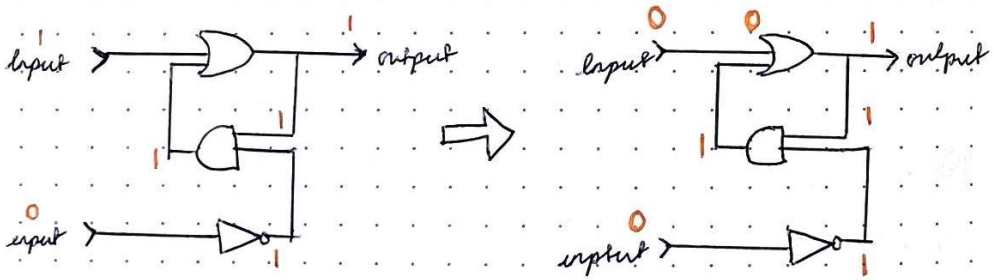
GATE can be constructed out of gears relays or optical devices. In a modern computer they are usually small electronic circuits where 1 and 0 is represented as voltage levels. A very important gate is a flip flop it accepts a 1 and 0 and will store that information until another stimuli is introduced.





looking at the flipflop gate, you pass a current through the upper gate and it will produce an output of 1.

The bottom gate at 0 will change to a 1 through the not gate. Which in turn means the and gate is producing a 1. This means that even if the upper gate is turned off the output remains a 1.



Temporarily changing the bottom input to 1 will force the output to be 0 until changed.

Circuit gates are important in CS because

- ★ It helps visualise digital circuit design and how they're constructed
- ★ Provides an abstraction for data storage
- ★ Provides an abstraction of the underlying circuitry

MAIN MEMORY

A computer contains a large number of flip flops capable of storing single bits within main memory.

We group these bits into 8, otherwise known as a byte and contains them within a cell. The left most bit of a byte is the 'high order' end (or the most significant) and the right most end is the 'low order' end (or least significant)

Each cell has a unique name known as its address

Because the cells can be accessed independently the main memory is usually called RAM (random access memory)

Kilobyte $= 2^{10} = 1{,}024$ bytes $= 1KB$
Megabyte $= 2^{20} = 1{,}048{,}576$ bytes $= 1MB$

RGB colour is made up of 3 byte strings or 24 bits per pixel (24 pixels x 3 bytes = 72 bytes)

MASS STORAGE

Due to the limited size and volatility of main memory most computers will have a mass storage, eg flash drives, cds, SSD and more. The advantages are low cost to storage size, less volatility and the ability to remove and archive them.

MAGNETIC - eg harddrive, where a thin disc with a magnetic coating is used to store information. The disc circumference has tracks that are split into sectors. And the read-write head positions itself over the sectors

OPTICAL - eg a cd, information is recorded on them by creating variations in the reflection which is read by a laser as the disc spins. CDs perform best when reading long continuous strings of data such as music due to the spiral design of the disc.

FLASH DRIVES - works by capturing electrons in small chambers of silicon dioxide, these are stored for many years without power and so perfect for stable non-volatile data storage. Though with repeated use the silicon loses its ability to capture electrons as effectively, in devices such as SSD that are meant to ~~play~~ replace harddrives the issue of wear is resolved by rotating the data.

REPRESENTING INFO

TEXT - this is coded into bits typically the most common words have shorter bit patterns when they go through compression. Standard letters use the ASCII (American Standard Code for Information Interchange). It uses 8 bit strings to encode the alphabet, punctuation and numbers.

There are extensions to ASCII that cover Ü, Ä etc but it became clear that there weren't enough variations to cover languages such as chinese, japanese etc. UTF-8 uses 24 or 32 bit patterns to represent these symbols.

NUMBERS - It requires more data to store numbers using ASCII so binary digits are used.

IMAGES - many images are made up of a collection of dots, each RGB of that dot is then encoded in a bit map. This is the most common method, Red, Blue Green and Luminiscence (essentially the sum of the RGB components).

This pixel bit map isn't great for scaling images (eg pixelated) and so images are sometimes described as a series of geometric shapes that are scaled in relation to each other. This is common in CAD

SOUND - is captured by recording the amplitude up to a rate of 44,100 times per second, each sample is 16 bits long.

MIDI is an alternative which looks more at what the sound should be and stores the average rather than constantly sampling.

COMPRESSION

Compression is either lossy (lose information) or lossless (information isn't lost)

IMAGE - There are a couple of different ways to compress images, GIF images there is a palette of 256 colours, this means each pixel can be encoded with 1 byte.

JPEG has several different methods of compressing pictures, the human eye is more sensitive to brightness than changes in colour so it first averages chrominance over 2x2 pixel squares. Then it takes 8x8 blocks and using discrete cosine transformation relates each pixel to it's neighbour rather than directly storing the information. It is then compressed using run length encoding

AUDIO AND VIDEO - the most common method of compression of video and audio is MPEG (motion picture experts group).

Video compression must cover a wide range of uses, video conferencing where speed is important vs HDTV where quality is important. The encoding works by encoding key frames fully and then referencing between frames to those fully encoded. Discrete cosine transformation is then used to further compress the images.

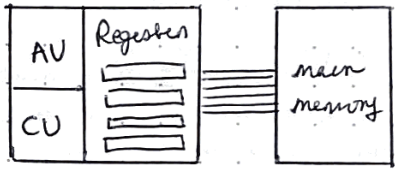
Mp3 is a common method of compressing audio and works by removing information that the human ear can't hear.

Computer Architecture

The part of the computer that controls the manipulation of data is the CPU or central processing unit.

A CPU consists of 3 parts

- Arithmetic/logic unit
- Control Unit
- Registers



THE BUS - a series of wires between the CPU and main memory that communicates bit patterns

THE REGISTERS - can either be general or special purpose. General registers hold information for the input and outputs that the arithmetic unit is currently working on.

CONTROL UNIT - transfers the data from memory into the registers and tells the logical unit where the information is and where the result should be stored

STEP1 - Collect value 1 from RAM store in register
STEP2 - Collect 2nd value and store in 2nd register
STEP3 - Activate addition logic on registers
STEP4 - Store the result in 3rd register
STEP5 - Stop
~~STE~~

Previously the arithmetic component the steps of the program were inflexible, until it was discovered that programs could be stored in main memory and the control unit could go and collect the information for the instructions from there, STORED-PROGRAM CONCEPT.

MACHINE LANGUAGE

Instructions are encoded as bit patterns and known as machine language. The types of instructions machines use are limited, adding additional functionality beyond the basics adds convenience but not additional capabilities

RISC - Reduced instruction set, small number of instructions. Fast efficient and cheaper

CISC - Complex instruction set, complex set of instructions, deals with complex software

DATA TRANSFER instructions, perform LOADS of data into registers, it is worth noting information is copied rather than transferred. As well as STORE instructions for results

I/O (input/output) are also considered under the data transfer header for communicating with peripherals

ARITHETIC/LOGIC instructions perform the operations eg OR, AND or arithmetic. It can also perform shift or rotate operations for moving information in the registers

CONTROL instructions direct the next step to be executed rather than controlling data. Can implement jumps to skips the next instruction either conditional (if this then that) or unconditional (go to next step)

example 1

156C register 5 = register 6C LOAD
166D register 6 = register 6D LOAD
3056 Add register 5 and 6 store in 0

0101 | 0011 0001 0010 - ADD | 3 | 1 3

op code — argument — store result — input locations

MACHINE CYCLE has 3 steps

1. FETCH
2. DECODE
3. EXECUTE

FETCH - CPU requests the data from main memory for the instruction stored at the location indicated by the program counter. CPU then places the instruction in the instruction register and then increments the program counter by 1.

DECODE - The CPU then decodes it into its component parts

EXECUTE - The programs then performs the instruction by LOADING data or engaging the logical unit to perform the calculation.

THEN REPEAT

PERIPHERALS

Communication between peripherals such as keyboard, mice etc is typically handled by a controller. Sometimes these controllers have their own basic memory and CPU. USB and firewire are examples of controllers that can handle multiple devices.

The controller connects to the bus and so has access to the CPU and main memory, and so the CPU can send instructions to the controller in much the same way that it would to main memory.

The controller can also act out instructions on behalf of the CPU that doubles the processing power on these moments the bus is available.

Operating Systems

Graphical User Interface is how the system communicates with it's users. Objects can be manipulated, program files and folders can all be displayed as pictures and input such as keyboards and mice can move them around.
The command prompt is another shell alternative to the main windows OS

WINDOW MANAGER is an important part of the GUI. It allocates blocks of the screen called windows, and the program associated with each window. It identifies where the mouse is when it is clicked and the program it is hovering over.

Kernel is the internal part of the operating system it contains the very basic functions of the operating system

FILE MANAGER - is the part of the kernel that helps manage mass storage
→ where the file is stored
→ what storage is available for use
→ which users are allowed to access what
The file manager allows users to group files into folders (directories) and create subdirectories called file paths.

DEVICE DRIVERS - is another component of the kernel this handles communication with the device drivers and sometimes the peripherals themselves

MEMORY MANAGER - coordinates the use of main memory and important functions in multitasking machines. It must assign each running program its space and ensure it can't access memory it's not supposed to
Paging is where the system needs more memory than you can actually have and so it creates the illusion of space by shuffling data between main storage and memory. This additional space is called virtual memory

Bootstrapping happens whenever the computer is turned on. The CPU program counter starts with a predetermined location whenever it is restarted, this is in main memory. But as main memory sucks at holding onto information this is pointed to a special part of main memory which it to load the operating system from mass storage.
This main memory is a form of flash storage.

The OS is stored in mass storage because it wouldn't be efficient to store it in RAM due to modern OS ~~stay~~ size. It is also needed to be updated regularly and this isn't straightforward in RAM.

Privileledged Instructions help interrupt malware from getting control over other programs data. When an attempt is made to run a privileged instruction when not in privileged mode an interrupt command is given. But the process is complex and open to exploitation

Admin User one way of having a secure system is to have users and accounts where one user cannot access the files of another on the same machine. The admin is the only one with overall access, the user identifies themself on logon.
The administrator can use auditing software to identify attempted access and any unusual behaviours

Co-ordinating Processes

A process is the act of running the program. You can think of the program as a piece of sheet music on a shelf and processing being the act of playing it.
While a musician plays one song after another, a program will play several at the same time.
The program being run is kept track of by the scheduler.

SCHEDULER - (part of the systems kernel) adds new processes and removes them when they are done. It keeps track of them in a block of information in main memory called the PROCESS TABLE.
The process table contains such information such as the processes priority, the memory location where the process is located. As well as if the process is READY to be run or WAITING for something to happen.

MULTIPROGRAMING - is where each program is divided into time slices the CPU runs the program for the allocated time slice and then is stopped by an interrupt signal and the next process time slice is run. The act of switching is known as process/context switch.
When the interrupt handler is run it:
→ Records where you are with the current task
→ Interrupt handler hands control to the dispatcher

→ dispatcher then starts the next priority process

It is worth noting that multiprocessing improves the computers efficiency.

HANDLING COMPETITION among processes is a key part of the operating system, how it allocates the resources.

→ FILE MANAGER, manages access and storage to files
→ MEMORY MANAGER, manages access to memory space
→ SCHEDULER, allocates space on the process table
→ DISPATCHER, allocates time slices

Semaphores if a resource eg a printer is available the OS needs to decide if it can give access to a process request to use it or identify if its being used by something and deny the request.

If the printer is in use it might deny the request and mark it as waiting for the printer to be available. It could do this by assigning a 1 for in use and a 0 for available.

However the interrupt of processes mean it's possible for the system to check the status and go to print then be interrupt and assign the printer to another task before restarting the first process.

This can be counteracted by not allowing the process to be interrupted when changing the flags by disabling the interrupt signal and then renabling it. Or by testing and setting the flag in one machine language instruction. The process of setting flags is known as semaphore.

Deadlock is where processes are blocked by each other from proceeding

STEP1 – locks the audio file then wants the audio card
STEP2 – locks the audio card then wants the audio file

Neither process can continue and are deadlocked. A deadlock can only occur if all of these conditions are satisfied.

1. Competition for resources that can't be shared
2. Resources are requested bit by bit
3. Once a resource is allocated it can't be given up

Most detection is at the third point and is known as DEADLOCK DETECTION and it can be resolved by forcibly returning the resources.

Those that look at the 2nd event are known as DEADLOCK AVOIDANCE, this can be done by making a process use a resource in one go or make unshareable resources shareable eg SPOOLING by storing the instruction to the resource in memory then executing it when its available

Network Classes

→ PAN (PERSONAL AREA NETWORK) eg wireless mouse and a PC
→ LAN (LOCAL AREA NETWORK) eg office computer network
→ MAN (METROPOLITAN AREA NETWORK) eg local community
→ WAN (WIDE AREA NETWORK) eg opposite sides of the world
  → OPEN NETWORK aka the INTERNET
  → CLOSED NETWORK / PROPRIETARY aka restricted access
    ↳ A bus network, star network, peer to peer

PROTOCOLS

Are standards that allow devices to communicate and understand each other, without standard protocols vendors would have to construct their own and then make sure the devices can communicate with other brands.

CSMA/CD (Carrier sense multiple access, collision detection) is for use on a bus network. All machines recieve all the messages but only keep those addressed to itself. If a machine trys to submit a message the same time as another they both stop transmitting and wait a random amount of time and then try again.

This doesn't happen on a wireless network as

1. Computers only have to be in range of the router not each other
2. As a result they won't know when each machine is transmitting

Wireless therefore uses CSMA/CA (Carrier sense, multiple access, collision avoidance) which works on avoidance rather than detection. When collisions do occur, messages can be resubmitted. The main method of avoidance is to give priority to messages that have been waiting longest. When the channel is silent it transmits, when it is busy it waits a random period before trying again.

REPEATER the simplest method of connecting 2 bus networks. It simply passes messages between the two networks.

BRIDGE similar to a repeater but only passes messages between networks if they are addressed to a computer on that network (more efficient)

SWITCH effectively same as a bridge but for more than 2 networks

ROUTER allows connections between different network types eg WIFI and ethernet. Builds a network of networks (an internet).

Distributed Systems is one where different devices perform different processes for a piece of software. There are several different types of such systems.

CLUSTER COMPUTING - many different computers work together to provide the computing power of that of a much larger machine.
GRID COMPUTING - more loosely defined than a cluster, and allows people to opt in their PCs computing power when not in use. For example assisting with large calculations on Research
CLOUD COMPUTING - where huge numbers of computers are connected and their computing power is utilised. Amazon rent out such machines, but does raise issues around security

INTERNET

The Internet (capital I) is a collection of networks maintained by ISPs. ISPs can be constructed into tiers

TIER 1 → very few high speed high capacity WAN
The backbone of the internet

TIER 2 → more regional less capacity

TIER 1 and TIER 2 ISPs networks of routers that provide infrastructure

TIER 3 → How the individual connects to the internet ISPS

THE LAST MILE PROBLEM — references the fact that the connection to the internet in most homes is typically via analogue phone lines. It is easier to update larger infrastructure, but that last little bit is proving costly.

IP ADDRESS is effectively the address of each computer on the internet. The ICANN (international corporation for assigned names and numbers) gives batches of IPs to ISPs, the ISPs then assign them to machines.

application

HTTP SMTP

Specific to applications eg HTTP for webpages, SMTP - Email

transport

TCP UDP

Handles transportation of the message

network

IP

Handles navigation between networks

link

ETHERNET WIFI

Handles transmission over individual networks

Common Protocols

SMTP → for sending email (simple mail transfer protocol). The sending server contacts the receiver using DNS to convert the email to the IP. The protocol dictates that it must identify itself to the sending server. SMTP was only intended for coding messages in ASCII so other protocols are used for data in different formats.

POP3 → (post office protocol 3) and IMAP (internet mail access protocol) are used for accessing mail that has already arrived on a computer. POP3 facilitates the download of the mail onto the users computer to be manipulated. IMAP covers the manipulation on the server

VOIP (voice over internet) There exists several forms of VOIP, mainly it uses a P2P network to connect the machines using existing speaker and microphone hardware aka SKYPE.

The second uses analogue phone adapters to connect to the existing telephone infrastructure.

ARTIFICIAL INTELLIGENCE

Artificial intelligence is the field of computer science seeking to build machines that do not require human intervention for complex tasks.

INTELLIGENT AGENTS - are devices that respond to stimuli from their environment eg. an autonomous airplane or a character in a video game

They will have sensors; microphones, cameras and actuators for affecting their environment; wheels, legs, grippers.

Most AI Research focuses on building AI that behaves intelligently by responding with its actuators to information from its sensors and can be categorised into several classes of intelligence.

SIMPLE - the simplest response is a reflex action, a predetermined response to inputted information

GOAL DIRECTED the result of forming a plan of action based on the current input eg walking to the end of a hall, playing chess. In some instances the machine can learn from trial and error and positive/negative reinforcement

TURING TEST used to serve as the benchmark for measuring progress in AI.

His proposal was the test should be if a human communicated with the device would they be able to determine if they were talking to a machine or not.

This is no longer considered relevant as it's fairly easy to mimic human response with basic rules

PERCEPTION - IMAGES identifying a grid of 8 digits would rely on the software matching bit patterns in the image to a prerecorded template and trying to compare the images. This is easily fallible especially when it comes to handwriting.

The second method is to match geometric features eg 1 would be a single vertical line, but this is still error prone depending on the quality of the images.

Trying to identify general pictures will be even more complex, it will involve edge processing (trying to view it as a line image), region matching (identifying bits of the image that are related), smoothing (to remove flaws)

LANGUAGE PROCESSING provides another tricky problem, while translating programming language to machine language happened fairly quickly. Spoken language is different as it relys on additional knowledge.

eg. CINDERELLA HAD A BALL
→ Cinderella had a ball like a football
→ Cinderella had fun
→ Cinderella threw a party.

SYNTACTIC ANALYSIS → identifies the grammatical role
SEMANTIC → identifies the actions
CONTEXTUAL → understanding what is meant by the sentence

REASONING rather than programming the machine to find all possible outcomes of a a problem it is more efficient to program the machine to construct it's own solutions. Each reasoning problems consist of PRODUCTION SYSTEM.

1. A COLLECTION OF STATES, the start state, the desired / goal state each state is a situation that might occur when solving the problem
2. A COLLECTION OF RULES/MOVES how we move from one state to another.
3. A CONTROL SYSTEM solves the problem of how to get from the start state to the goal state.

A state graph is a map of all the different states of the problem and has arrows determining which way you can move through the states. The control system determines the path to take to get to the goal state.

SEARCH TREES is the route the computer builds to get to the goal, it tests each possible path from current state and identifies how to get to the goal.

However sometimes constructing search trees can be time consuming, so the program will select the direction that looks the most promising. To do

thus we use a heuristic, which in some cases is a quantitive value measuring the distance between the current state and the goal. The heuristic shouldn't be onerous to calculate, it's meant to be less intensive than constructing a complete search tree

1.1 Abstraction

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-10-09 12:12 PM
Author:	SUKHJIT MANN		
Tags:	Abstraction, CPU, Notional Machine, RAM		

Cornell Notes	**Topic:** 1.1 Abstraction	Course: BSc Computer Science Class: How Computer Work [CM1030} Date: October 9, 2019

Essential Question:

- What is abstraction in terms of computer science?

Questions/Cues:

- What is a simple hierarchy of a computer process?
- What is the measurement for Computer Speed?
- How is Computer Memory measured?
- What is Abstraction?
- What is the most basic abstraction that we used to represent data on a computer?
- How is a computer picture represented?
- What is notional machine?
- What is a Central Processing Unit or CPU?
- What is RAM?
- What is the basics of how a Computer works?

Notes

- Modern webpage
    - Web Browser Application
        - Operating System
            - Computers
                - Central Processing Unit (CPU)/Main Microchip
                - Quantum Processes in Semiconductors

- Comp Speed = Gigahertz, 1 GHz = about a billion math operations per sec
- Memory measured in GB (Gigabytes)
- Abstraction = representation of key features, without unnecessary details
    - Ie. In Eng drawing, simple diagram to extract important features of mechanism
- Abstraction (Comp Sci Terms) = simplified representation of thinking; piece of software with other competing tech, emphasizes key details without distracting complexity.
- Basic Abstraction is to represent data on Comp as numbers

- Comp Pic = grid of tiny squares or pixels (short for picture elements); pixels have colour. Mixing together 3 color values in lights Red, Green, Blue. Color on screen can be rep'ed by 3 #'s RGB values.
- Higher Tier (simpler understanding) = more "layers" of abstraction
- Layers of Abstractions = something simpler to understand built-in to ongoing complexities underneath
- Abstraction of Comp Actions = Notional Machine
- Notional Machine = simplified version of how comp prog works
- CPU (Core of Comp) = chip to do all calculation on comp, run instructions which make up comp progs or code.
- Memory of Comp = Random Access Memory or RAM, where Comp stores all data it's working on.
- Data from Memory $\longrightarrow$ CPU $\longrightarrow$ Write back to Memory

Summary

In this week, we learned the basics of a computer process, how Computer Speed and Memory is measured. Furthermore, we've learned how abstraction is vital to our understanding of computers, what abstraction is ie. a picture on a computer, and layers of abstraction are built-in to understand ongoing complexities underneath as a way of representation. On the other hand, we also learned how abstraction can not only be used as representation but as a way to showcase the actions of a computer or a notional machine. In closing, we also touched on how the basics of how computers works with components such as CPU and RAM working together in sync.

2.1 State and 2.2 Modularity and Applications

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-10-13 3:45 PM
Author:	SUKHJIT MANN		
Tags:	Bug, library, Locking, Metadata, Modularity, State		

Cornell Notes	**Topic:** 2.1 State 2.2 Modularity and applications	Course: BSc Computer Science Class: How Computer Work [CM1030} Date: October 13, 2019

Essential Question:

- What are states, how they present in various aspects of the computing experience and to that how software modular affects our understanding and debugging of such a system?

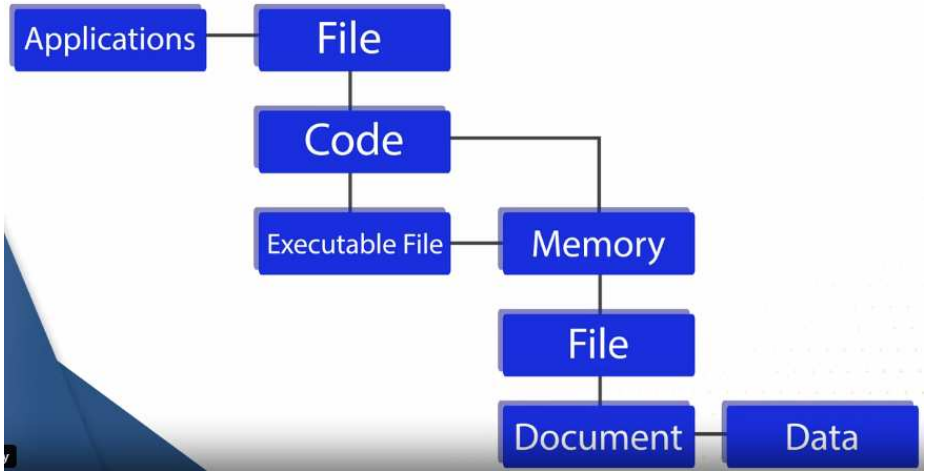
Questions/Cues:

- What is a state?
- What are two states of a computer?
- How does a computer change between states?
- What are basic states of a video player?
- What are the drawbacks of having internet connected applications?
- What is a website where you go to buy something?
- What are states of a e-commerce site?
- What does a state depend on?
- What does numbers and states have in common?
- What is a file?
- What are files on a hard disk?
- What can files contain?
- What is metadata?
- What is a permission system on a Operating System?
- What is file locking?
- What is Modularity?
- What are drivers?
- What are some tasks that the Operating System handles?
- What are libraries?
- What is a bug and what is debugging?
- What are some simple steps to debugging?

Notes

- State = temp config of a Comp system, affects how it responds to interaction
- Basic Comp States = on or off

- Any given time, comp in one of several states, changes between b/c of user input or possibly through own internal processes
- Video player = play state, pause state, stop state. For online video, sometimes a buffer state.
- Internet connected apps heavily relies on the strength of internet connection, if it fails, apps interrupted, freeze, crash or produces unexpected behavior ie. email doesn't know what to do with a half email so crashes/ refuses to fetch emails properly and update client.
- Buying website = e-commerce (electronic commerce) website
- Home page ⟶ Search page (# of results, possible products) ⟶ looking at particular product⟶ Add to/Added to Shop cart ⟶ Possible Login page ⟶ Logged in and item added to shop cart ⟶ Checkout state ⟶ Delivery address⟶Card details⟶Delivery method⟶Click purchase now⟶Processing Card transaction⟶Payment Sucessful⟶ Item bought
- State depends on how Comp Memory is set up
- Comp Memory stores billions of #'s, changing any #'s = changing state of Comp
- # of comp states comp can have is astronomical, more than # of atoms in the universe
- Progs written to avoid nonsense states
- Modern apps connect to internet to get data or checksum info, internet breaks or not logged in, stuck in frozen loop
- Start of prog = default state, cleared of last state, best fix for strange state; **turn it off and turn it on again**
- File = abstraction of some data, doc, pic or song
- Files on a hard disk = loads of patterns on disk surfaces
- OS presents the patterns of raw data as set of files in a folders
- Files = Data, words, pixels, or audio samples and associated metadata
- Metadata = data of file, when created or who owns file
- Metadata not part of file, stored by OS
- Metadata used by OS when dealing with file itself, recent files, OS looks at Mod dates
- Permission System = prevents opening of other ppl files
- File lock = locking files when apps uses them so other apps cannot open them, app opens file sets the locked metadata and when app closes file it unlocks it
- File in Close state on HDD⟶App opens file and its locked⟶App copies file to memory to edit(while keeping HDD copy open and locked until copying is complete)⟶ File on HDD closed and unlocked and app now shifted to work on file data in memory⟶ Closed, unlocked, unchanged file on HDD and copy on memory that is changed⟶ App opens HDD file again and locks it to save changes to file⟶Changes copied back to HDD and HDD file is closed and unlocked again⟶App closes file and back to closed state of original file, contents changed
- If app crashes between editing and saving to file, data lost, but modern systems use backups to prevent this
- Modularity = splitting into smaller chunks called modules, about creating systems of simple modules that interact with each other
- Drivers = software that controls hardware, independent of any application
- Code is stored as bit in memory, files before app starts just like doc, instructions to CPU(code) just another type of data



- OS tasks = drivers, writing to files, interacting with networks, and drawing to screen.
- Libraries = bits of code, work with app to make app can use surface, sometimes part of exe file as app. Can be separate file called dynamic library that interacts with main exe file
- User interface library = code to display buttons, menus, and other interface items
- Resource file = data applies to app as whole, not to individual docs, ie. to make UI look right or store setting preferences
- Bug = error in Comp prog
- Debugging = getting rid of bugs, or errors
- Understand the System→ Look, don't Think (carefully looking at what Comp is doing and gathering info)→ Make it fail (Know when problem happens, test system and reproduce bug reliably)→ Only change one thing at time

Summary

In this week, we looked at computer states and in turn states of various applications we use on said computer. Alongside this, we discussed the importance and impact of modularity of a application system and its various components which work together to provide a user their experience. Lastly, we discussed the effect of these components when bugged and producing erratic results and method of debugging we would use to bring the system to a functional working state.

3.1 Networks and 3.2 Security

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-10-19 2:11 PM
Author:	SUKHJIT MANN		
Tags:	2FA, Encryption, Internet, ISP, Network, Security, Virus		

Cornell Notes	**Topic:** 3.1 Networks, 3.2 Security	Course: BSc Computer Science Class: How Computer Work [CM1030} Date: October 19, 2019

Essential Question:

- How do computers communicate with each other and what security measures can be put in place to ensure the safety of computers, their contents and their messages?

Questions/Cues:

- What is a network?
- What is bandwidth?
- What are protocols?
- What is the internet?
- What is a Router?
- What is a ISP and IP Address?
- What is a Packet?
- What is an URL?
- What is Domain Name and the Domain Name Service?
- What is Virus?
- What is Spyware?
- What is a Trojan Horse?
- What is Phishing?
- What is Encryption
- What is an Encryption Key or Key?
- What is Public Key Encryption?
- What is a Certificate?

Notes

- Network = # of comps connected by links, like a fishing net
    - Traditionally comps connected using wires
    - Cooper cables replaced by fiber-optic or wireless
- Fibre-optic = signals sent via light
- Wireless = comms via electromagnetic waves which require no physically connected comp
- Bandwidth = amount of data that can be sent over a network within given time
    - "How fast" network is

- Reliability = msg sent over network will get to destination
    - Wired networks super reliable b/c wire from one comp to another
    - Wireless unreliable b/c of bad signal strength (walls or obstacles), or too many people on network
- Bandwidth and Reliability effect Network Performance
- Protocols = lang used for network comms
    - Not like human langs, very strict rules
    - Protocol defines rules on how to communicate
- Wifi and Cell networks same basic tech, but cell network longer range and generally lower bandwidth than Wifi, diff protocols for comms
- Internet = network of networks, connects together diff networks, allows comps to communicate with other comps across diff networks
- Router = connects together 2 networks, forwarding msgs from 1 network to other, also translating between network protocols
- When we use internet, we connect local network, local nets don't directly comm with each other
- ISP (Internet Service Provider) = big networks, span whole countries, evolution of old wired telephone nets
    - Nationalized ISPs are connected to International ISPs, called the Internet Backbone
- Packet = like virtual envelope with internet address and other info
- IP Msg wrapped in wifi protocol and sent to home router $\rightarrow$ ISP protocol envelope
- Emails use Send-mail protocol, web pages used HTTP Protocol
- Every Comp on Internet has IP Address, know IP address allows routers to reach comp
- URL(Uniform Resource Locator) = human-friendly version of IP Address
    - middle part of URL = domain name, name of server where site is stored
    - Domain name is converted to IP Address using Internet Service called Domain Name Service or DNS
- Virus = software that copies itself from one comp to another without user knowledge, aimed at damaging comp
    - slows down comp, interferes with OS, or stealing data from comp
- Spyware = records what you do on comp and sends to third party
    - recording what you type on keyboard, getting password
- Trojan Horse = appears to be legit software, but performs harmful actions in secret
- Hacking = getting access to system that you had no previous access to; to steal data
    - Hackers normally go after web servers or internet servers that have lots of data, rather than indiv comp
- Phishing = fooled into giving password or bank details, through a plausible-looking site or email that looks like from bank or similar; controlled by criminals
- Encryption = sending msgs in codes, so can only be read by person who is supposed to receive them
    - Code in special language cannot be read unless you know code
- Encryption Algorithm = series of math operations converts text into another form
    - designed so msg cannot be deciphered, even if all details of encrypt algo are known as long as key unknown
- Encryption Key or Key = a # that controls how encrypt algo works
- Public Key Encryption = diff keys used to encrypt msg and to decrypt it
    - Encrypt key is public, so anyone can used to create encrypt msg; only people with decrypt key can read msg
- Certificate = to demonstrate validity of public key; keys and certs issued by # of trusted security companies
    - Web browser can check trustworthiness of certs
- Antivirus Software = includes info on all known virus and searches comp for them
- Firewall = stops suspect-looking msgs from coming in or out of comp
    - prevents direct hacking of comp
- Access Control = certain people can have access to comp or server
    - Done by making people log-in with username and password

    - Diff people have access to diff parts on comp or server to do what they need, but can't damage it accidentally or deliberately
    - Rely on passwords (always encrypted, some easy to guess)
- 2FA (Two factor authentication) = includes password, but require to enter passcode that site sends to mobile phone

Summary

In this week, we learned how computer communicate with each other, what the internet is and the different protocols required to do communications over networks. Alongside this, we learned about potential security threats to our information when using the internet and ways to effectively ensure the safe use of our time on the internet when communicating and storing sensitive information.

4.1 Web Applications

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-10-25 6:56 PM
Author:	SUKHJIT MANN		
Tags:	API, Client, Database, HTML, HTTP, Internet, Server, Template		

Cornell Notes	**Topic:** 4.1 Web Applications	Course: BSc Computer Science Class: How Computer Work [CM1030} Date: October 25, 2019

Essential Question:

What is a website, web pages and what is the inner workings of web content consumed on the Web?

Questions/Cues:

- What is the difference between the Internet and the World Wide Web?
- What is a webpage and website?
- Where are web pages located?
- What two protocols are used by web applications?
- What is HTTP and HTTPS?
- What is the state of a web server?
- What are Cookies?
- What is a Database?
- What is a Template?
- How do Templates and databases work together to produce new pages?
- What is an API?

Notes

- Internet = underlying network that connects comps together across world
    - Network that connects computers together
- World Wide Web = way of communicating and sharing info that uses Internet, anything viewed through web browser part of WWW
    - the Web consists of pages connected by links
- Webpage = single doc on web which loaded into browser at given time
- Website = collection of page
    - interactive, make use of data from many sources, and constantly updated based on how people interacting with site
- The web not so much as people communicating with each other, but more about people getting info from organizations
- Web pages located on organization web server, no matter where they are accessed from
- Client = essentially a comp that person is working on

    - term "Clients" used both for comp and software used to access server, the web browser
- Server = comp that holds info including web pages, used by many people via different clients
- Web apps use IP (Internet Protocol) and HTTP (Hypertext Transfer Protocol) to comm over internet
- Tech def of WWW = anything that uses HTTP
- Browser send HTTP Request (packet asking for web page)
- HTTP Request consist of URL, first part is HTTP (showing it uses HTTP protocol) ie. **http://**, second is domain name ie. **london.ac.uk** (name of website), third part tells server exactly what page of website you want ie. **/search?,** and finally optional fourth part called query string because web pages are created especially for you based on requirements, query allows to specify those requirements ie. **search=computing**
- HTTPS = Hypertext Transfer Protocol Secure
    - when using HTTPS, all data transmitted is encrypted
    - https shows in url instead of http when https is used (padlock sign)
- Lots of websites used access control, ask user to log in before accessing parts of website using private or sensitive info
- Viewing state of website = is page you are viewing
- Website also have logged in and not logged in state each with different level of access
- Web servers, where websites stored mostly have no state
    - Respond to every request same way, no change based on differing interaction
- Cookies = small amounts of data websites store in our comps as we use sites
    - stored by browser software on comp not server
    - designed to keep track of state of websites
    - track whether logged in or not and how they know our browsing history
- Server can have different elements, basic web server consists of # of web pages, each being 1 or more files called HTML files
- Web pages created only when they needed on response to particular requests ie. profile on social media site
- Database = software platform designed to store data in structured way, easy access for comp; efficient to get data from DB but also to add new data to it
    - make possible to extend info available on website without anyone creating new pages
- pages we see created by server using data from DB, website stores template page which contains formatting for web page. Template has lots of empty spaces to be filled with data from DB. Web server software combines templates with data from DB to create pages then transmitted to browser
- Template maybe stored as file, web page formatting done using HTML; basic lang of web
- API = Application Programming Interface
    - gain data from other servers, version of website for comps progs to look at
    - how we have different versions of the same app ie. mobile or desktop

Summary

In this week, we learned what is a website and what webpages are. In detail, we learned about the various components that go into making a website appear on your browser in working condition and allow you to interact with it. Alongside this, we learned modern website are much more complex than before because of the interplay between web servers, databases, templates and APIs.

5.1 Data Storage and 5.2 Data Representation

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-10-30 3:01 PM
Author:	SUKHJIT MANN		
Tags:	AND, ASCII, Bit, Byte, Cache, NOT, OR, RGB, UNICODE		

Cornell Notes	**Topic:** 5.1 Data Storage 5.2 Data Representation	Course: BSc Computer Science Class: How Computer Work [CM1030]-Lecture Date: October 30, 2019

Essential Question:

What are different methods of data storage? Alongside this, what is a bit and/or bytes and how can we use these to represent our data?

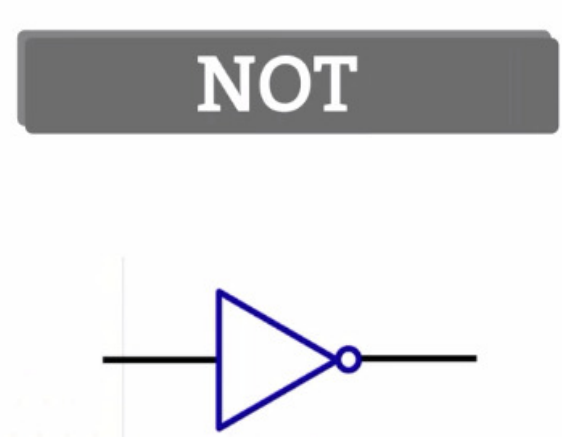
Questions/Cues:

- What is a bit?
- What is the main difference between Main Memory and Mass Storage?
- What is the NOT operation on a bit?
- What is the AND operation on a bit?
- What is the OR operation on a bit?
- What are the different types of storage media?
- What is cache and virtual memory?
- How many bit patterns can a byte represent?
- What is a binary number and how is it represented?
- What is ASCII and Unicode?
- How can we represent colour to humans on a computer using bits and bytes?

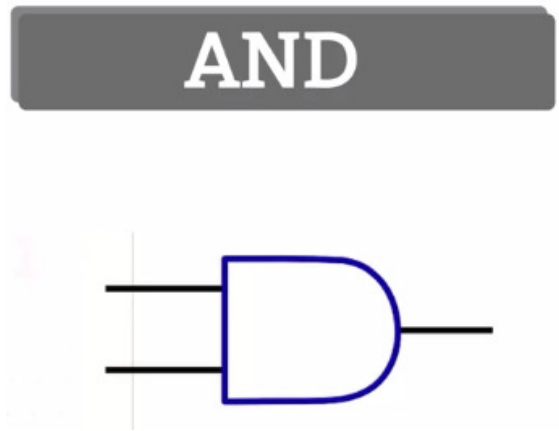
Notes

- Bit = fundamental unit of digital info, can have one of 2 states at any 1 time; (1 or 0, true or false, on or off)
    - Is an abstract representation that represent any duality, binary distinction between 2 things
    - Comp memory measured in Gigabytes (GB), byte is 8 bits, GB is a billion bytes
    - Modern HDD measured in Terabytes(TB) or 8 trillion bytes
- Main Memory(Temporary Storage) = where info is stored that Comp currently using, made up of RAM (Random Access Memory)
    - Everything in main memory forgotten when comp turned off
    - Fast and easy to write to but capacity small < 100GB
- Mass Storage (Permanent Storage) = where things are stored permanently, where doc when you save it.
    - Slower, but saves data when comp turned off and capacity into TB
- Microchips handle bits, microchips perform calculations on bits

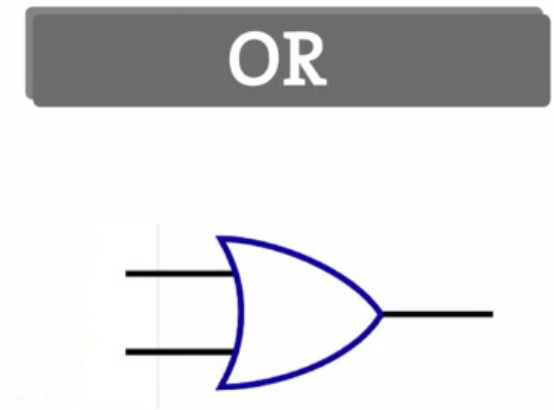
- NOT operation = taking single bit and making it its opposite. ie. One become Zero



- AND operation = when dealing with two bits, has output of 1 if both inputs are 1, but 0 if either input is 0



- OR operation = when dealing with two bits, if either input is 1, output is 1. Output only 0 if both inputs are 0



- Hard disk stores data magnetically, DVD is optical storage and stores data that can be read with light, Flash USB stores data permanently alternative to RAM, also this Flash storage comes as alternative to HDD as flash-based storage drives called SSD (Solid State Drives).
    - SSD slower than RAM, but much faster than HDD or optical drives
    - Optical drives slow to write to, best used for permanent archiving
    - With Storage we trade off between storage capacity, power usage and speed
- Cloud storage = data not stored on own comp, but comp somewhere on Internet

    - since data transmitted over Internet, storing and accessing data much slower than device on comp.
- Cache = often on same chip as main processor, used for storing data being used at very moment
- Virtual Memory = when main memory full, comp saves some of what should be in memory to HDD, area called virtual memory
    - Comp does its best to make sure what is placed in Virtual Memory isn't something to be used immediately
- On modern comp, smallest element to access is 64 bits or 8 bytes
- A byte can represent 256 different patterns of 1s and 0s called bit patterns. With single byte rep'ping #'s up to 255
- Binary number rep'ed by base 2 instead of 10, decimal number can be converted to binary rep like so

1	1	0	1
1x	1x	0x	1x
8	4	2	1
=	=	=	=
8	4	0	1

- American Standard Code for Information Interchange (ASCII) = standard for rep'ping texts, includes all Eng alphabet, numbers and special characters in seven bits
    - Gradually being replaced by UNICODE to include almost any writing system in the world
- Shades of B & W between 0 and 255
- We use RGB and optionally A to represent colour to humans with R being red, G being green, B being blue and A being alpha or opacity (transparency)

Summary

In this week, we learned what is a bit is and bytes are. Also we looked at the different operations performed on bits. Alongside this, we looked at the binary representation of decimal numbers, letters and etc. Further, we touched on the ASCII and UNICODE standards in computing. Finally, we explored the various types of storage media

5.1 Data Storage and 5.2 Data Representation-Reading

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-11-10 10:03 AM
Author:	SUKHJIT MANN		
Tags:	AND, ASCII, Bit, Byte, NOT, OR, RGB, UNICODE, XOR		

Cornell Notes	**Topic:** 5.1 Data Storage 5.2 Data Representation-Reading	Course: BSc Computer Science Class: How Computer Work [CM1030]-Reading Date: October 30, 2019

Essential Question:

What are different methods of data storage? Alongside this, what is a bit and/or bytes and how can we use these to represent our data?

Questions/Cues:

- What are bits and bit patterns
- What are the Boolean operations on bits?
- What is a Gate?
- What is a flipflop circuit?
- What is Very Large-scale Integration?
- What is Hexadecimal Notation?
- What is Main Memory?
- What is an Address?
- What are the Read and Write Operations in Memory?
- What is DRAM and SDRAM?
- What is an Hard disk or Magnetic Disk?
- What are the measurements to evaluate a disk's performance?
- What is an CD?
- What is flash memory?
- What is a Solid State Disk?
- What is ASCII?
- What is Unicode?
- What is a text file?
- What is binary notation?
- What is a pixel and a bit map?
- What is RGB Encoding?

Notes

- Bits = binary digits, encode info inside comp as a patterns of 0's and 1's called bit patterns
    - Sometimes they represent numeric values, characters of alphabet, images or sounds

- Boolean operations = manipulate true/false values
  - bit 0 reps value false and bit 1 reps value true
  - AND reflects truth or falseness of compound statement formed by combining 2 smaller or simpler statement with conjunction and. Both components of the statement must be true for statement to be true ie. 1 and 1 = 1, conversely 1 and 0 = 0
  - OR is also again a compound statement of two smaller or simpler statement
    - OR is true when at least one component is true ie. 1 or 0 = 1
  - XOR is also a compound statement, produces output of 1(true) when one of its inputs is 1 and other is 0 (false)
    - "either P or Q but not both"
    - XOR operation produce output of 1 when inputs are different
  - NOT operation has only one input, output the opposite of the input

The AND operation

0	0	1	1
AND 0	AND 1	AND 0	AND 1
0	0	0	1

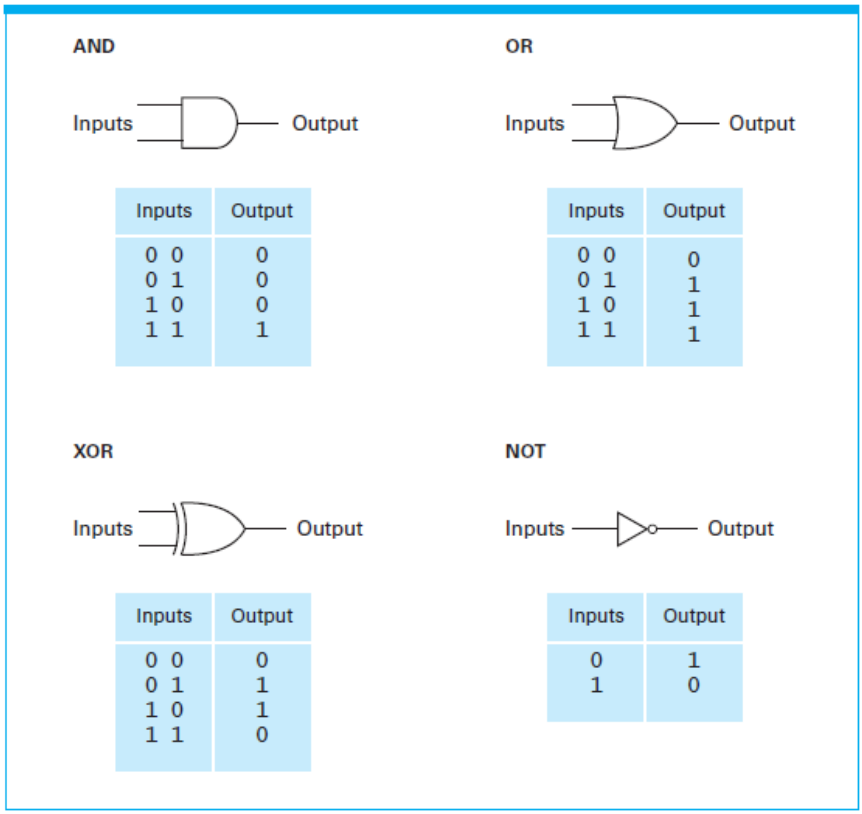
The OR operation

0	0	1	1
OR 0	OR 1	OR 0	OR 1
0	1	1	1

The XOR operation

0	0	1	1
XOR 0	XOR 1	XOR 0	XOR 1
0	1	1	0

- Gate = device that produces output of Boolean operation when given operation's input values
  - constructed from variety of tech such as gears, relays, and optic devices
  - in today's comps usually implemented as small electronic circuits, where 1 and 0 rep'ed as voltage levels



Inputs	Output
0 0	0
0 1	0
1 0	0
1 1	1

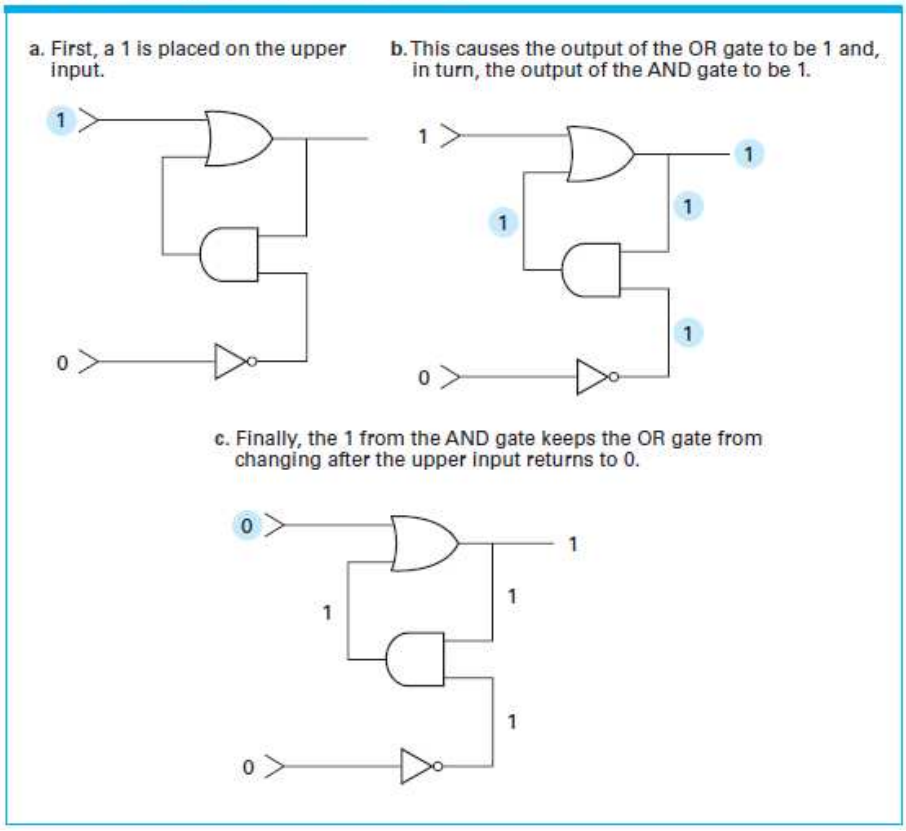
Inputs	Output
0 0	0
0 1	1
1 0	1
1 1	1

Inputs	Output
0 0	0
0 1	1
1 0	1
1 1	0

Inputs	Output
0	1
1	0

- Flip flop = fundamental unit of comp memory, circuit that produces output of 0 or 1 which remains constant until pulse (temp change to a 1 that returns to 0) from another circuit causes it to shift to the other value
    - the output can be set to "remember" a 0 or 1 under control of external stimuli

Figure 1.4 Setting the output of a flip-flop to 1



- Very large-scale integration = allows millions of electronic components to be constructed on wafer (chip), used to create mini devices containing millions of flip-

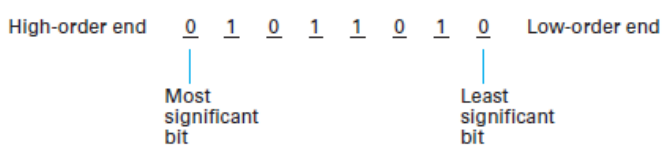
flops along with controlling circuitry
    - VLSI sometimes used to create entire comp system on single chip
- Stream = long string of bits
- Hexadecimal Notation = to simplify bit patterns, because patterns within machine have lengths in multiples of 4. Hexa Note uses single symbol to rep pattern of 4 bits

Figure 1.6 The hexadecimal encoding system

Bit pattern	Hexadecimal representation
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

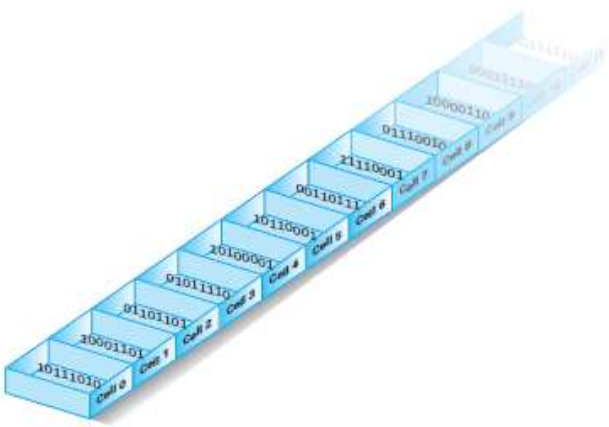
- Main memory = large collection of circuits (such as flip-flops) each capable of storing single bit comes together to form bit reservoir for purpose of storing data
    - organized in manageable units called cells, size of 8 bits or 1 byte
    - most significant or high-order bit determines numercial value, if this were a number

Figure 1.7 The organization of a byte-size memory cell



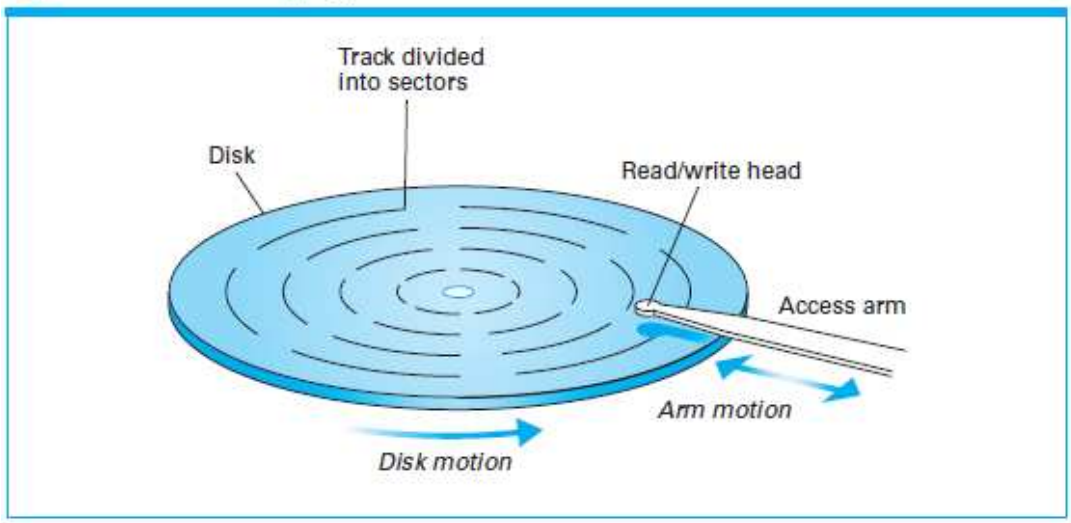
- Address = unique name to identify individual cells in comp's main memory
    - addresses entirely numeric

Figure 1.8 Memory cells arranged by address



- Read operation = circuitry that hold bits combined with other circuitry to allow other circuits to store and retrieve data, other circuits get data from memory by electronically asking for contents of certain address
- Write operation = To record info in memory by requesting that a certain bit pattern be placed in cell at particular address
- Because comp's main memory organized as individual, addressable cells accessed in any order, other name is RAM (Random Access Memory)
- DRAM(Dynamic RAM) = constructed from tech like refresh circuit which replenishes charges many times a sec because bits stored as tiny electric charges which dissipate quickly
- SDRAM(Synchronous DRAM) = applies additional techniques to decrease time need to retrieve contents from its memory cells
- Kilobyte(KB) = 1024 bytes
- International Standards Organized put out specialized terms for use for powers of two, kibi-, mebi-, gibi-, tebi-, but abandon by international community for kilo-, mega-, etc. which refer to powers of two in terms of comp measurements but refer to powers of 1000 when used in other contexts
- Advantage of mass storage over main memory include less volatility, large store capacity, low cost and the ability to remove storage medium from machine for archival purposes
- Magnetic & optical require mech motion so ergo, more time to store & retrieve data. Also more prone to mech failures than solid state systems
- HDD or magnetic disk = thin spinning disk with magnetic coating used to hold data, Read/write heads place above and/or below disk so as disk spins, each head transverses a circle called track
    - by re-positioning read/write heads different tracks accessed
    - disk system consists of several disk mounted on common spindle, 1 on top of other, read/write heads slip between platters and work in unions. Each time heads re-positioned, set of new tracks or cylinder becomes accessible
    - each track divided in small arks called sectors, info is recorded as continuous string of bits, all sectors contain same # of bits (512 bytes to few KB) and in simplest disk store system each track have the same # of sectors

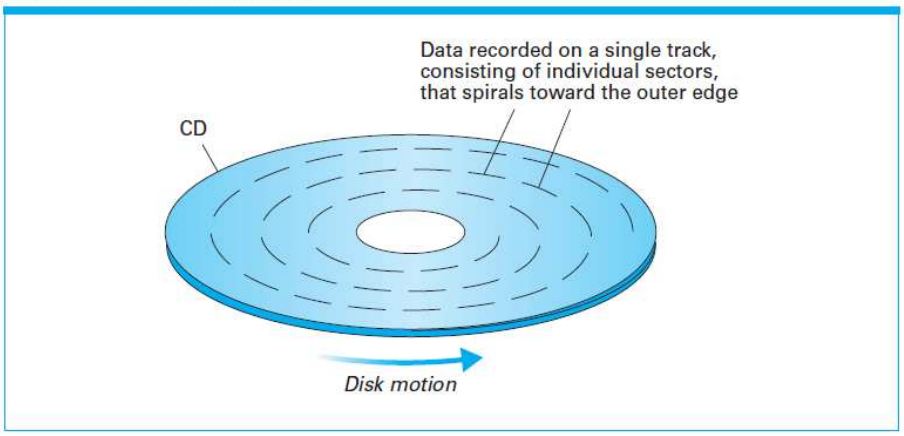
Figure 1.9 A disk storage system



    - zone-bit recording = several adjacent tracks collectively known as zones, with typical disk containing 10 zones. All tracks within zone have same # of sectors, but each zone has more sectors per track than the zone inside of it
    - capacity depends on # of platters used and density in which tracks and sectors are placed
- Measurements to evaluate disk system's performance:

    - Seek time = time required to move read/write heads from 1 track to another
    - Rotation delay or Latency time = half the time required for disk to make complete rotation, which is average amount of time required for desired data to rotate around to read/write head once head positioned over desired track
    - Access time = sum of seek time & rotation delay
    - Transfer rate = rate at which data can be transferred to or from disk
        - Note** in case of zone-bit recording, amt of data passing read/write head in single disk rotation > for tracks in outer zone than inner zone; data transfer rate varies depending on portion of disk being used
        - measured in MB per sec
- CD (Compact Disk) = 12cm in diameter, info recorded on them by creating variation in their reflective surfaces, info retrieved by laser that detects irregularities on reflective surface as CD spins
    - originally to audio reccordings using CD-DA (compact disk-digital audio)
    - CD today for comp data store use same format, info is stored on single track that spirals around CD, track spirals from inside out, track divided into sectors each with identifying marks and capacity of 2KB of data

Figure 1.10 CD storage format



    - Traditional CDs have capacities of 600-700MB, DVD(Digital Versatile Disks) have several GB, and BDs(Blu-ray Disks) have 5x over the DVD
- Flash memory = bits stored by sending electronic signals directly to storage medium where they cause electrons to be trapped in tiny chambers of silicon dioxide, thus altering characteristics of small electronic circuits. Since chambers are able to hold captive electrons for many yrs without external power; tech perfect for portable, nonvolatile data storage
- SSD(Solid-state disks) = larger flash memory devices, resilient to vibrations & physical shock, quiet operation (no moving parts) and lower access times
    - suffer from limited lifetime out of all flash tech
        - Wear-level techniques to alleviate this by relocating frequently altered data blocks to fresh locations on drive
    - Other flash tech include SD(Secure Digital) memory card, SDHC(High Capacity) memory cards and next-gen SDXC(Extended Capacity) memory cards
- ASCII (American Standard Code for Information Interchange) = uses bit patterns of length seven to rep upper & lowercase letters of the Eng alphabet, punctuation symbols, digits 0-9, and certain info like line feeds, carriage returns and tabs
    - ASCII extended to 8-bit-per-symbol format by adding a 0 at most significant end of each 7-bit pattern
        - this technique produces code in which each pattern into typical byte-sized memory cell. Also provides 128 additional bit patterns (obtained by assigning the extra bit value 1) that can be used to rep symbols beyond Eng Alphabet and associated punctuation

Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex
line feed	00001010	0A	>	00111110	3E	^	01011110	5E
carriage return	00001011	0B	?	00111111	3F	_	01011111	5F
space	00100000	20	@	01000000	40	`	01100000	60
!	00100001	21	A	01000001	41	a	01100001	61
"	00100010	22	B	01000010	42	b	01100010	62
#	00100011	23	C	01000011	43	c	01100011	63
$	00100100	24	D	01000100	44	d	01100100	64
%	00100101	25	E	01000101	45	e	01100101	65
&	00100110	26	F	01000110	46	f	01100110	66
'	00100111	27	G	01000111	47	g	01100111	67
(	00101000	28	H	01001000	48	h	01101000	68
)	00101001	29	I	01001001	49	i	01101001	69
*	00101010	2A	J	01001010	4A	j	01101010	6A
+	00101011	2B	K	01001011	4B	k	01101011	6B
,	00101100	2C	L	01001100	4C	l	01101100	6C
.	00101101	2D	M	01001101	4D	m	01101101	6D
/	00101110	2E	N	01001110	4E	n	01101110	6E
0	00101111	2F	O	01001111	4F	o	01101111	6F
1	00110000	30	P	01010000	50	p	01110000	70
2	00110001	31	Q	01010001	51	q	01110001	71
3	00110010	32	R	01010010	52	r	01110010	72
4	00110011	33	S	01010011	53	s	01110011	73
5	00110100	34	T	01010100	54	t	01110100	74
6	00110101	35	U	01010101	55	u	01110101	75
7	00110110	36	V	01010110	56	v	01110110	76
8	00110111	37	W	01010111	57	w	01110111	77
9	00111000	38	X	01011000	58	x	01111000	78
:	00111001	39	Y	01011001	59	y	01111001	79
;	00111010	3A	Z	01011010	5A	z	01111010	7A
<	00111011	3B	[	01011011	5B	{	01111011	7B
=	00111100	3C	\	01011100	5C	|	01111100	7C
	00111101	3D	]	01011101	5D	}	01111101	7D

- UNICODE = uses a unique pattern of up to 21 bits to rep each symbol
  - combined with Unicode Transformation Format 8-bit (UTF-8), traditional ASCII chars still rep'ed with 8 bits while thousands of additional chars form langs like Chinese, Japanese and Hebrew can be rep'ed by 16 bits
    - UTF-8 uses 24- or 32-bit patterns to rep more obscure Unicode symbols
- Text file = consisting of long sequence of symbols encoded using ASCII or Unicode
  - text file and file produced by word processor both consist of textual material, but txt file contains only char-by-char encoding of text, whereas file produced by word processor contains numerous proprietary codes rep'ing changes in fonts, align info, and other parameters
- Binary notation = way of rep'ing numeric values using only digits 0 & 1

Figure 1.13 The base 10 and binary systems

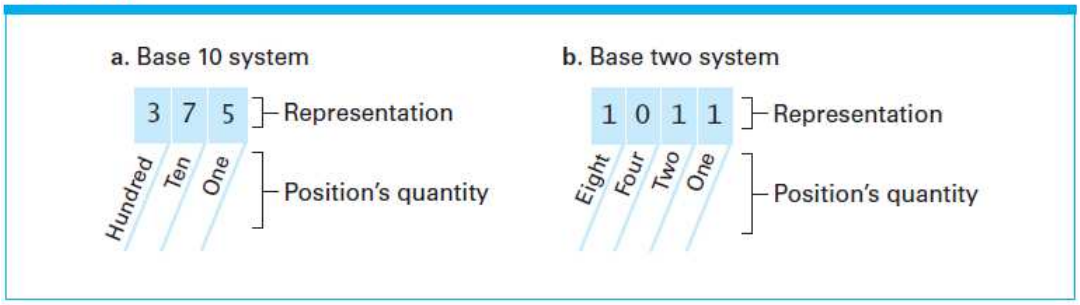


Figure 1.14 Decoding the binary representation 100101

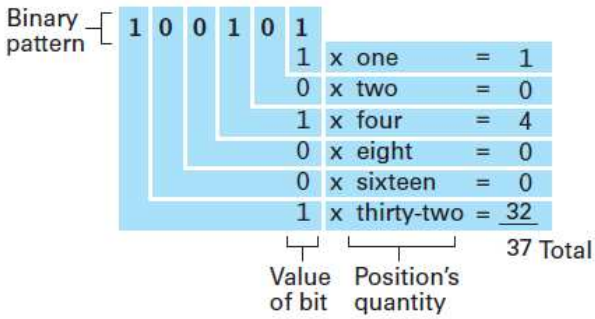


Figure 1.15 An algorithm for finding the binary representation of a positive integer

Step 1. Divide the value by two and record the remainder.

Step 2. As long as the quotient obtained is not zero, continue to divide the newest quotient by two and record the remainder.

Step 3. Now that a quotient of zero has been obtained, the binary representation of the original value consists of the remainders listed from right to left in the order they were recorded.

Figure 1.16 Applying the algorithm in Figure 1.15 to obtain the binary representation of thirteen

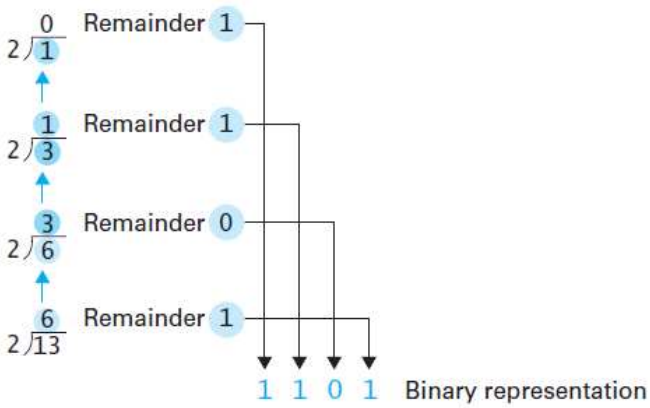
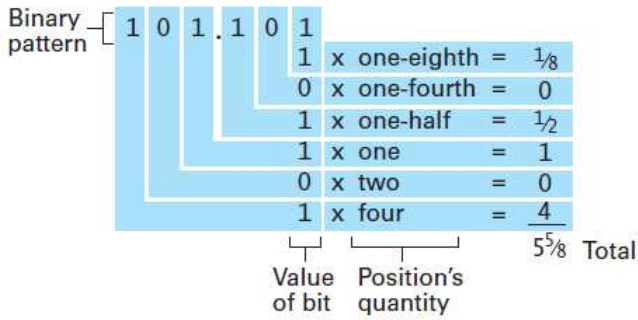


Figure 1.17 The binary addition facts

$$\begin{array}{r} 0 \\ +0 \\ \hline 0 \end{array} \quad \begin{array}{r} 1 \\ +0 \\ \hline 1 \end{array} \quad \begin{array}{r} 0 \\ +1 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ +1 \\ \hline 10 \end{array}$$

Figure 1.18 Decoding the binary representation 101.101



- Pixel (picture element) = dot
- Bit map = appearance of each pixel is encoded and an entire image is rep as collection of the encoded pixels
    - image cannot be easily to any arbitrary size
- For eleborate B & W pics, each pixel can be rep'ed by collection of bits (usually 8) which allows variety of shades of grayness to be rep'ed
- RGB encoding = each pixel is rep'ed as 3 color components- red, green, and blue, 1 byte is normally used to rep intensity of each color component. So 3 bytes of storage required to rep single pixel in original image

Summary

In this week, we learned what is a bit is and bytes are. Also we looked at the different operations performed on bits. Alongside this, we looked at the binary representation of decimal numbers, letters and etc. Further, we touched on the ASCII and UNICODE standards in computing. Finally, we explored the various types of storage media

6.1 Compression

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-11-04 12:22 PM
Author:	SUKHJIT MANN		
Tags:	Compression, Lossless, Lossy, Run		

Cornell Notes	**Topic:** 6.1 Compression	Course: BSc Computer Science Class: How Computer Work [CM1030]-Lecture Date: November 04, 2019

Essential Question:

What are the various algorithms to compress data and how are they achieved?

Questions/Cues:

- What is a compression algorithm?
- What is Redundant Data Representation?
- What is a run?
- What is loseless and lossy compression?

Notes

- Compression algo = make it possible to reduce size of files so they can be stored more efficiently
- Redundant Data rep = meaning data reps use more bits to rep something when necessary
- Run = sequence of pixels of same colour
- Lossless Compression = compress data without losing info
- Lossy Compression = compress data by losing some info
    - With images and audio, possible to lose info almost unnoticeable to most people
    - Can achieve much better compression even when data loss is not much
    - By losing info, it makes image rep more redundant and easier to compress

Summary

In this week, we learned about the general data compression algorithms lossless and lossy and how most algorithms in regards to images and video perform redundant data representation to represent something with more bits when necessary.

6.1 Compression-Reading

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-11-04 5:04 PM
Author:	SUKHJIT MANN		
Tags:	Compression, Dictionary, Frequency, GIF, Image, JPEG, Lossless, Lossy, MP3, Pixel, RGB, ...		

Cornell Notes	**Topic:** 6.1 Compression	Course: BSc Computer Science Class: How Computer Work [CM1030]-Reading Date: November 04, 2019

Essential Question:

What is data compression and the various techniques/file formats used to compress Image and Audio files?

Questions/Cues:

- What is data compression?
- What is lossless and lossy data compression?
- What is Run-length encoding?
- What is Frequency-dependent encoding?
- What is Relative encoding?
- What is Dictionary encoding?
- What is Adaptive Dictionary encoding?
- What is a GIF?
- What is a JPEG?
- What is TIFF?
- What is MP3?

Notes

- Data Compression = to reduce size of data involved while retaining underlying info
- Lossless = don't lose info in compression process
- Lossy = may lead to loss of info in compression process
    - provide more compression than lossless, popular in case of images and audio where minor errors tolerated
- Run-length encoding = process of replacing sequences of identical data elem. with code indicating element that is repeated & # of times it occurs in the sequences
    - where data being compressed consists of long sequences of same value
- Frequency-dependent encoding = a system in which length of bit pattern used to rep data item is inversely related to frequency of item's use
- Relative encoding (differential encoding) = record differences between consecutive data units rather than entire units; each unit encoded in terms of relationship to previous unit

    - can be implemented in either lossless or lossy depending on whether differences between consecutive data units encoded precisely or approximated.
- Dictionary encoding = term dictionary refers to collection of building blocks from which message being compression is constructed & msg itself encoded as sequence of references to dictionary.
    - Normally thought of as lossless system, but when img compressing, there's times when entires in dictionary are only approximations of correct data elem. resulting in lossy compression
- Adaptive dictionary encoding (dynamic dictionary encoding) = dictionary allowed to change during encoding process
    - Example of this is Lempel-Ziv-Welsh (LZW) encoding. To encode msg using LZW, one starts with dictionary containing basic building blocks from which msg is constructed, but as larger units found in msg they added to dictionary, meaning future occurrences of those units can be encoded as single, rather than multiple dictionary refs.
- GIF (Graphic Interchange Format) = dictionary encoding system, reduces # of colours that can be assigned to pixel to only 256
    - RGB combo for each of these colors encoded using 3 bytes and these 256 encoding stored in table (a dictionary) called palette. Each px in img can then be rep'ed by single byte whose value indicates which of the 256 palette entires reps the px's color.
    - GIF is lossy system when applied to arbitrary imgs because colors in palette may not be identical to colors in original img.
    - GIF can obtain additional compression using LZW techniques. Final dictionary consists of original palette and collection of pixel patterns
    - 1 of colours in GIF palette assigned value "transparent", meaning bg shows through anything assigned this color. This make GIF logical choice in simple animation apps but GIF unsuitable for apps of higher precision
- JPEG (Joint Photographic Experts Group) = for compressing color photos, JPEG baseline standard is standard choice, requires a sequence of steps concerning human eye's limitations, eye more sensitive to brightness changes than changes in color
    - starting from image that is encoded in terms of luminance & chrominance components, 1st step to average chrominance values over 2 by 2 pixel squares. This reduces size of the chrominance info by factor of 4 while preserving all original brightness info. Result is significant degree of compression with noticeable loss of image quality
    - Next step to divide image into 8 by 8 pixel blocks and compress info in each block as unit, applying discrete cosine transform. This transformation converts original 8 by 8 into another block whose entries reflect how pixels in original block relate to each other rather than actual pixel values.
    - Values in new block under predetermined threshold replaced by 0s
    - JPEG baseline standard normally color images by factor of at least 10 and often by as much as 30
- TIFF (Tagged Image File Format) = standard format for storing photographs along with related info such as date, time and camera settings
    - image itself is stored as RGB pixel components with compression
    - TIFF collection of standards includes data compression techniques designed for compressing images of text documents
- MP3 (MPEG Layer 3) = takes advantage of properties of human ear, removing details that human ear cannot precieve
    - Temporal Masking = short period after loud sound, human ear cannot detect softer sounds that would otherwise be audible
    - Frequency Masking = sound at 1 frequency tends to mask softer sounds at nearby frequencies
    - Transfer rates of no more than 64 Kbps

Summary

In this week, we learned what data compression is and what is meant by lossless and lossy compression. Also we looked at the different file format used to compress Image and Audio files. Alongside this, we explored in detail how each file format derives compression using dictionary or run-length encoding techniques.

7.1 Computer Architecture and 7.2 Machine Language & Execution

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-11-11 7:23 PM
Author:	SUKHJIT MANN		
Tags:	ALU, Bus, Compiler, CU, Data Transfer, Interpreter, IR, Jump, Load, Opcode, Operand, PC...		

Cornell Notes	**Topic:** 7.1 Computer Architecture 7.2 Machine language & Execution	Course: BSc Computer Science Class: How Computer Work [CM1030]-Lecture Date: November 11, 2019

Essential Question:

What are the inner workings of a CPU and how it processes/stores machine readable code?

Questions/Cues:

- How is data moved between Memory and the CPU
- What is a bus?
- What is ALU?
- What is CU?
- What are Registers?
- What is a Cache?
- What is an instruction?
- What is PC?
- What is IR?
- What is Opcode and Operands?
- What are various instructions performable on data?
- What is Conditional Jump?
- What is a compiler versus an interpreter?

Notes

- Data moved between Memory & CPU via bus
- Bus = Set of wiring connect CPU & Memory and allows fast interchange between two of them
- Arithmetic logic unit (ALU) = performs numerical mathematical calculations inside CPU
- Control Unit (CU) = control running of prog
- Registers = Another bit of memory that store data, much tinier than Main Memory, but actually on CPU; accessed very quickly compared to main memory
    - CPU does calculations with data stored on registers
- Load from Memory to Register -> From register, manipulate data using ALU -> ALU writes back to Register --> data written back to memory
- Cache = smaller, but faster set of memory you can access more quickly than main memory, slower than registers
    - located either on CPU chip or very close to it

    - used to store values CPU likely to use
    - Lot more data than registers, act similar to RAM but faster
    - Performance of comp depends on access to memory; how big & fast cache is
- Instruction = for now, code that acts on data and registers
- Code is stored in memory like data, called stored program concept
- Program Counter (PC) = tells CPU where the current instruction is in memory
- Instructions Register (IR) = which actually current instruction is stored
    - Whenever CPU finish current instruction, looks at PC and gets data from that memory location, loads into instruction register, whatever binary pattern is in IR it will intrepreted as instruction and executed. Sometimes called **fetch-execute cycle**.
- Opcode (Operation Code) = start of instruction
    - Opcode like function name in prog lang
- Operands = rest of the instruction, various other data ie. like numbers to add, those numbers are operands
    - Operands lot like arguments of that function, data passing into function
    - Operands not actual values like 312 but number of a register on CPU. So ADD 312 means Add value in register 1 and add value in register 2 and store new value in register 3
- Load instruction = load data from memory into registers
- Data Transfer Instruction = moves data from 1 part of a comp to another
- Store instruction = Store is exactly the same in reverse, grabs something out of register and puts in memory
- Jump instruction = example of a control statement, instead of moving on to next instruction, jump to different instruction somewhere else in memory
    - puts different number in PC
    - takes last 2 parts of operand & copies them into PC
- Conditional Jump = takes value in register given by 1st operand, and compares it to value of register zero, if values same; jump is done, otherwise CPU carries on & moves to next instruction in sequence
- Compiler = takes high-level prog lang & convert directly machine instructions
- Interpreter = instead of converting code all at once, it will convert prog lang like JS into machine instruction as it running or line-by-line.

In this week, we learned about computer architecture, the different units inside of a CPU about the basics of machine readable code and simple instructions that are given to the CPU and how they are performed.

7.1 Computer Architecture and 7.2 Machine Language & Execution-Reading

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-11-11 7:22 PM
Author:	SUKHJIT MANN		
Tags:	ALU, Bus, CU, Data Transfer, IR, Jump, Load, Opcode, Operand, PC, Store		

Cornell Notes	**Topic:** 7.1 Computer Architecture 7.2 Machine language & Execution-Reading	Course: BSc Computer Science Class: How Computer Work [CM1030]-Reading Date: November 11, 2019

Essential Question:

What are the inner workings of a CPU and how it processes/stores machine readable code?

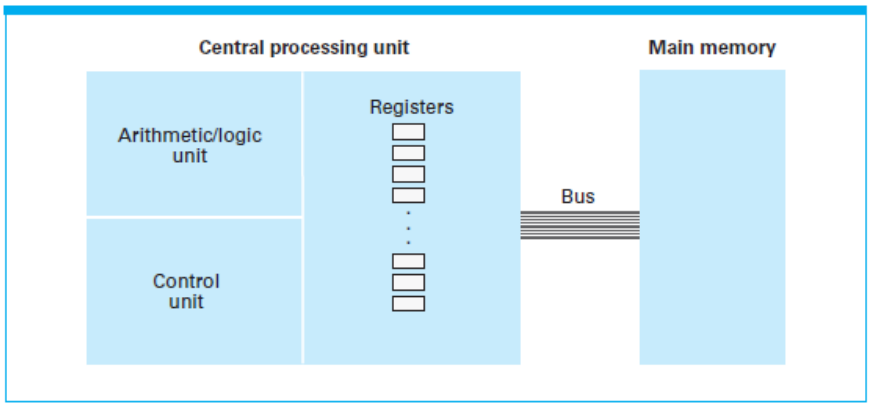
Questions/Cues:

- What controls the manipulation of data?
- What is a Motherboard?
- What is an Arithmetic/Logic Unit?
- What is a Control Unit?
- What is a Register Unit?
- What is a bus?
- What is cache memory?
- What is a stored-program concept?
- What is Machine Language/Machine Instruction?
- What is RISC/CISC?
- How can machine instructions be categorized?
- What is an instruction register?
- What is a program counter?
- What is the machine cycle of a CPU?

Notes

- CPU (Central Processing Unit) = circuitry in a comp that controls the manipulation of data
    - today's desktop comps, CPU packaged as small flat squares (2 x 2 inches)
    - Small size processors in smartphones & mobile internet Devices called microprocessors
- Motherboard = Comp's main circuit board
    - CPU plug into socket on motherboard via connecting pins
- Arithmetic/logic unit = circuitry that performs operations on data (addition & subtraction)
- Control Unit = circuitry for coordinating machine's activities
- Register unit/Registers = contains data storage cells (similar to RAM) called registers; used for temp storage of info within CPU

Figure 2.1 CPU and main memory connected via a bus



- General-purpose registers = serve as temp holding places for data being manipulated by CPU
  - hold inputs to ALU unit & provide storage space for results

- To perform operation on data in main memory: CU transfers data from memory to GPR -> informs ALU which registers hold the data -> activates appropriate circuitry within ALU -> tells ALU which register should receive result
- Bus = purpose of transferring bit patterns between CPU and main memory
  - Through bus, CPU reads data from main memory by supplying address of pertinent memory cell along with electronic signal telling memory circuitry that it's supposed to retrieve data in indicated cell.
  - CPU writes data in memory by providing address of destination cell & data to be stored together with appropriate electronic signal telling main memory that it's supposed to store data being sent to it

Figure 2.2 Adding values stored in memory

Step 1. Get one of the values to be added from memory and place it in a register.

Step 2. Get the other value to be added from memory and place it in another register.

Step 3. Activate the addition circuitry with the registers used in Steps 1 and 2 as inputs and another register designated to hold the result.

Step 4. Store the result in memory.

Step 5. Stop.

- Cache memory = portion (perhaps several hundred KB) of high-speed memory located within the CPU itself. In this special memory area, comp attempts to keep copy of that portion of main memory that of current interest. Data transfer normally between registers & main memory now reduced to be made between registers & cache memory. Changes made to cache memory then transferred to main memory at more opportune time. Result is CPU that can execute machine cycle more rapidly because not delayed by main memory communication
- Stored-program concept = idea of storing comp's prog in its main memory
- Machine language = collection of instructions along with encoding system encoded as bit patterns
- Machine instruction = instruction expressed in machine lang
- Reduced instruction set computer (RISC) = that a CPU should be designed to execute minimal set of machine instructions
  - machine designed using RISC is efficient, fast & less expensive to manufacture

- Complex instruction set computer (CISC) = CPUs with ability to execute large number of complex instructions; even though many are redundant
    - Argument in favour states more complex CPU better cope with ever-increasing complexities of today's software
- Machine's instructions can be categorized into 3 groupings: (1) data transfer, (2) ALU & (3) Control group
- Data Transfer = group consists of instructions that request movement of data from one location to another
    - rare for data being transferred is erased form its original location; process involved more like copying data rather than moving it
    - LOAD instruction = request to fill GPR with contents of memory cell
    - STORE instruction = request to transfer contents of register to memory cell
    - I/O instructions = commands for communicating with devices outside CPU-main memory context
- Arithmetic/Logic = consists of instructions that tell CU to request activity within ALU. ALU capable of performing operations other than basic arithmetic operations. Additional operations are Boolean operations AND, OR, & XOR. Also includes operations that allow contents of registers to be moved right/left within register known as SHIFT or ROTATE
- Control group = instructions that direct execution of program rather than manipulation of data. Contain family of JUMP (or BRANCH) instructions used to direct CPU to execute instruction other than next one in line. JUMP instructions appear in two ways, unconditional jumps "Skip to step 5" or conditional jump "If value obtained is 0, then skip to step 5". Distinction is conditional jump results in "change of venue" only if certain condition is satisfied.

Figure 2.3 Dividing values stored in memory

Step 1. LOAD a register with a value from memory.

Step 2. LOAD another register with another value from memory.

Step 3. If this second value is zero, JUMP to Step 6.

Step 4. Divide the contents of the first register by the second register and leave the result in a third register.

Step 5. STORE the contents of the third register in memory.

Step 6. STOP.

- Encoded version of machine instruction consists of two parts, opcode and operand field

Figure 2.4 The architecture of the machine described in Appendix C

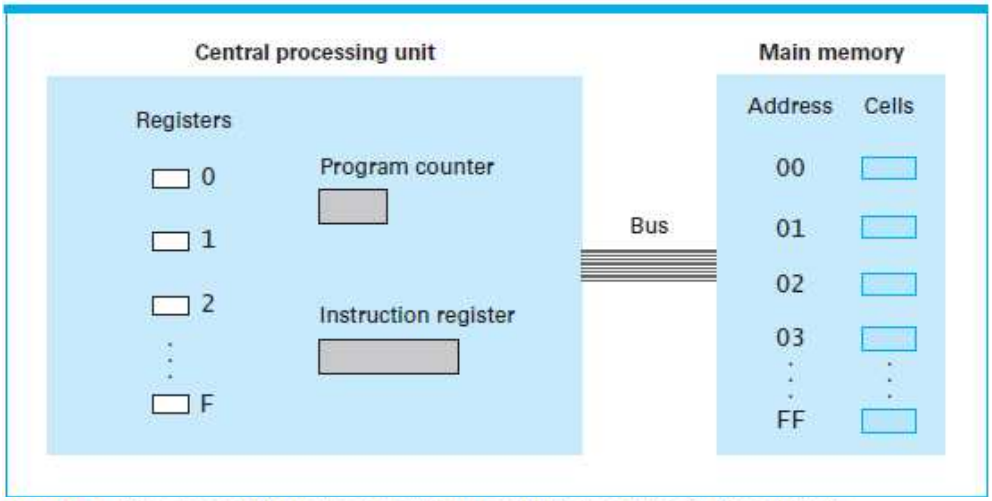


Figure 2.5 The composition of an instruction for the machine in Appendix C

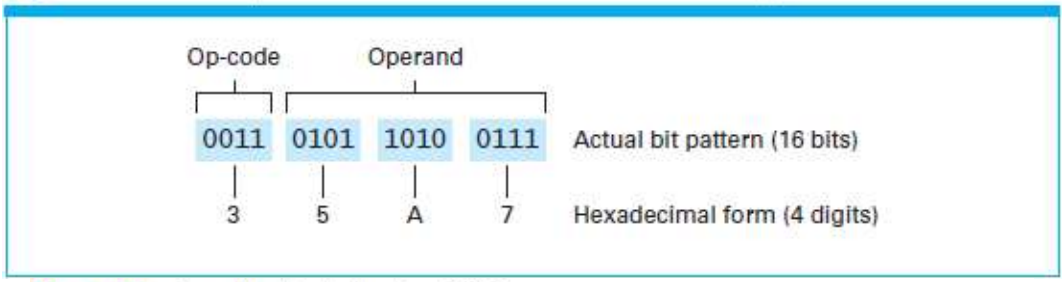


Figure 2.6 Decoding the instruction 35A7

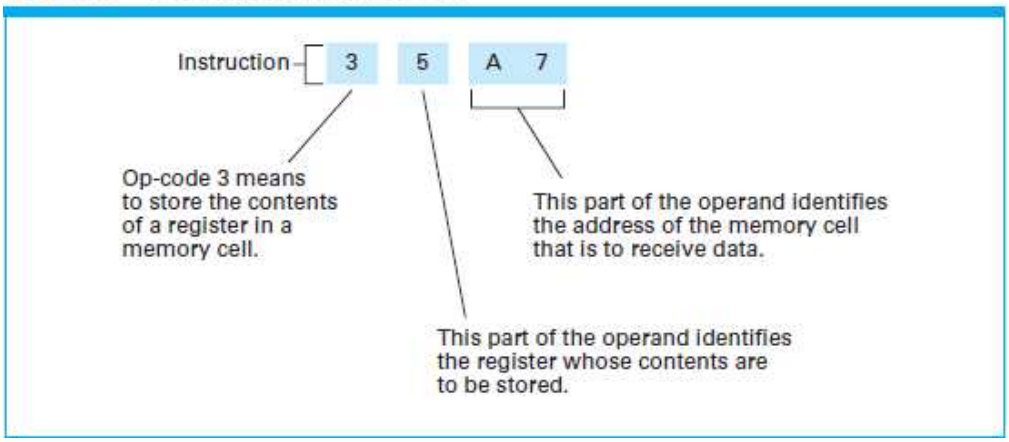


Figure 2.7 An encoded version of the instructions in Figure 2.2

Encoded instructions	Translation
156C	Load register 5 with the bit pattern found in the memory cell at address 6C.
166D	Load register 6 with the bit pattern found in the memory cell at address 6D.
5056	Add the contents of register 5 and 6 as though they were two's complement representation and leave the result in register 0.
306E	Store the contents of register 0 in the memory cell at address 6E.
C000	Halt.

- Instruction register = used to hold instruction being executed
- Program counter = contains address of next instruction to be executed; serving as machine's way of keep track of where it's in prog
- Machine cycle = three-step process guided by algorithm for CPU; repeated continuously
    - Fetch step = CPU request main memory provide with instruction that is stored at address indicated by PC. Fetch involves retrieving content of 2 memory cells from main memory since instruction is 2 bytes long. CPU places instruction received from memory in instruction register & then increments PC by 2 so counter contains address of next instruction stored in memory
    - Decode step = involves breaking operand field into its proper components based on instruction's opcode.
    - Execute step = CPU executes instruction by activating appropriate circuitry to perform requested task. Once executed finished, CPU begins machine cycle fetch step again.

Figure 2.8 The machine cycle

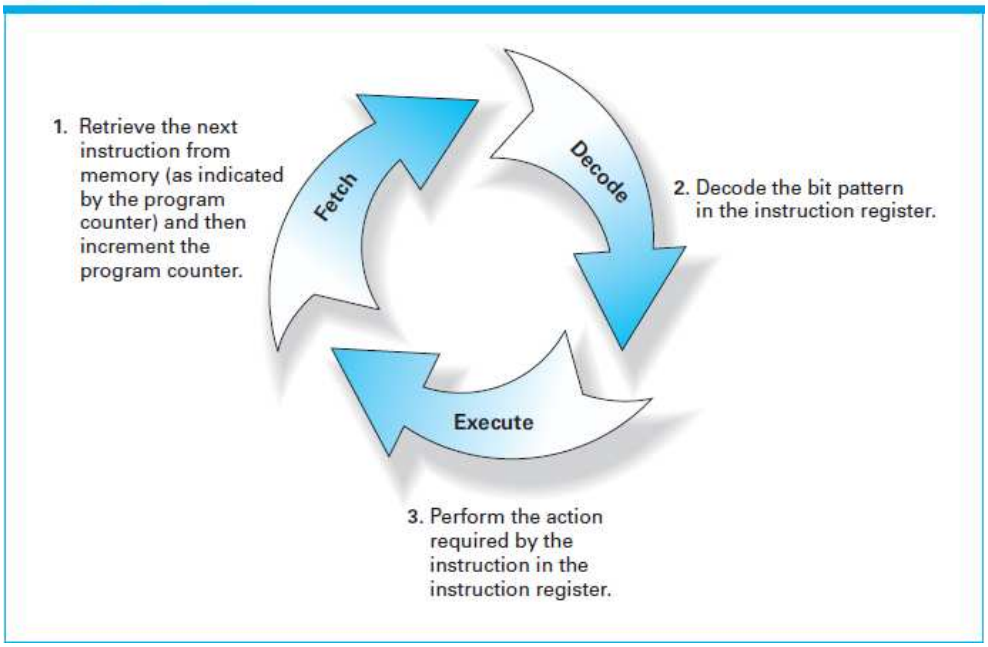


Figure 2.9 Decoding the instruction B258

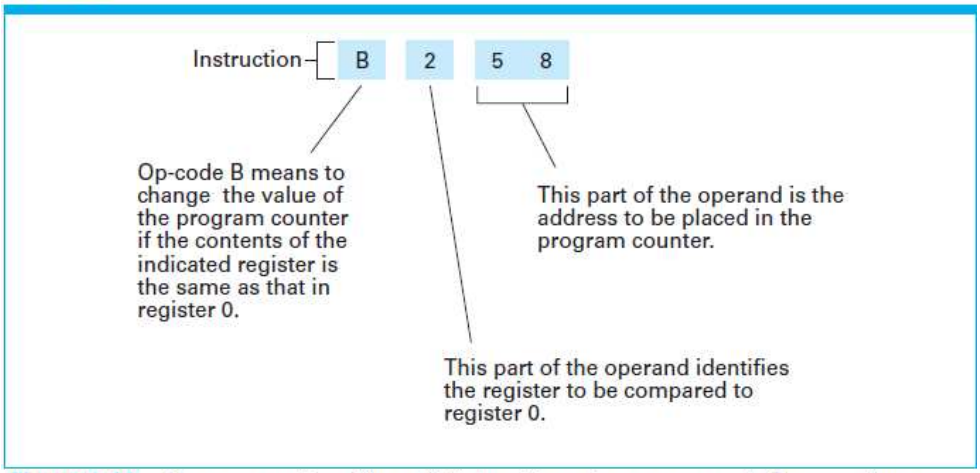


Figure 2.10 The program from Figure 2.7 stored in main memory ready for execution

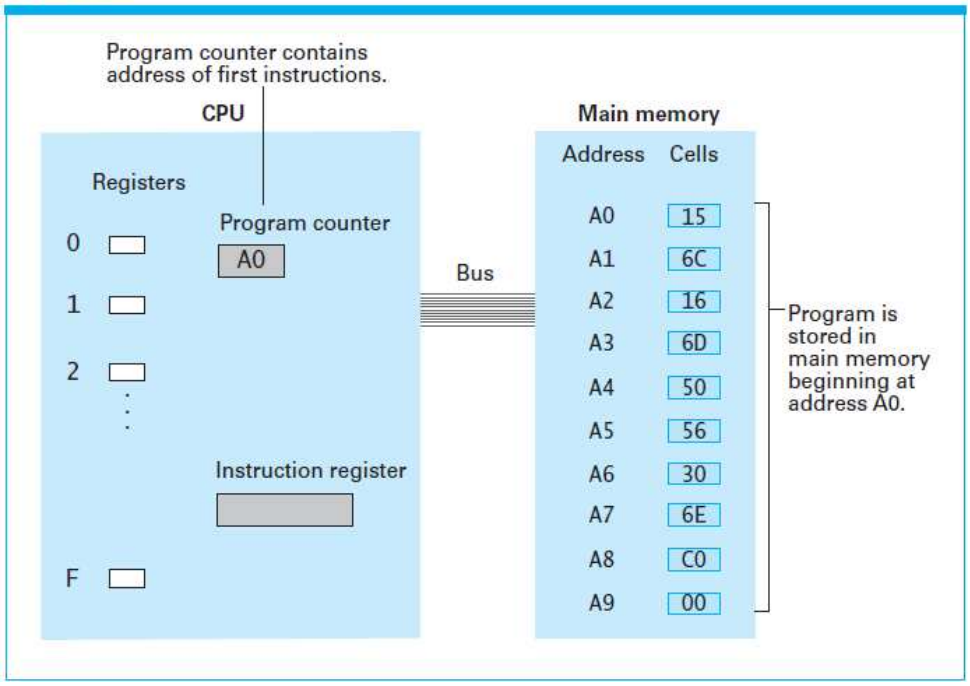
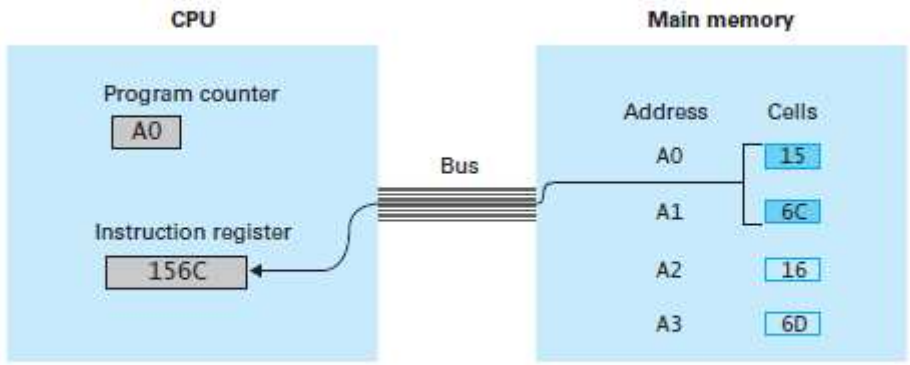
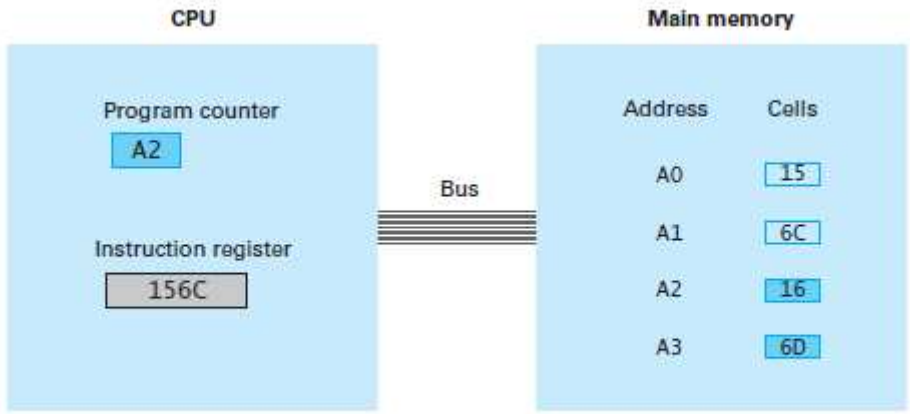


Figure 2.11 Performing the fetch step of the machine cycle



a. At the beginning of the fetch step the instruction starting at address A0 is retrieved from memory and placed in the instruction register.



b. Then the program counter is incremented so that it points to the next instruction.

In this week, we learned about computer architecture, the different units inside of a CPU about the basics of machine readable code and simple instructions that are given to the CPU and how they are performed.

8.1 Communicating with Devices

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-11-19 5:04 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 8.1 Communicating with Devices	Course: BSc Computer Science Class: How Computer Work [CM1030]-Lecture Date: November 19, 2019

Essential Question:

What are peripherals and the various technologies that make their use possible?

Questions/Cues:

- What are peripheral devices?
- What is Controller?
- What is GPU?
- What is USB?
- What is Bluetooth?
- What is Memory mapped IO?
- What is DMA?

Notes

- Peripheral devices = not core of the comp, attached to the edge or periphery
- Peripheral devices also use bus for data transfer
- Controller = decodes what's happening on the bus and sends it down to device; bit of hardware between bus and peripheral device
- Graphics Card or Graphics processing unit (GPU) = highly powerful processor; highly optimized bit of silicon, different model from CPU. It is parallel programming chips which produce graphics at great speed. Very specialist bit of control hardware.
- USB (Universal Serial Bus) = type of controller that handles many different types of devices, allows data transfer between CPU & device in a way that's independent of details of exact device
- Bluetooth = similar to WiFi, but designed to communicate over internet & much shorter range. Designed to communicate between peripheral devices and computer. Controller inside that connect to main bus. Uses it's own wireless transmitter to communicate with devices; no wire.
- Type on keyboard -> CPU uses load instruction to request data from keyboard -> Send signal along bus to controller -> Controller will then send character typed back along bus -> Ends up in register (probably going via memory)
- Memory mapped IO (Memory mapped Input & Output) = Each controller has a memory address, for example keyboard, if you read from keyboard's memory address you'll get last key that's typed to keyboard

- DMA (Direct Memory Access) = allows certain devices like HDD to communicate directly with memory, faster access to devices without clogging up CPU with lots of loads & stores

Summary

In this week, we learned about peripherals or external devices connected to the computer outside its core mechanism and the various technologies that go into making connections with external devices possible.

8.1 Communicating with Devices-Reading

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-11-20 4:29 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 8.1 Communicating with Devices-Reading	Course: BSc Computer Science Class: How Computer Work [CM1030]-Lecture Date: November 20, 2019

Essential Question:

What are peripherals and the various technologies that make their use possible?

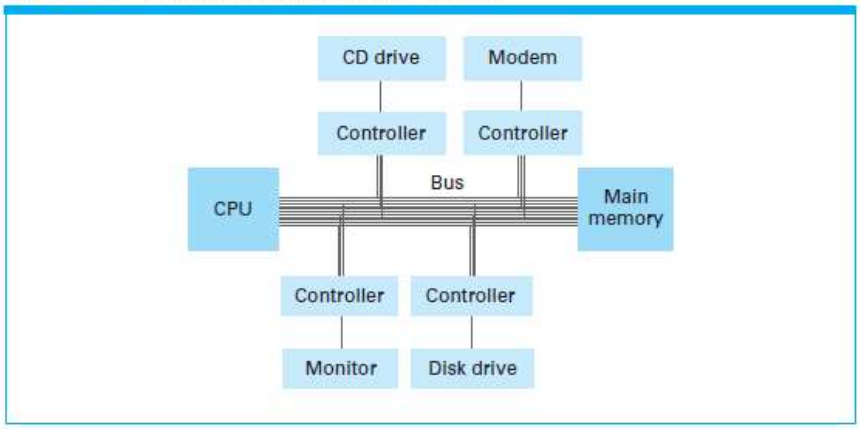
Questions/Cues:

- What is Controller?
- What is USB/FireWire?
- What is Memory mapped IO?
- What is DMA?
- What is Handshaking?
- What is Serial/Parallel Communication?
- What is Multiplexing?

Notes

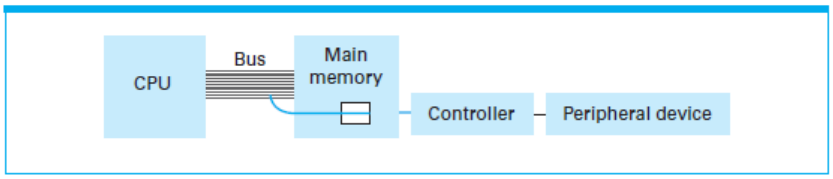
- Controller = handles comms between comp and other devices, may consist of circuitry permanently mounted on comp's mobo(motherboard) or it may take form of board that plugs into slot on mobo.
    - connects via cables to peripherals devices within case or perhaps to connector called a port on back of comp where external devices can be attached
    - sometimes each with own memory circuitry and simple CPU which performs program directing activities of controller
    - Each controller comms with comp itself by means of connections to same bus that connects CPU & Main memory. Like this controller can monitor signals between CPU & Main memory & inject its own

Figure 2.13 Controllers attached to a machine's bus



- USB(Universal Serial Bus)/Firewire = single controller to handle a variety of devices
- Memory-mapped IO = memory locations assigned to controller and peripherals, Comp's IO devices appear as memory address
    - Comp interacts directly with via memory address; main memory ignores this

Figure 2.14 A conceptual representation of memory-mapped I/O



- DMA (Direct Memory Access) = ability of controller to communicate directly with main memory with interact of CPU, which frees up CPU for other tasks
    - DMA also has effect of complicating communication over comp's bus since the data between CPU & Main memory, CPU & each controller and between each controller & main memory. Coordination of activity is major design issue as CPU & controller compete for bus access.
    - DMA impediment known as von Neumann bottleneck because it a consequence of underlying von Neumann architecture in CPU fetches instructions from memory over central bus
- Handshaking = comp & peripheral device exchange info about device's status & coordinate their activities
    - involves status word, a bit pattern that is generated by peripheral device & sent to controller. Status word is bit map in which bits reflect conditions of device
- Parallel Communication = several signals are transferred at the same time, each on separate "line"
    - Capable of transferring data fast but requires relatively complex comms path
- Serial communication = transferring signals one after the after over single line
    - requires single path for comms, this why is popular over parallel comms
- Rate at which bits are transferred from one computing component to another is measured in bits per second (bps)
    - Kbps (Kilo-bps, 1000 bps), Mbps(mega-bps = 1 millions bps) and Gbps(giga-bps = 1 billion bps)
    - Note distinction between bits and bytes 8 Kbps = 1 KB per second. Lower case b means bit whereas an uppercase B means byte
- Multiplexing = encoding or interweaving of data so that single comm path serves the purpose of multiple paths
- High bandwidth means comm path has the ability to transfer bits at high rate as well as capacity to carry large amounts of info simultaneously

Summary

In this week, we learned about peripherals or external devices connected to the computer outside its core mechanism and the various technologies that go into making connections with external devices possible.

9.1 What is an Operating System and 9.2 Operating system function: application loading, file systems & memory managment

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-11-27 11:18 AM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 9.1 What is an Operating System? 9.2 Operating system function: application loading, file systems, & memory management	Course: BSc Computer Science Class: How Computer Work [CM1030]-Lecture Date: November 27, 2019

Essential Question:

What is an operating system and what are the main functions of an operating system?

Questions/Cues:

- What are utilities?
- What is a GUI?
- What is the Window Manager?
- What is the Kernel?
- What is the file manager?
- What is Folder/Directory?
- What is the memory manager?
- What is paging?

Notes

- Utilities =in between OS and application. Utilities are actually apps, but they're mini-apps bundled with the OS
- GUI(Graphical User Interface) = the visual portion of OS, is the first thing most people think of when OS is mentioned, it's also a service provided by the OS
    - Often basic buttons and text entry boxes are supplied by OS or libraries associated with the OS via GUI library, but GUI library can also be third-party
- Window manager = part of OS that provides space on screen, it's a service provided by OS
- Kernel = fundamental core of OS and it's what provides most important functionality like access to hardware, networking & files
- The OS Kernel manages device drivers, provides a standard interface to all apps and behind the scenes it comms with the drivers which interact with hardware itself
- When data stored to HDD, those files managed by OS with function called the file manager
- File manager = software that manages how files and their metadata is stored on disk. Also looks after folders

- Folders also called directories, terms are interchangeable
- Folder or Directory = contains files; way of grouping files together, can also contain folders/directories it creates a hierarchy
- Each app on comp needs some memory to function, that's where app stores all data it's working on
- When app starts, OS or part of OS called memory manager allocates app some memory or rather gives it range of memory address it can work with. OS can stop one app from accessing memory area of another app
- Memory manager = allocates memory to apps & makes sure that one app can't interfere with memory area of another app.
    - When memory fills up, a new app starts or an app asks for more memory then memory manager takes app not active at present moment & moves it memory area of app from main memory onto HDD. App is still loaded, data still there, could be reloaded into memory whenever, but it's stored on HDD. Data will be slower to access, but frees up space for other apps.
- Paging = process of copying memory to HDD. A page is chuck of memory attached to an app that is saved to disk.
    - When we use disk space as memory, we called it virtual memory, disk space is acting as if it were memory

Summary

In this week, we learned about what an Operating system is, the different functionalities/services it provides. Also, we delved deeper into what functions of the kernel (core of OS) are, for example file and memory management.

9.1 What is an Operating System and 9.2 Operating system function: application loading, file systems & memory managment Reading

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-11-27 1:33 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 9.1 What is an Operating System? 9.2 Operating system function: application loading, file systems, & memory management Reading	Course: BSc Computer Science Class: How Computer Work [CM1030]-Reading Date: November 27, 2019

Essential Question:

What is an operating system and what are the main functions of an operating system?

Questions/Cues:

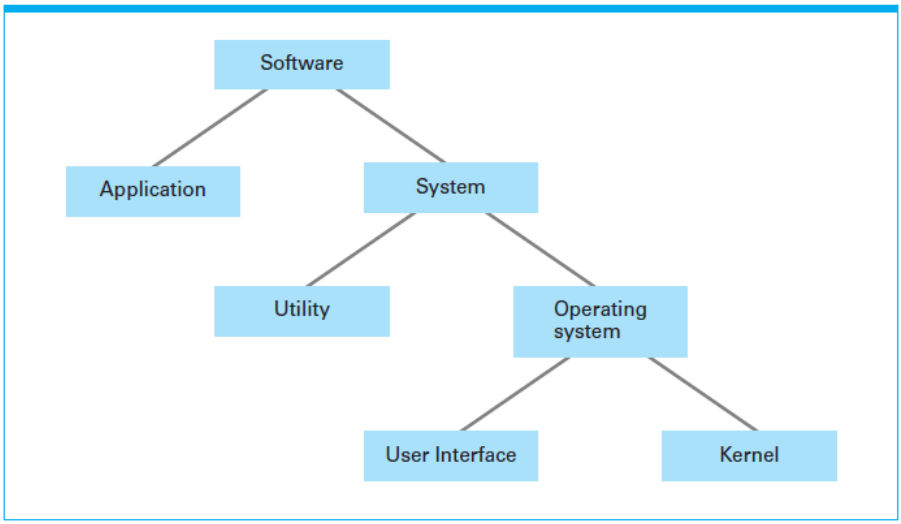
- What is Application software?
- What is System software?
- What is Utility software?
- What is an User interface?
- What is a shell?
- What is a GUI?
- What is the window manager?
- What is the kernel?
- What is the file manager?
- What is a file/directory?
- What is a directory path?
- What is device driver?
- What is the memory manager?
- What is paging?
- What is the scheduler/dispatcher?
- What is ROM?
- What is a boot-loader and what is meant by the term "booting the computer"?
- What is a firmware update?

Notes

- Application software = consists of programs for performing task particular to the machine's utilization
    - Examples include spreadsheets, accounting systems, games and much more
- System software = performs tasks that common to comp systems in general. System software provides the infrastructure that the app software requires

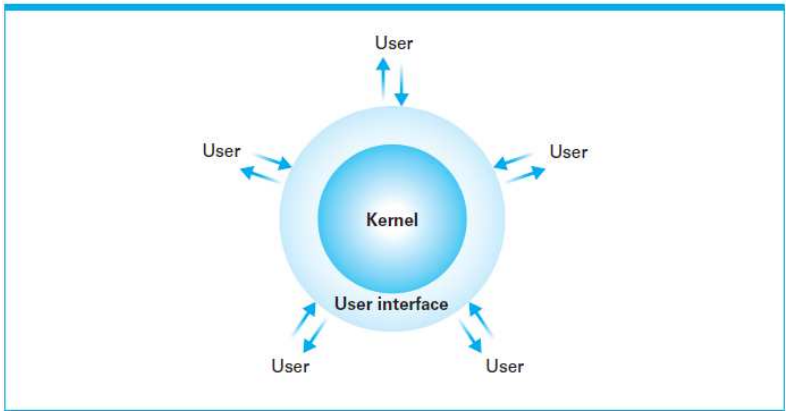
    - Within system software, there are two categories of division, Operating system & utility software
- Utility software = programs for performing activities fundamental to comp installations, but not included in OS or consists of software units that extend (or customize) the capabilities of OS, for example the ability to format HDD or copy file from HDD to CD, functionality not implemented by OS itself but instead provided by the means of a utility program.

Figure 3.3 Software classification



- User Interface = portion of OS that handles comp's ability to communicate with users
- Shell = communicated with users through textual messages using a keyboard and monitor screen
- Graphical user interface (GUI) = in which objs to be manipulated such as files & programs, are rep'ed pictorially on the display as icons. These allows users to issue commands via one of several common input devices such mouse or styluses. Advances in touch screen tech has allowed to user to use their finger as input device.

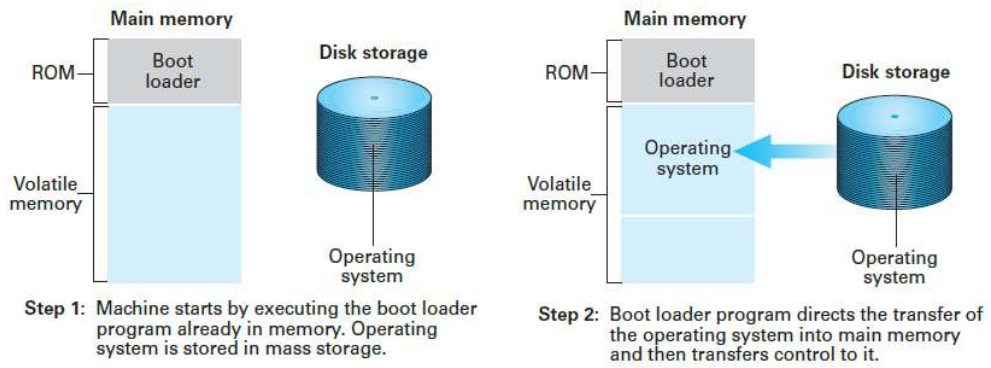
Figure 3.4 The user interface acts as an intermediary between users and the operating system's kernel



- Window manager = allocates blocks of space on screen called windows & keeps track of which app is associated with each window.
    - When app wants to display something on screen, it notify window manager and WM places desired image in window assigned to app. When mouse is clicked, WM computes mouse's location and notifies appropriate app of mouse action
- Kernel = internal part of OS, contains software components that perform the very basic functions required by comp installation.
- Kernel coordinates or overlooks file management, memory management & use of device drivers
- File Manager = coordinates the use of comp's mass storage facilities; maintains records of all the files stored in mass storage. Also in the records of such files, the FM

tells us where file is located, which users are allowed to access the various files, and which portions of mass storage are available for new files or extensions to existing files.
    - The records are kept in the same location as the files they speak of, so that when the storage goes online, FM can retrieve the record and know what's stored in that portion of mass storage.
- File/directory = grouping of files into a bundle
    - This allows for users to organize files according their purpose by placing related file in same folder
    - Folder/directory can contain other folders/directories called sub-directories which help to form a hierarchy
- Directory path = A chain of directories within directories
    - paths are expressed by listing the directories along the path separated by slashes
- Any access to a file by other software is regulated by FM, it obtain by process called "opening the file". If FM approves requested access, it provides info needed to find and manipulate file
- Another component of the kernel consists of a collection of device drivers
- Device driver = software units that communicate with the controllers (or directly with peripheral devices) to carry out operations on peripheral devices attached to comp.
    - Each device driver is designed particularly for its particular device & translates generic requests into more tech step required by device assigned to the particular driver.
- Memory manager = coordinates the comp's use of main memory
    - When comp is performing one task at a time, prog for such task is placed at predetermined location in main memory, executed, & then replaced by prog for performing next task
    - In case of multi-tasking needs, many progs & blocks of data must reside in main memory at same time. MM must find find & assign memory space for these needs and ensure that actions of each prog are restricted to prog's allotted space. Moreover, MM must keep track of memory area no longer occupied
- Paging = when total main memory space required exceeds space available in comp, MM creates additional temp memory by rotating progs & data back and forth between main memory and mass storage. This large temp memory space created by paging is called virtual memory.
- Scheduler = in a multi-programming system, it determines which activities are to be considered for execution
- Dispatcher = controls the allocation of time to the activities deemed ready for execution by the scheduler
- Boot strapping (booting) = performed by a comp each time it is turned on, transfers OS from HDD into Main memory
- Read-only memory (ROM)= small portion of a comp's main memory where the CPU expects to find its initial program (in most cases the OS) is constructed by special non-volatile memory cells. Its contents can be read not altered.
    - This is done to solve the dilemma where main memory is wiped of data stored in it when comp turned off.
- Boot-loader = stored permanently in ROM and is the prog initially executed when comp turned on. Instructions in boot-loader direct CPU to transfer OS from predetermined location into volatile area of main memory, then boot-loader directs CPU to execute jump instruction to that area of memory. Overall process of executing boot-loader and hence starting OS is called booting the comp.

Figure 3.5 The booting process



- Firmware update = act of updating the OSes and boot-loaders stored in ROM

Summary

In this week, we learned about what an Operating system is, the different functionalities/services it provides. Also, we delved deeper into what functions of the kernel (core of OS) are, for example file and memory management. Lastly, we touch on the process of actually starting the OS and what's involved.

10.1 Operating System Security

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-11-28 3:33 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 10.1 Operating System Security	Course: BSc Computer Science Class: How Computer Work [CM1030]-Lecture Date: November 28, 2019

Essential Question:

What are security measures are in place on an Operating System to make sure the environment and the user are safe?

Questions/Cues:

- What is a login?
- What is a user?
- What is an administrator?
- What are the two levels of machine instruction?

Notes

- Login = most security function OS provides, so that you don't the right to access comp, you can't do it because you have to enter a password or in case of a phone a pin #
- User = Can access own files, change their own personal data & use all the apps on comp
    - One user cannot access other user's files or install software or make changes to the comp
- Administrator= has complete rights to comp, can install new apps, add new users, delete user accounts and even sometimes go in and change user files
- So another key function of OS is defining who can changes to comp and who is simply able to just "use" it.
- Two levels of machine instruction:
    1. Instructions any bit of code can execute ADD, LOAD, STORE, JUMP
    2. Special instructions that only certain bits of code can do like changing memory area than an app can access
        - They can called Privileged Instructions, means they are restricted to Privileged Code. Simple way of thinking of privileged code is that it is the code that exists inside the kernel of OS
- Kernel like admin, it can do whatever it feels like and apps are like users. Apps are restricted in what can do, have to follow rules of OS. If apps need to access privileged instructions, they do it via kernel
- Privilege levels are part of hardware of CPU, not part of OS, so impossible to get around.

Summary

In this week, we learned about the types of different users on an OS and building on this further, we explored how a similar relationship exists in the levels of machine instruction.

10.1 Operating System Security Reading

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-11-28 4:11 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 10.1 Operating System Security-Reading	Course: BSc Computer Science Class: How Computer Work [CM1030]-Reading Date: November 28, 2019

Essential Question:

What are security measures are in place on an Operating System to make sure the environment and the user are safe?

Questions/Cues:

- What is an account?
- What is a login procedure?
- What is an administrator/Super-user?
- What is auditing software?
- What is privilege/ non-privileged mode?

Notes

- Account = a record within OS containing entries such as user's name, password and privileges to be granted to that user.
- Login procedure = a sequence of transactions in which user establishes initial contact with a comp's OS
- Administrator/Super-user = establishes accounts on OS, gains high privileged access to OS by identifying as admin during login. From here, admin can alter settings within OS, modify critical software packages, adjust privileges granted to other users & perform variety of other maintence activities that denied to normal users
    - Also being a super-user, an admin is able to monitor activity with comp system in an effort to detect destructive behavior, whether malicious or accidental
- Auditing software = record and then analyze the activities taking place within comp system.
    - may also identify activities within user's account that don't conform to that user's past behavior
    - Also designed to detect the presence of sniffing software, software when left running on comp records activities and later reports them to a would-be intruder.
- CPU's designed for multi-programming systems contain special-purpose registers in which OS can store upper & lower limits of a process's allotted memory area. If process outside allotted area, CPU does interrupt sequence to transfer control back to OS. Even so, process could merely change upper-limit of SP-register for more memory without permission of OA

    - To counteract this, multi-programming CPU's designed to operate in two privilege levels, "privilege mode" & non-privileged mode"
- Privileged mode = CPU able to execute all instructions in its machine lang. Also it can execute special instructions only available in privilege mode called privileged instructions
    - list of acceptable instructions in non-privileged mode is limited
    - Examples of privileged instructions are: instructions that change contents of memory limit registers & instructions that change current privilege mode of CPU
    - Attempt to execute privilege instruction when CPU in non-privileged mode causes an interrupt. Interrupt converts CPU to privilege mode & transfers control to interrupt handler within OS
- When comp first turned on, CPU is in privileged mode, when OS starts at end of boot process, all instructions are executable

Summary

In this week, we learned about the types of different users on an OS and building on this further, we explored how a similar relationship exists in the levels of machine instruction.

11.1 History of Operating Systems & 11.2 Processes

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-12-10 1:37 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 11.1 History of Operating Systems 11.2 Processes	Course: BSc Computer Science Class: How Computer Work [CM1030]-Lecture Date: December 10 2019

Essential Question:

What were early computers like and what are the processes that simulate to give the illusion of simultaneous execution in computers?

Questions/Cues:

- What is the name of the first fully-operational computer?
- What is thought of when mentioning early computers?
- What was the function/role of a punched card in terms of computers in the early days of computing?
- What was a time-sharing computer in the 70's like?
- What is Load Balancing?
- What is a process?
- What is thread?
- What is a time slice?
- What is an Interrupt?

Notes

- It was called EDSAC & it was built at University of Cambridge, designed by Maurice Wilkes
    - It was first "fully-operational" computer and in use, but not the first computer to be built, ACE built at University of Manchester Northern England was actually "working" first, but was a prototype designed by Alan Turing.
    - John Von Neumann, J Presper Eckert, and John Mauchly in US laid the technical foundations & came up with first plans for a comp
        - This work by used by Wilkes & Turing as basis for their machines
- Early comps consisted of no OS, screen or keyboard, just on/off switches
- Then came punched cards, which were used to enter programs into computers. To program a punch card, you would make holes in the card at the positions of the numbers you wanted to select. Those numbers represented machine instructions. The card would then be fed into special reader, then comp would interpret those machine instructions and run prog.
- In 1970's thing with a screen not a comp, but a terminal. All terminals did was input & output, & they were linked to the main computer (row of server racked box in A/C

room). Several people would be working on the same comp at same time using these terminals.
        - This was called a multi-processing or time-sharing comp
- Load Balancing = task of allocating apps to CPUs so as to most efficiently use computing resources.
    - Very important in multi-core processing
- Process = Is a coherent piece of software that is running on a comp & which can run at the same time as & independent of other processes.
    - can correspond to app or OS kernel, attached to app within
    - runs independently of other bits of code and its own memory
- Thread = Similar to process code, in that it runs independently as the same time as other bits of code, but it doesn't have its own memory.
- Time slice = After the process starts running & sends its code to CPU, it does stuff, but it only runs for a certain amount of time, after which it stops. This amount of time is called a time slice.
    - On single CPU, we are slicing up the amount of time we have on CPU between processes
    - When first process's time slice runs out, it gets an interrupt.
- Interrupt = A message sent to the process that says stop, "get off CPU, it's some else's turn"
    - The procedure of time slices & interrupts repeat for the remaining processes and then goes back to the original or first process. But processes don't always happen in order, sometimes processes are simply high-priority or that a process needs time to load a resource/data from somewhere and cannot execute until then; other processes would run and do their bit & once data is obtained it can start again instead of everything stopping and waiting.

Summary

In this week, we learned about the history of early computing & the processes behind the scenes that execute in a timely manner to provide fast & easy access to a user using a computer.

11.1 History of Operating Systems & 11.2 Processes-Reading

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2020-01-06 10:41 AM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 11.1 History of Operating Systems 11.2 Processes Reading	Course: BSc Computer Science Class: How Computer Work [CM1030]-Reading Date: December 10 2019

Essential Question:

What were early computers like and what are the processes that simulate to give the illusion of simultaneous execution in computers?

Questions/Cues:

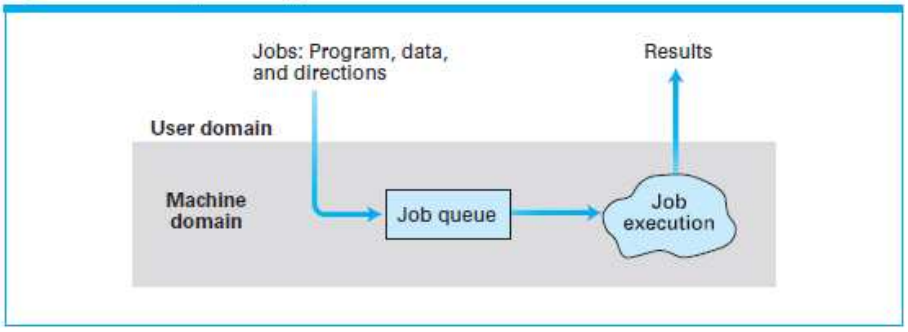
- What is a job?
- What happened when several users needed to use a machine in the early days?
- What was the initial job of an Operating System?
- What was the role of an Operator?
- What is Batch Processing?
- What is an Queue?
- What is Interactive Processing?
- What is real-time processing?
- What is time-sharing?
- What is Multi-programming?
- What is Multi-tasking?
- What is Load Balancing?
- What is Scaling?
- What is the role of the Scheduler within System's Kernel in terms of processes?
- What is the role of the Dispatcher within System's Kernel in terms of processes?
- What it meant by ready or waiting in regards to a process?
- What is a process/context switch?
- What is an interrupt and an interrupt handler?

Notes

- Job = the execution of a comp prog
    - handled as an isolated activity
        - machine was prepared for executing the prog, the prog was executed, then all tapes, punched cards, etc. had to be retrieved before the next prog preparation could begin
- When several users needed to share a machine, sign-up sheets were used to reserve machine for blocks of time in which machine was totally under user's control
- OS began as systems for simplifying prog setup & for streamlining transition between jobs

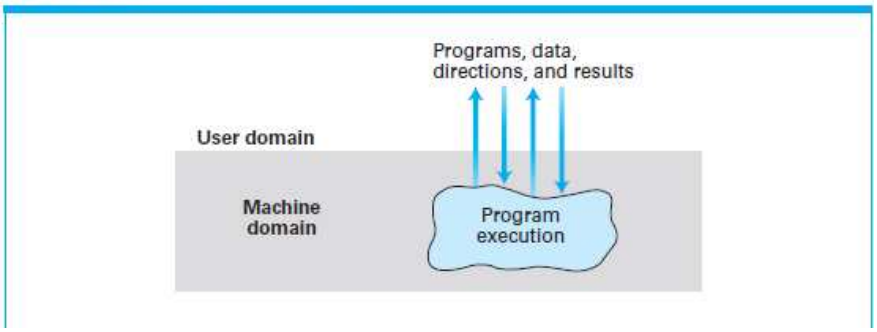
    - Early development was separation of users & equipment; eliminated people in & out of comp room
        - A comp operator was hired to operate machine. Anyone wanting a prog run was required to submit it, along with any required data & special directions about prog's requirements to operator & return later for results.
        - The operator loaded these materials into machine's mass storage where a prog called OS could read & execute them one at a time.
- Batch Processing = execution of jobs by collecting them in single batch, then executing them without further interaction with the user
    - jobs residing in Mass Storage for BP, wait for execution in a job queue
- Queue = storage organization in which objects (jobs) are ordered in first-in, first-out (FIFO) fashion.
    - No strict following of FIFO structure in most OSes, since job waiting in queue can be bumped by higher-priority job
- In early batch-processing systems, each jobs was accompanied by set of instructions explaining steps required to prepare machine for that particular job. Instructions were encoded using a system known as job control lang (JCL) & stored with job in job queue. When job selected for execution, OS printed instructions at a printer so they could be read and followed by comp operator.
- Major drawback to using comp operator as an intermediary between comp and its users is that users have no interaction with their jobs once they are submitted to operator.

Figure 3.1 Batch processing



- Interactive Processing = allows a prog being executed to carry on a dialogue with the user through remote terminals
    - In early days, terminals were a electronic typewritter by which user could type input and read comp's response that was printed on paper
    - To be successful, interactive processing must have that the actions of comp be sufficiently fast to coordinate with needs of users rather than forcing user to conform to machine's timetable.

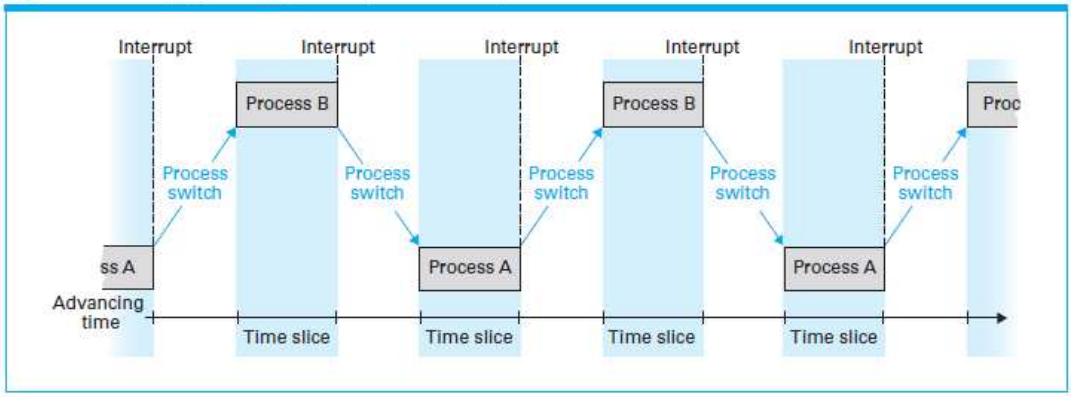
Figure 3.2 Interactive processing



- Real-time processing = Actions performed by a comp occur in real-time. Comp performs task in real-time means that the comp performs the task in accordance with deadlines in its (external real-world) environment.

- Time-sharing is feature in which OSes are designed to provide service to multiple users at the same time.
- Multi-programming= A way to implement time-sharing in which time is divided into intervals & then execution of each job is restricted to only one interval at a time. At end of each interval, current job is temporarily set aside & another is allowed to execute during next interval. By rapidly shuffling jobs back & forth, illusion of several jobs executing simultaneously is created.
    - Depending on job being executed, early time-sharing systems were able to provide real-time processing to as many as 30 users simultaneously.
- Multi-tasking = one user executing numerous task simultaneously.
- Load Balancing = dynamically allocating tasks to the various processors so that all processors are used efficiently)
- Scaling = Breaking tasks into a number of subtasks compatible with the number of processors available.
- Process = activity of executing a prog under the control of OS
    - Process state = current status of the activity
        - state includes current position in prog being executed (value of program counter) as well as the values in other CPU registers & associated memory cells; process state is a snapshot of machine at particular time
- Tasks associated with coordinating execution of processes are handled by scheduler & dispatcher within OS Kernel
    - Scheduler maintains record of processes present in comp system, introduces new processes to this pool, & removes completed processes from pool.
        - To keep track of all processes, scheduler maintains block of info in main memory called process table. Each time execution of prog is requested, scheduler creates new entry for that process in process table
            - Entry contains info such as memory area assigned to process (obtained from memory manager), priority of process & whether process is ready or waiting
- Process is ready if it's in a state in which its progress can continue; waiting if its progress is currently delayed until some external event occurs.
- Dispatcher, the component of kernel that oversees execution of scheduled processes. In time-sharing/multi-tasking system, this accomplished by multi-programming; dividing time in short segments, each called time slice (measured in mill or microseconds) & then switching CPU's attention among the processes as each is allowed to execute for one time slice.
- Process/context switch = procedure of changing from one process to another

Figure 3.6 Multiprogramming between process A and process B



- Each time dispatcher awards time slice to process, it starts timer circuit that will indicate end of slice by generating signal called an interrupt. CPU completes its machine cycle, saves its position in current process, & begins executing prog called interrupt handler which is stored at predetermined location in main memory. IH part of dispatcher, tells dispatcher how to deal with interrupt signal

Summary

In this week, we learned about the processes behind the scenes that execute in a timely manner to provide fast & easy access to a user using a computer.

12.1 Inter-process Commuication

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-12-18 12:04 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 12.1 Inter-process Communication	Course: BSc Computer Science Class: How Computer Work [CM1030]-Lecture Date: December 16, 2019

Essential Question:

What happens when multiples processes share or attempt to access the same resources?

Questions/Cues:

- What is meant by locking a resource?
- What is a semaphore?
- What is parallel software?

Notes

- Locking(Resource) = means setting a value of some bits of memory, probably literally one bit of memory. 1 for locked & 0 for unlocked
- Semaphore = special instruction in hardware that first checks whether a lock is locked & sets the lock at exactly the same time as it checks, nothing can interrupt this interaction; instruction has to be implemented directly in CPU. A flag used to control access to a particular resource.
- Deadlock = where processes wait on each other to unlock their respective resources in order to gain access to the other's resources and proceed, but both processes fail to unlock their resources and end up in a loop of sorts.
- Parallel Software = software where multiple processes are running at the same, particularly when they share resources.

Summary

In this week, we learned about the Deadlock, the locking of a resource and the remedy to Deadlock, the concept of a Semaphore.

12.1 Inter-process Commuication-Reading

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-12-21 5:48 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 12.1 Inter-process Communication	Course: BSc Computer Science Class: How Computer Work [CM1030]-Reading Date: December 16, 2019

Essential Question:

What happens when multiples processes share or attempt to access the same resources?

Questions/Cues:

- What is a flag?
- What is an interrupt disable/enable?

- What is a test-and-set instruction and what is a semaphore?
- What is a critical region?
- What is mutual exclusion?
- What is Deadlock?
- What are the conditions for Deadlock to occur?
- What is Spooling?

Notes

The file manager allocates access to files as well and allocates mass storage space for the construction of new files; the memory manager allocates memory space; the scheduler allocates space in the process table; and the dispatcher allocates time slices.

- Flag = In the context of control access, refers to a bit in memory whose states are often referred to as set and clear, rather than 1 and 0.
- Interrupt disable = instruction causes future interrupts to be blocked
- Interrupt enable = instruction that causes CPU to resume responding to interrupt signals
    - If OS starts the flag-testing routine with a disable interrupt instruction and ends it with an enable interrupt instruction, no other activity can interrupt the routine once it starts.
- Test-and-set instruction = directs CPU to retrieve the value of a flag, note the value received, and then set the flag--all within a single machine instruction.
    - Advantage here is that because CPU always completes an instruction before recognizing an interrupt, the task of testing and setting the flag cannot be split when it is implemented as a single instruction.
        - A properly implemented flag as described above is called a semaphore

- Critical Region = Sequence of instructions that should be executed by only one process at a time.
- Mutual Exclusion = Requirement that only one process at a time be allowed to execute a critical region
    - Common way of obtaining mutual exclusion to a critical region is to guard critical region with a semaphore.
    - To enter critical region, a process must find the semaphore clear and then set semaphore before entering critical region; then upon exiting critical region, the process must clear semaphore. If semaphore is found in its set state, the process trying to enter critical region must wait until semaphore has been cleared
- Deadlock = condition in which two or more processes are blocked from progressing because each is waiting for a resource that is allocated to another.

Analysis of deadlock has revealed that it cannot occur unless all three of the following conditions are satisfied:

1. There is competition for nonsharable resources.
2. The resources are requested on a partial basis; that is, having received some resources, a process will return later to request more.
3. Once a resource has been allocated, it cannot be forcibly retrieved.

- Spooling = technique of holding data for output at a later but more convenient time

Summary

In this week, we learned about the Deadlock, the locking of a resource and the remedy to Deadlock, the concept of a Semaphore and the technique called spooling.

13.1 Network Architectures & 13.2 Network Protocols

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-12-27 4:19 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 13.1 Network Architectures 13.2 Network Protocols	Course: BSc Computer Science Class: How Computer Work [CM1030]-Lecture Date: December 27, 2019

Essential Question:

What are the different configurations of a network and the different protocols that govern its communication?

Questions/Cues:

- What is one way of classifying networks?
- What is LAN?
- What is WAN?
- What is PAN?
- What is a Cluster?
- What is a Star Network?
- What is a Bus Network?
- What is a Switch?
- What is Router?
- What is the Client-Server model of communication in terms of network classification?
- What is a Peer-to-peer Network?
- What is a Protocol?
- What is CSMA/CD?
- What is the Application layer?
- What is the Transport layer?
- What is the Network Layer?
- What is a Network Port?
- What is TCP?
- What is UDP?

Notes

- Aside from physical connection, networks can be classified by how large an area they cover
- LAN(Local Area Network) = connects comps that are close together or local to one another. ie. connected together in a single office or house.
- WAN (Wide Area Network) = connects comps over a much bigger or wider area than a LAN
- PAN (Personal Area Network) = mini wireless network that connects all devices on a person together. ie. mobile phone, laptop, smartwatch, wireless headphones, anything

a person carries around with them that requires an active internet connection.

- Cluster = Comps are connected together so they can work together for needs that need a lot of computing power. These comps are networked together and do their work in parallel, but to the outside world they act as if they were a single computer. This type of combined computer is called a cluster.
    - More reliable than a machine because if one comp fails then others can carry on without it.
- Star Network = One central comp/device that all other comps are connected to, but aren't connected to each other directly. This is called a Star network.
    - Star networks very widely used in early day of computing, but still used today for WiFi where all devices connect to a wireless access point.
    - Less popular in modern times for other network setups because it is very reliant on main comp and whole network fails if main comp fails.
- Bus Network = all comps connected to single long wire which is called bus. Comps in network are all independently connected to the bus and can communicate with all other comps on bus without relying on central comp. Bus is conceptually a long wire, but in fact might be quite small, in most cases bus will actually be a hub like a connective box
    - Ethernet, most popular wide network architecture is a bus network
- Small Local Area Networks are connected to other networks to form wider area networks. They're connected by little comps that forward packets from one network to the other.
- Switch = connects networks of same type and typically forwards packets unchanged
- Router = connect different types of networks Ethernet and WiFi and have to translate packets between different protocols
- Client-Server model = individual comps called clients, all communicate with central comp called server. Clients & servers not typically connected directly to each other. Client-server isn't about physical connections but about virtual connections using network protocols.
- Peer-to-peer network = individual people's comps communicate directly with each other
- Protocol = set of rules that comps use to comm over networks.
- CSMA/CD(Carrier Sense Multiple Access with Collision Detection) = when comp transmits, it also checks if there are other transmissions on bus, if there are, it stops transmitting. Each comp waits a random amount of time until re-transmission. Since the time was random, it is very unlikely that other comp will transmit at same time. CSMA/CD implemented as part of Ethernet standard.
- App layer = is the innermost layer of the layers surrounding packet transmission. App protocols are specific to a particular app or type of app. ie. HTTP protocols for web pages & web browsers
- Transport layer = handles comm between start point and destination of a comm. Manages what the comps at each end do to communicate without worrying either about the content of the message or all the networks & routers in between the two comps. ie. TCP & UDP.
- Network layer = handles how packets are transmitted across different networks from one comp to another. It handles how the message is passed between one network and the next via routers. Network layer protocol is almost always IP, the internet protocol.
- Link layer = handles transmission across a single physical network, for example the Ethernet or WiFi protocols
- Network port = number attached to a packet, which tells your network software where to send that packet. Port 80 is used for HTTP protocol, Port 110 is used by POP3 email protocol, & classic video game DOOM has it own Port 666.
- TCP(Transmission Control Protocol) = a reliable transport layer protocol, which means that messages get through. TCP starts by establishing a connection between the two comps. This means it checks that they're both ready and able to exchange data, it does this by sending some packets whose sole purpose is to make sure that both comps are ready. In TCP, the device that wants to start communicating first sends a packet called "syn" short for synchronize, which is a request for a connection. The other comp gets

that packet and replies with a "syn-ack" which means synchronized-acknowledge; this means it confirms that its received a syn & that it's ready to comm. Original device replies with an ack to acknowledge that it received the syn-ack & at this point they're both ready to talk.
    - The first device sends its first packet & once the device at other end has received it, it sends an ack to acknowledge that it has been received, this way your device can be sure that the comm worked. TCP gives numbers to packets to keep track of data over the internet. The acknowledge always includes the packet #, so sender knows exactly which packets have been received. Once all packets are sent & received, connection is closed same way it was started.
- UDP(User Datagram Protocol) = is an unreliable transport layer protocol. It sends packets without acknowledging them or re-sending them. Ideally for fast communication in real-time that wouldn't work with TCP because consistent acknowledges would slow this down.

Summary

In this week, we learned about the different configurations of networks, the layers of packet transmission and in detail the two different kinds of transport layer protocols, TCP & UDP.

13.1 Network Architectures & 13.2 Network Protocols-Reading

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2020-01-22 9:47 AM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 13.1 Network Architectures 13.2 Network Protocols	Course: BSc Computer Science Class: How Computer Work [CM1030]-Reading Date: December 28, 2019

Essential Question:

What are the different configurations of a network and the different protocols that govern its communication?

Questions/Cues:

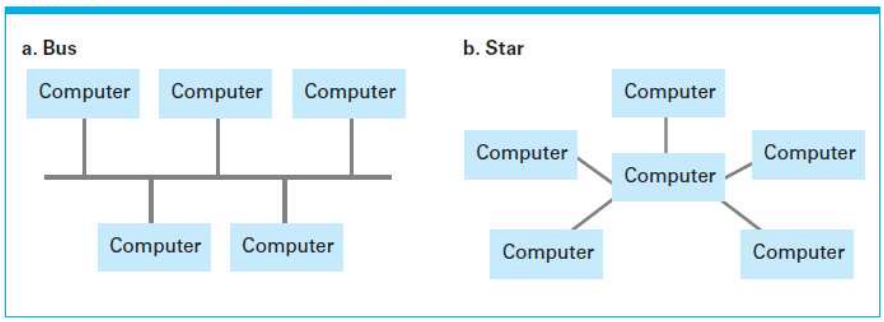
- What is PAN?
- What is LAN?
- What is MAN?
- What is WAN?
- What is a Open Network?
- What is a Closed Network?
- What is a Bus Network?
- What is a Star Network?
- What is a hub?
- What are Protocols?
- What is CSMA/CD?
- What is the Hidden Terminal Problem?
- What is a repeater?
- What is a bridge?
- What is a switch?
- What is a router?
- What is a forwarding table?
- What is a gateway?
- What is inter-process communication?
- What is the Client/Server Model?
- What is P2P?
- What are distributed systems?
- What is cluster computing?
- What is grid computing?
- What is cloud computing?
- What are tier-1 ISPs?
- What are tier-2 ISPs?
- What are tier-3 ISPs?
- What is a host?
- What is a hot spot?

- What is an IP Address?
- What is ICANN?
- What is a domain?
- What is a domain name?
- What are TLDs?
- What are Subdomains?
- What are Name Servers?
- What is DNS Lookup?
- What is VoIP?
- What is streaming?
- What is N-unicast?
- What is Multicast?
- What is on-demand streaming?
- What are CDNs?

Notes

- PAN(Personal Area Network) = used for short-range comms, typically less than a few meters such as between a wireless headset and a smartphone.
- LAN(Local Area Network) = consists of a collection of comps in a single building or building complex, for example the comps on uni campus
- MAN (Metropolitan Area Network) = network of intermediate size, such as one spanning a local community
- WAN(Wide Area Network) = links machines over a greater distance perhaps in neighboring cities or on opposite sides of the world
- Open Network = network's internal operation is based on designs that are in the public domain
- Closed Network (Proprietary Network) = network's internal operation is based on innovations owned & controlled by particular entity such as an individual or corporation; restricted by license fees & contract conditions
- Bus Network = machines are all connected to a common comm line called a bus; machines in network comm directly with each other over common bus. Sometimes, a bus network is created by running links from each comp to a central location where they are connected to a device called a hub.
- Star Network = in which one machine serves as a central focal point to which all the others are connected. Popular today in wireless networks where comm is conducted by means of radio broadcast and central machine called the access point(AP) serves as focal point around which all comm is coordinated; machines in network comm with each other indirectly through an intermediary central machine

Figure 4.1 Two popular network topologies



- Hub = little more than a very short bus. All it does is relay any signal it receives (with perhaps some amplification) back out to all the machines connected to it.
- Protocols = rules established by which activities are conducted to ensure network reliability
- CSMA/CD (Carrier Sense, Multiple Access with Collision Detection) = dictates that each message be broadcast to all machines on bus. Each machine monitors all the messages

but keeps only those addressed to itself. To transmit a message, a machine waits until the bus is silent, at this time it begins transmitting while continuing to monitor the bus. If another machine also begins transmitting, both machines detect the clash & pause for a brief, independently random period of time before trying to transmit again.
    - CSMA/CD is not compatible with wireless star networks in which all machines comm through a central AP, this is because a machine unable to detect that its transmissions are colliding with those of another.
- Hidden terminal problem = signals from different machines are blocked from each other by objs or distance even though they can all comm with central AP.
    - Result is that wireless networks adopt policy of trying to avoid collisions rather than trying to detect them, classified as CSMA/CA (Carrier Sense, Multiple Access with Collision Avoidance)
        - In CSMA/CA, when machine first needs to transmit and finds comm channel silent it doesn't transmit immediately. Instead it waits for a short period time and starts transmitting if channel has remained silent throughout that period
        - If busy channel is experienced during this process, machine waits for randomly determined period before trying again. Once period is exhausted, machine is allow to claim a silent channel without hesitation
    - To solve HTP some WiFi networks require that each machine send a short "request" message to AP & wait until AP acknowledges that request before transmitting an entire message. If AP is busy because it is dealing with "hidden terminal", it will ignore request & requesting machine will know to wait. Otherwise, AP will acknowledge request & machine will know it's safe to transmit. Note all machines in network will hear all acknowledgements sent from AP & thus have a good idea of whether AP is busy at any given time even though they may not be able to hear the transmissions taking place

Figure 4.2 Communication over a bus network

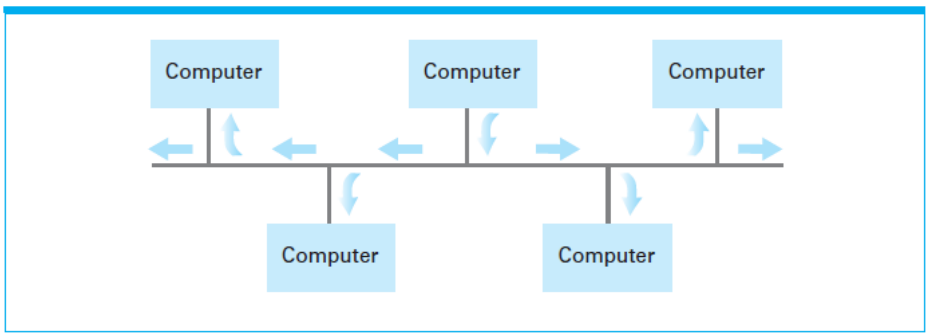
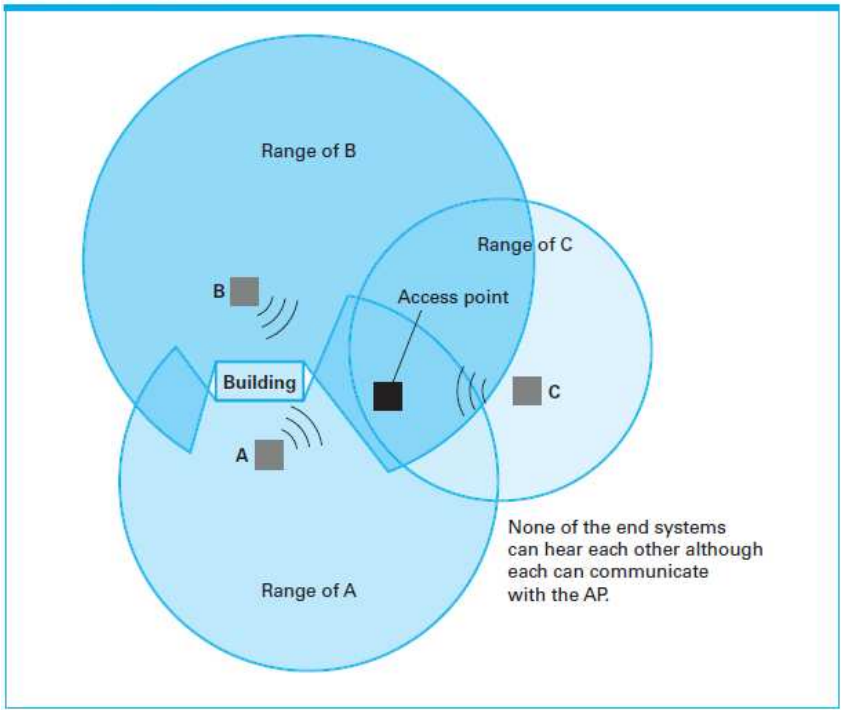
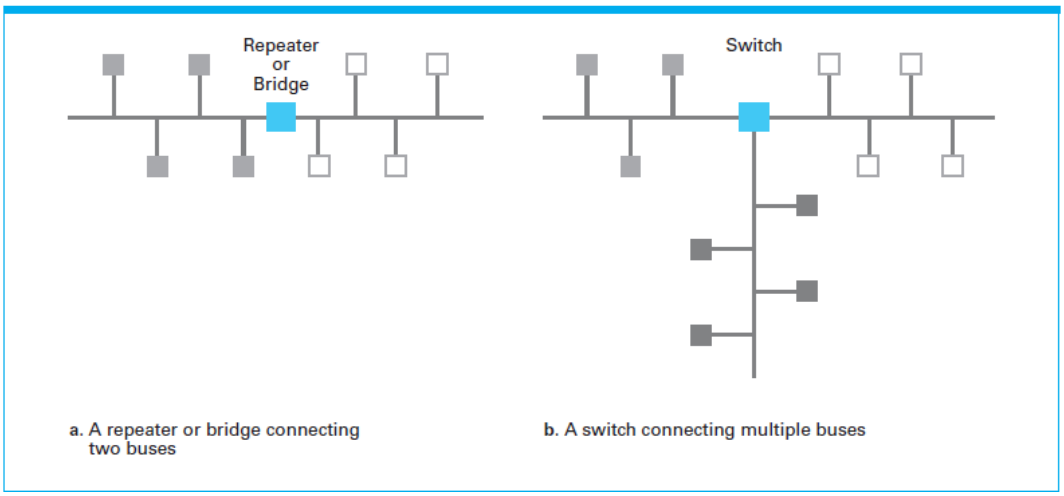


Figure 4.3 The hidden terminal problem



- Repeater = a device that passes signals back and forth between the original buses (usually with some form of amplification) without considering the meaning of the signals
- Bridge = Like a repeater, it connects two buses, but it doesn't necessarily pass all messages across the connection. Instead, it looks at the destination address that accompanies each message & forwards a message across the connection only when that message is destined for a comp on the other side. Thus, two machines on the same side of the bridge can exchange messages without interfering with comm taking place on the other side.
- Switch = is essentially a bridge with multiple connections, allowing it to connect several buses rather than just two. A switch produces a network consisting of several buses extending from the switch as spokes on a wheel. Like a bridge, a switch considers the destination addresses of all messages & forwards only those messages destined for other spokes. Moreover each message that is forwarded is relayed only into the appropriate spoke, thus minimizing the traffic in each spoke

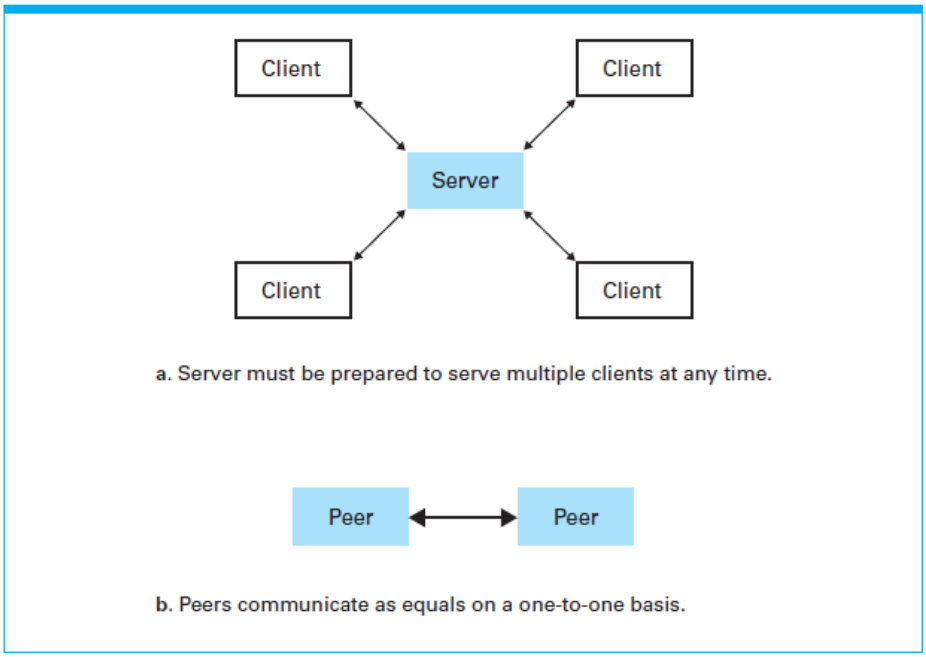
Figure 4.4 Building a large bus network from smaller ones



- Router = special purpose comps used for forwarding messages. Routers provide links between networks while allowing each network to maintain its unique internal characteristics

- Forwarding table = contains the router's knowledge about the direction in which messages should be sent depending on their destination addresses.
- Gateway = "point" at which one network is linked to an internet is often called a gateway because it serves as a passageway between network & outside world.
- Inter-process communication = various activities(or processes) executing on different comps with a network (or even executing on the same machine via time-sharing/multitasking), communicating with each other to coordinate their action & to perform their designated tasks.
- Client/Sever model = defines basic roles by processes as either a client--which makes requests of other processes or a server--which satisfies the requests made by clients
    - Early app of C/S model was in the office with a single printer acting as a print sever with all other machines acting clients making requests to the printer
    - Another app of C/S model was where one machine was equipped with high-capacity mass storage that contained all of an organization's records. Other machines requested access to records as needed. The machine with mass storage acted like the file server and other machines acted like clients that requested access to files stored at file server.
- Peer-to-peer (P2P) model = involves processes that provide service to & receive service from each other. P2P model usually involves process that execute on a temporary basis. Examples include instant message and playing competitive interactive games.
    - Swarm distribution is where one peer may receive a file from another and then provide that file to other peers.

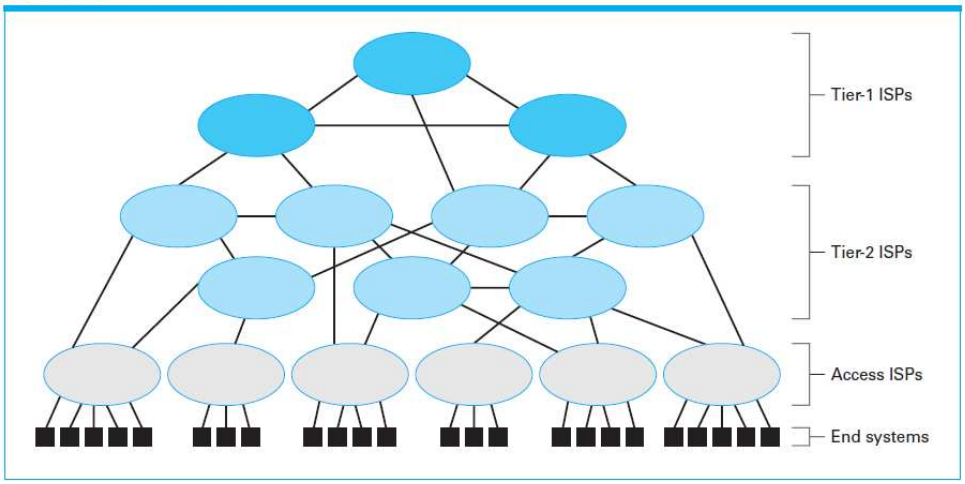
Figure 4.6 The client/server model compared to the peer-to-peer model



- Distributed systems = consist of software units that execute as processes on different computers.
- Cluster computing = a distributed system in which many independent comps work closely together to provide computation or services comparable to much larger machine
    - Cost of individual machines + high-speed network to connect them is cheaper than higher-priced supercomputer, higher reliability & lower maintenance costs
    - Such systems used to provide high-availability--at least one member of cluster will be able to answer request, even if other cluster members break down or are unavailable. Secondly, such systems also provide load-balancing--workload can be shifted automatically from members of cluster that have too much to do to those that may have too little.

- Grid computing = distributed systems that are more loosely coupled than clusters but still work together to accomplish large tasks, can involve specialized software to make it easier to distribute data & algorithms to machines participating in a grid
- Cloud computing = huge pools of shared comps on the network can allocated for use by clients as needed.
- Tier-1 ISPs = very high-speed, high capacity, international WANs. These Networks thought of as backbone of the Internet. Typically operated by large companies that are comms business.
- Tier-2 ISPs = tend to be more regional in scope and less potent in their capabilities. Typically also operated by large companies that are comms business.
    - Tier-1 & tier-2 ISPs are essentially networks of routers that collectively provide the Internet's comm infrastructure; they are core of the Internet.
- Tier-3 ISPs (Access ISPs) = Intermediary that provides access to core of Internet. Access ISP is an independent internet, also called an intranet, operated by single authority that is in business of supplying Internet access to individual homes & businesses

Figure 4.7 Internet composition



- End System or Hosts = devices that individual users connect to the access ISPs
- Hot spot = Area within the AP or group of AP's range
- IP Address = unique identifying address to a comp on the Internet. IP short for Internet Protocol. Originally, each IP address was a pattern of 32 bits
    - IP addresses traditionally written in dotted decimal notation in which bytes of the address are separated by periods & each byte is expressed as an integer rep'ed in traditional base 10 notation
- ICANN(Internet Corporation for Assigned Names and Numbers) = corp established to coordinate Internet's operation
- Domain = a "region" of the Internet operated by a single authority such as a university, club, company, or government agency.
    - Each domain must be registered with ICANN, a process handled by companies called registrars that have assigned this role by ICANN
- Domain Name = a mnemonic DN which is unique among all domain names throughout the Internet. Domain name are often descriptive of the organization registering the domain.
- Top-level domains (TLD) = For example, the edu suffix for "educational" is considered a TLD. Others are com for commercial institutions, gov for US government, org for nonprofit & info for unrestricted use.
- Country-code TLD = two-letter TLD for specific countries such as ca for Canada.
- Sub-domain = multiple extensions are used as a means of organizing the names within a domain. SDs often rep different networks within the domain's jurisdiction
- Name servers = essentially directories that provide address translation services to clients. Collectively, name servers are used as an Internet-wide directory system known as the domain name system (DNS).

- DNS Lookup = The process of using DNS to perform a translation is called a DNS lookup
    - for a machine to be accessible by means of a mnemonic domain name, that name must be rep'ed in a name server within the DNS
- Voice over Internet Protocol(VoIP) = Internet infrastructure is used to provide voice comm similar to that of traditional telephone systems.
- Streaming = transporting audio & video across the Internet in real-time
- N-unicast = single sender involved with multiple unicasts. For example, establishing a server that would send program messages to each of the clients who requested them in case of a Internet radio station.
    - Unicast refers to one sender sending messages to one receiver
- Multicast = server transmits a message to multiple clients by means of a single address and relies on the routers in the Internet to recognize significance of that address & to produce & forward copies of the message to the appropriate destinations
- On-demand streaming = end user expects to view or listen to media at an arbitrary time of their choosing
- Content delivery networks (CDNs) = groups of servers distributed strategically around the Internet that specialize in streaming copies of content to nearby end users in their network "neighborhood"

Summary

In this week, we learned about the different configurations of networks, different types of distributed computing, different tiers of ISPs, the DNS & domain names, and finally what streaming is & various technologies involved in making it possible.

14.1 Network Security

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2020-01-13 2:24 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 14.1 Network Security	Course: BSc Computer Science Class: How Computer Work [CM1030]-Lecture Date: January 13, 2020

Essential Question:

What are the various ways in which the security of a network can be compromised?

Questions/Cues:

- What is HTTPS?
- What are external attacks?
- What is a distributed denial of service attack?
- What are botnets?
- What is a firewall?

Notes

- HTTPS = encrypted version of HTTP protocol, it's an app layer protocol.
- External attacks = comps that attempt to disable the whole network or comps on it
- Hack = trying to get access to another machine without being allowed to so
    - can be avoided by using authentication; using a login with secure password
- DDos (Distributed denial of service) attack = involves alot of comps simultaneously sending huge amts of network packets to specific machine or to lots of machines on network. This sends so many packets that network or comps completely overload & cannot handle its normal business.
    - DDos typically carried out using botnets.
    - DDos diffcult to defend against because it's done using valid network packets.
- Botnets = network of comps that have been infected by virus which makes them perform DDos.
- Firewall = filter that prevents certain types of packets from coming in & out of a machine or network
    - can often block legit traffic
    - often implemented together with proxy servers
- Blacklisting = rejecting IP addresses than can be known to be malicious
- Whitelisting = only allowing packets that come from list of approved good IP addresses
- Proxy server = comp that acts as an intermediary between a client & a server or between client & comps on network. All traffic must go through proxy server which forwards packs to appropriate comps. This means no external comps can interact directly with comps in network & all traffic can be filtered with firewall

Summary

In this week, we learned the dangers to network security & remedies to a few such attacks.

14.1 Network Security-Reading

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2020-01-13 6:25 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 14.1 Network Security-Reading	Course: BSc Computer Science Class: How Computer Work [CM1030]-Reading Date: January 13, 2020

Essential Question:

What are the various ways in which the security of a network can be compromised?

Questions/Cues:

- What is a virus?
- What is a worm?
- What is a Trojan horse?
- What is Spyware?
- What is phishing?
- What is DoS attack?
- What is a firewall?
- What is Spoofing?
- What is a spam filter?
- What is a proxy server?
- What is anti-virus software?
- What is SSL?
- What is public-key encryption?
- Who are the certificate authorities and what are certificates?
- What is authentication?

Notes

- Virus = software that infects comp by inserting itself into progs that already reside in machine. When "host" prog is executed, virus is executed. When executed, virus try to copy itself to other progs. Some viruses perform actions like degrading portions of OS, erasing large blocks or otherwise corrupting data & other progs
- Worm = autonomous prog that transfers itself through a network taking up residence in comp & forwarding copies of itself to other comps. Like a virus, worm can copy itself or perform extreme actions. Consequence of worm is an explosion of the worm's replicated copies that degrade performance of legit apps & can lead to overload of entire network or internet
- Trojan horse = enters comp disguised as desirable prog, imported by user. Once in comp, TH performs additional activities harmful with effects; additional activities start immediately. In other cases, TH may lie dormant until triggered by specific event such as a date.

- Spyware = sniffing software that collects info about activities at comp on it resides & reports that info back to designer
- Phishing = technique of obtaining info explicitly by simply asking for it. Process involves casting numerous "lines" in hopes someone will "take the bait"
- DoS (Denial of Service) attack = process of overloading a comp with msgs. To accomplish this, attacker usually plants software on numerous unsuspecting comps that will generate msgs when a signal is given(called botnet). When signal given, botnet swamp target with msgs
- Firewall = filtering traffic passing through a point in network.
- Spoofing = Masquerading as a party other than one's self
- Spam filters = firewalls designed to block unwanted email
- Proxy server = software unit that acts as an intermediary between client & server with goal of shielding client from adverse actions of server. PS filters all msgs from server to client.
- Anti-virus software = used to detect & remove presence of known viruses & other infections
- SSL(Secure Sockets Layer) = provide secure comm links between web clients & servers
- Public-key encryption = involves techniques by which encryption systems are designed so that having knowledge about how msgs are encrypted does not allow one to decrypt msgs. PKE system involves use of values called keys, one called public key (used to encrypt msgs), the other key known as private key (used to decrypt msgs)
- Certificate authorities = whose task is to maintain accurate lists of parties & their public keys. These authorities then acting as severs, provide reliable public-key info to their clients in packages known as certificates.
- Certificate = package containing a party's name and that party's public key
- Authentication = making sure the author of msg is in fact, the party it claims to be.
- Digital Signature = a bit pattern produced by the holder a private key that encrypts their msgs. A pattern that only they can produce. A digital signature can be as simple as the encrypted version of the msg itself.

Summary

In this week, we learned the dangers to network security & remedies to a few such attacks.

15.1 The Internet & 15.2 The web

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2020-01-18 3:28 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 15.1 The internet & 15.2 The web	Course: BSc Computer Science Class: How Computer Work [CM1030]-Lecture Date: January 16, 2020

Essential Question:

What are the various technologies/protocols that make up the World Wide Web and its Websites?

Questions/Cues:

- What is SMTP?
- What is VoIP?
- What is the WWW?
- What is a Domain Name?
- What is DNS?
- What is a Cluster?
- What is a Virtual Server?
- What are Virtual Machines?
- What are Containers?
- What is LAMP?
- What is MEAN?

Notes

RECAP: The link, network & transport layers are all about getting packets from one end to another. The application layer on the other hand, is more about the content of the communication, what data is sent and how that data is used by your applications.

- SMTP (Simple Mail Transfer Protocol) = When email servers send emails to each. This protocol used for comms between servers. Also used to send email from your comp to server
    - Comps & phones use different protocols to fetch emails from server, POP/POP3 designed to downloading email to your comp & storing it there. IMAP stores email on the sever, but lets you access it on your phone or comp.
        - Microsoft has own email protocol called Exchange
    - Nowadays, we use webmail- which is email access to on a webpage, not an app on your comp, protocol used here is HTTP for viewing your email, not downloading them

- VoIP (Internet Telephony) = telephone convo by sending sound of your voices over the Internet rather than traditional phone lines
    - phone convos nowadays are all sent using IP. In VoIP, when making your comp sends request to VoIP server for connection, once connection is established, peer-to-peer comms begins
- The World Wide Web(WWW) is a client/server protocol, involves client sending a request for webpage using an HTTP request to server, which creates page & sends it back.
    - We humans use URLs (Uniform Resource Locators) to locate and identify webpages, but internet protocols don't understand URLs, they use numerical addresses called IP addresses
    - The middle part of URL ie. **www.gold.ac.uk** is address of a web server, which the comp needs to contact the server.
- Domain Name = The readable part of the address
    - This is easy for us to understand, but not network protocols
- Domain Name Server (DNS) = converts Domain names in IP address. An DNS server knows how to convert domain names into IP addresses.
    - To find out IP address of the domain you want to use, your client sends a request to DNS server, which sends back an IP address, which your comp can use an HTTP request to the server
    - Most apps will DNS first to get a hold of an IP address before starting main comms
        - This means it doesn't need to use the full reliable channels setup by TCP. It only involves two msgs & if your client doesn't get a reply it know something is wrong. This is why it's an example of an app protocol that uses UDP in the transport layer
- Modern websites use a combination of a database, assets(audio, video, images), & a template to generate the page to your browser when requested.
    - Even more modern websites don't even use the file system anymore or even include a file in the URL. They use a piece of software called a routing engine to interpret URL & work out what web page to generate. All elements of URL after domain name essentially become arguments to a single routing scripts. Routing on a web server is working out which bit of the server program to use to generate a web page
- Any client-side interaction is done in your web browser & is implemented in a lang called JavaScript
- Templating, previously discussed as server-side activity, but is now client-side. Site template are sent directly to client, data is fetched from database separately & sent to client independently from the template. It is client that combines the two.
    - Advantage of this is, server doesn't have send whole webpages every time changes, just sends little bit of data and client updates the user's visual.
    - Also output doesn't have to be HTML and the client viewing doesn't have be web browsers. Mobile apps use a similar framework, in that they also receive webpages, but use it in a different way to display on a phone screen.
- Cluster = multiple comps networked to perform single function
- Virtual servers = server that are implemented in software, so that several virtual servers can run on a single comp
- Virtual machine = one comp implements machine lang of another comp in software, so each virtual machine seems like its own independent comp with its own OS & app software, but it's actually implemented in software in another comp.
- Containers = like a virtual machine except all of the containers share the same OS kernel, but otherwise act like different comps with different app software.
- Most websites are now implemented on cloud servers. These cloud server use both clusters & virtual servers. They consist of huge clusters of comps, and each website has one or more virtual servers implemented not on in a single comp but on a cluster.
- LAMP = is a "stack", collection of software that run websites: L for Linux (OS), Apache (Web Server), MySQL (Database management system) & P for PHP, Python, or in the early days Perl. In the early days, LAMP wasn't distributed as a single unit.

- MEAN= also a stack: M for MongoDB (different database management system), E for Express (web app framework that does a number of things, but particularly routing), A for Angular (HTML generation & templating engine) & N for Node (web server framework). MEAN doesn't include OS in its stack, Node has some features of Apache but all of them so its used in conjunction with other web server software. PHP is prog lang, but none of the MEAN apps are a lang; Node, Angular & Express are all based on JavaScript, a lang that originated as a client-side scripting lang, but Node make it possible to run it on the server.
    - Angular & Express are frameworks that do many of the things that would otherwise have to written by hand, by a PHP programmer, so they make it easier to write complex websites

Summary

In this week, we learned about the Internet app protocols like SMTP, IMAP, VoIP. Alongside this, we explored the make of website technology with Domain names, DNS, Databases, & Templates. Finally, we looked at the various ways a web server could and is implemented ie. clusters or containers.

15.1 The Internet & 15.2 The web-Reading

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2020-01-17 6:13 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 15.1 The internet & 15.2 The web-Reading	Course: BSc Computer Science Class: How Computer Work [CM1030]-Lecture Date: January 16, 2020

Essential Question:

What are the various technologies/protocols that make up the World Wide Web and its Websites?

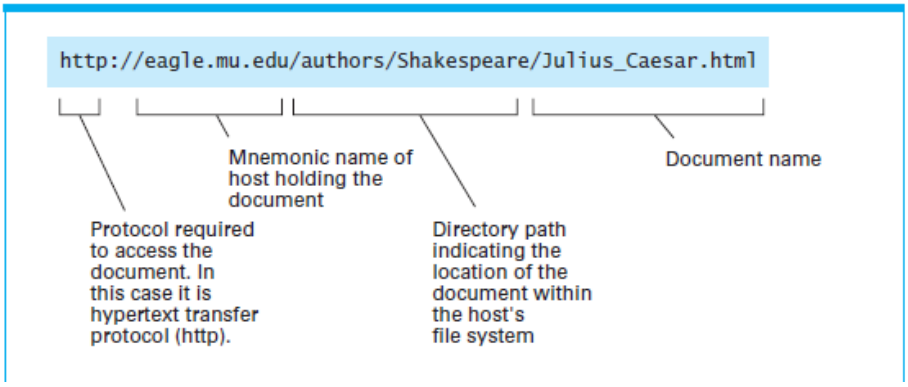
Questions/Cues:

- What is Hypertext?
- What is a Browser?
- What are web servers?
- What is a URL?
- What is HTML?
- What is XML?
- What are Search Engines?
- What is the Application Layer?
- What is the Transport Layer?
- What is the Network Layer?
- What is the Link Layer?
- What is TCP?
- What is UDP?
- What is Hop count?

Notes

- Hypertext = linked documents
- World Wide Web = hypertext document format for embedding hyperlinks to other docs; a protocol for transferring hypertext across the network and a server process that supplied hypertext pages upon request in the early days
- Browser = is on user's comp and is charged with the tasks of obtaining materials requested by user & presenting these materials to the user in organized manner
- Web servers = reside on on comp containing hypertext docs to be accessed. Its task is to provide access to docs under its control as requested by clients (browsers). Hypertext docs transferred between web servers and browsers using protocol known as Hypertext Transfer Protocol (HTTP)
- URL(Uniform Resource Locator) = A unique address, Each URL contains the info needed by browser to contact the proper server & request the desired doc

Figure 4.8 A typical URL



- Hypertext Markup Language (HTML) = system of tags, that describe how the doc should appear on a display screen, what multimedia resources (such as images) should accompany the doc & which items within the doc are linked to other docs. Tags are denoted by "<" & ">"

Figure 4.9 A simple webpage

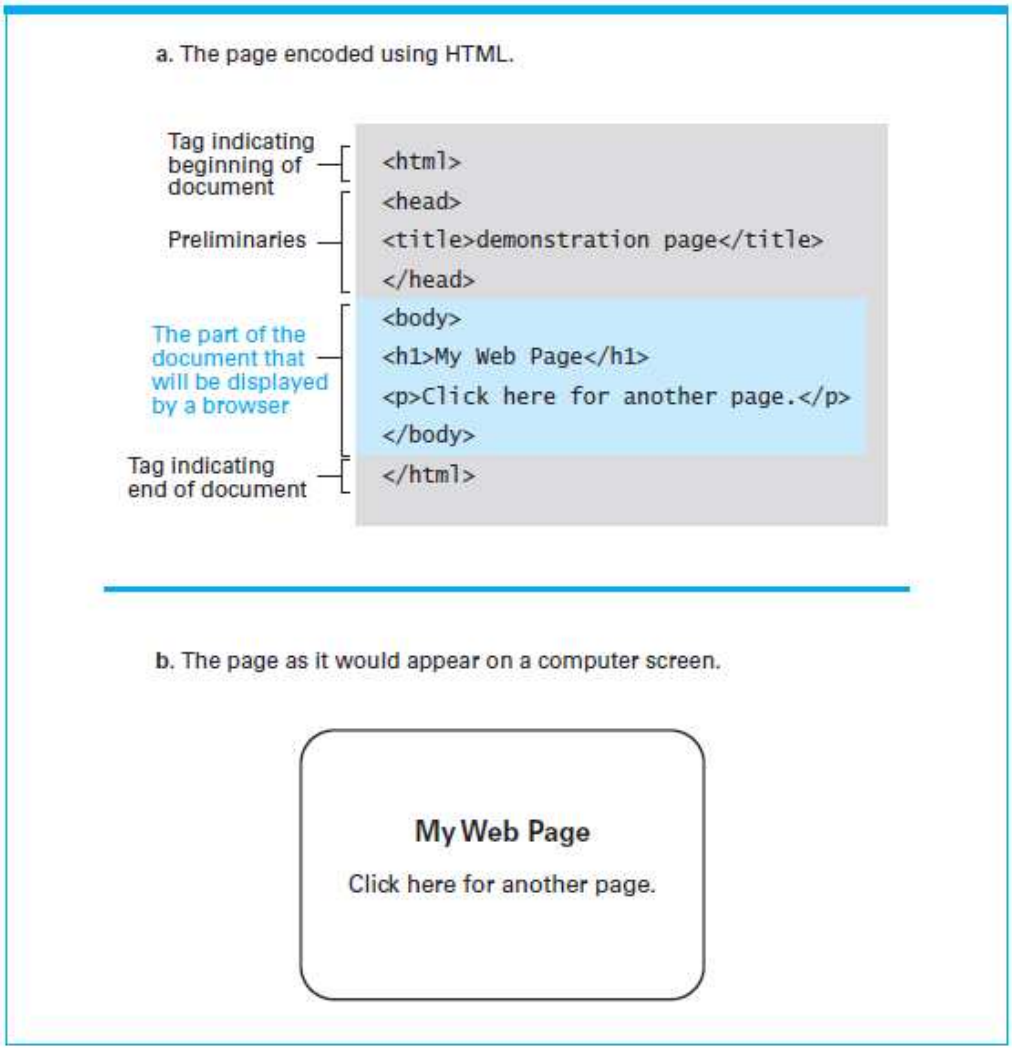


Figure 4.10 An enhanced simple webpage

a. The page encoded using HTML.

```
<html>
<head>
<title>demonstration page</title>
</head>
<body>
<h1>My Web Page</h1>
<p>Click
    <a href="http://crafty.com/demo.html">
    here
    </a>
    for another page.</p>
</body>
</html>
```

Anchor tag containing parameter

Closing anchor tag

b. The page as it would appear on a computer screen.

My Web Page

Click here for another page.

- eXtensible Markup Language (XML) = standardized style for designing notational systems for rep'ing data as text files. Following XML, notational systems call markup langs have been developed for reping math, multimedia presentations and music. In fact, HTML is markup lang based on XML standard that was developed for reping webpages (Original HTML was developed before solidification of XML, so some features of HTML don't conform to XML standard). XML allows development of new markup langs that emphasize semantics rather than appearance like HTML
- Search engines = websites that assist user in locating Web material pertaining to a subject of interest
- Application layer = consists of software units such clients and servers that use Internet comm to carry out their task, not restricted to the software in the app but includes many utility packages (file transfer using FTP or remote login using SSH). App layer uses transport lay to send & receive msgs over Internet, it's the app layer's responsibility to provide an address that is compatible with the Internet infrastructure. To fulfill this, app layer may use the services of name servers within the Internet to translate mnemonic addresses used by humans into Internet-compatible IP addresses
- Transport Layer = accepts msgs from app layer & ensures that msgs are properly formatted for transmission over Internet. It divides long msgs into small segments, which are transmitted over Internet as individual units. Then it adds sequence nums to the small segments it produces so that segments can be reassembled at msg's destination. Then it hands these segments known as packets over to the network layer.

- Network Layer = job to decide in which direction a packet should be sent at each step along the packet's path through Internet. It's a combination of the network & link layer that make up the software on Internet routers. Network layer is in charge of maintaining router's forwarding table & uses this table to determine direction in which to forward packets.
- Link Layer = at router level is in charge of receiving & transmitting packets. Link Layer has responsibilty of transferring packet, so it must deal with comm details particular to individual network in which comp resides (i.e CSMA/CD for Ethernet & CSMA/CA for WiFi).

Figure 4.13 The Internet software layers

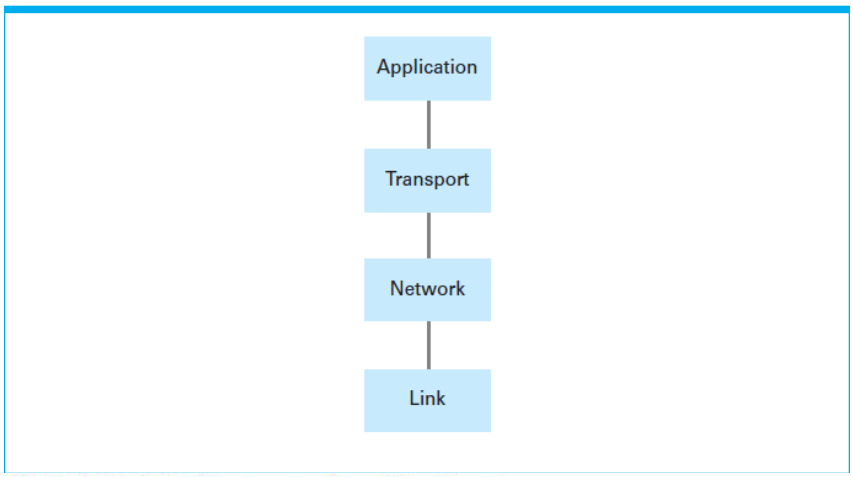
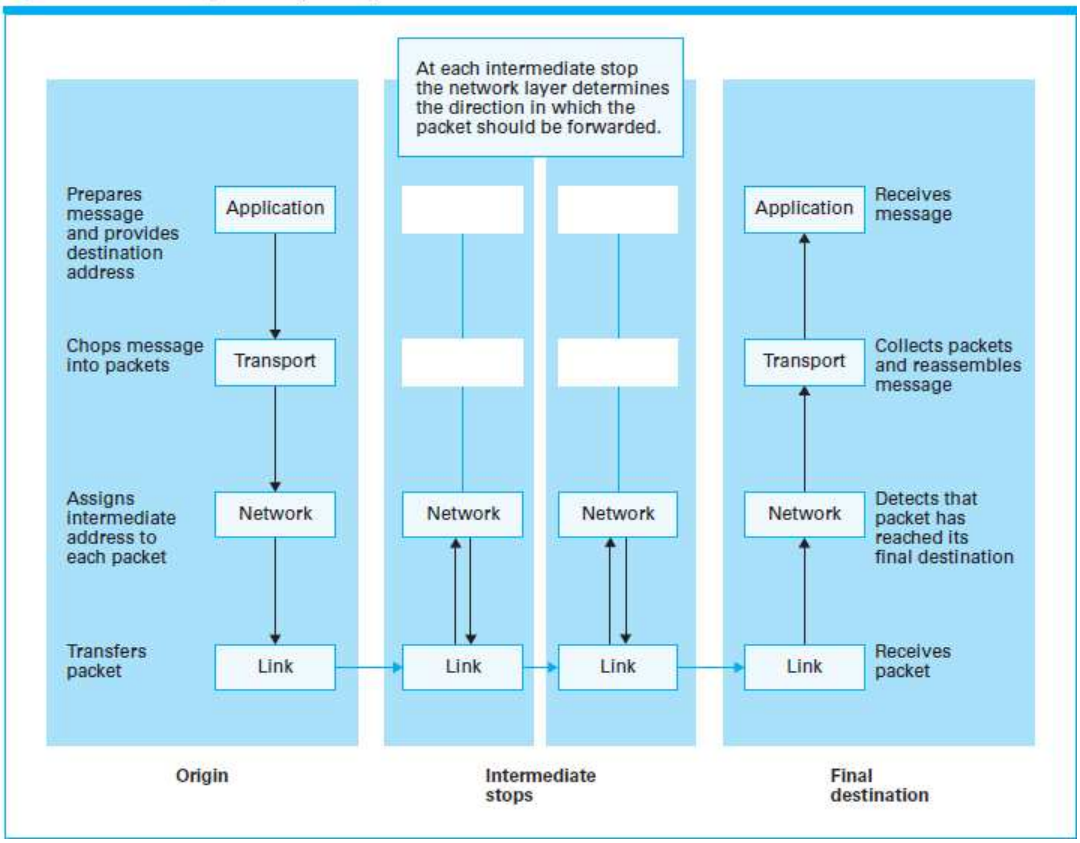


Figure 4.14 Following a message through the Internet



- Transmission Control Protocol (TCP) = transport layer protocol, before sending msg it send an "acknowledge" msg to establish a connection with receiving end.
    - TCP at both ends works together by means of acknowledgements & packet retransmissions to assure all segments of msg are successfully transferred to destination. TCP is called a reliable protocol
    - TCP offers flow control, meaning TCP at origin can reduce the rate of segment transmission from overwhelming its end receiver & congestion control also at

origin to adjust rate to alleviate congestion between nodes

- User Datagram Protocol (UDP) = transport layer protocol that doesn't establish a connection with receiving end (connectionless). UDP doesn't offer packet retransmission, so it's called unreliable protocol.
- Hop count = time to live for a packet, limit of the # of times a packet can be forwarded, each time IP network layer forwards a packet it decrements it by one.

Summary

In this week, we learned about the Internet app protocols like HTTP, what a browser is, what a web server, & the various Layer of involved in Internet Transmission. Alongside this, we explored, web markup languages like HTML & XML, the difference TCP & UDP and the hop count of a packet.

16.1 Databases

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2020-01-24 1:52 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 16.1 Databases	Course: BSc Computer Science Class: How Computer Work [CM1030]-Lecture Date: January 16, 2020

Essential Question:

What is a database?

Questions/Cues:

- What is a database?
- What is a database management system?
- What is database model?
- What is a relational database?
- What is the name of the operation to get data from a particular tuple?
- What is the name of the operation that accesses only some attributes of a table?
- What is the join operation?

Notes

- Database = collection of data stored in a structured way
- Database management system (DMS) = software which controls how data is stored and accessed by other apps, provides abstraction of data to apps. In production system, database has its own database server
- Database model = how the data is represented and accessed
- Relational Database = stores all its data in tables like a spreadsheet, each row of the data table is a different entry called a Tuple, each column is a particular aspect of that entry called an attribute. Finally, every row of the table has exactly the same elements in it to make the data easy to handle. Each cell contain data itself, which is normally text or numbers
    - More complex data like photo typically to big to store in database, so they stored in files and file names are stored in database
    - Most database systems add an extra attribute call an "id", typically a unique number. We can use this ID to identify an entry from one table to another
    - Relational DMS have a range of functions to access data in a database
- Select = operation to get data from a particular tuple
    - Select always doesn't return one result
- Project = operation to access only some of the attributes of a table
- Join = operation that takes data from one table and joins it with data in another table to create a new combines table

- Many relational databases use a mini-programming lang called SQL (Structured Query Language) to access these commands/operations

Summary

In this week, we learned about what database is and the range of functions used to access data from a database.

16.1 Databases-Reading

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2020-01-25 1:36 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 16.1 Databases-Reading	Course: BSc Computer Science Class: How Computer Work [CM1030]-Reading Date: January 24, 2020

Essential Question:

What is a database?

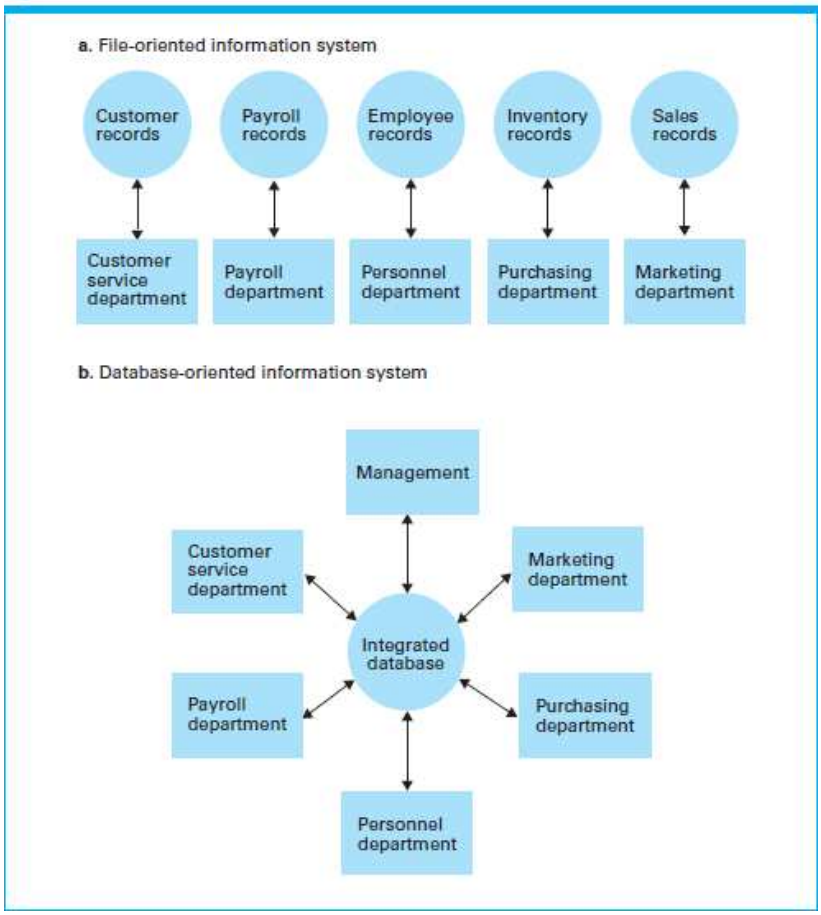
Questions/Cues:

- What is a database?
- What is a flat file?
- What is a schema
- What is a subschema?
- What is a database management system?
- What is database model?
- What is a relational database?
- What is lossless/Nonloss decomposition?
- What are the Select, Project and Join Operations in a relational database?

Notes

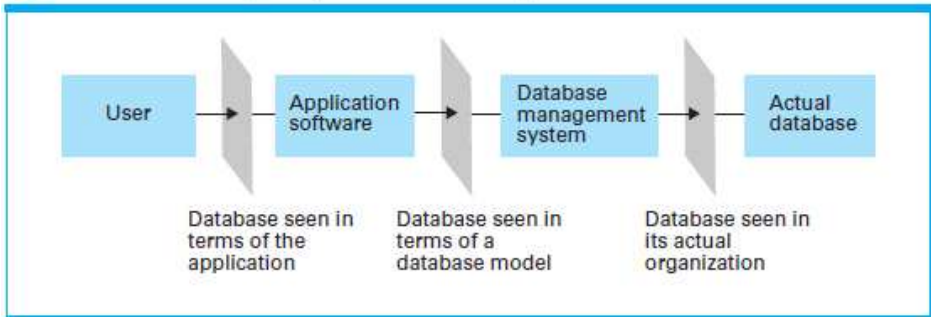
- Database = collection of data that is multidimensional, internal links between its entries make the info accessible form a variety of perspectives
- Flat File = one-dimensional storage system, meaning that it presents its info from a single point of view

Figure 9.1 A file versus a database organization



- Schema = description of the entire database structure that is used by the database software to maintain the database
- subschema = description of only that portion of the database pertinent to a particular user's need

Figure 9.2 The conceptual layers of a database implementation



- Database management system (DBMS) = the one that actually performs the manipulation of database, if user requests to add/delete info; it alters the database. If the users requests to retrieve info, it is the DBMS that is the one that performs the required searches
- Data independence = the ability to change the organization of the database itself without changing the app software
- Database model = conceptual view of the database
- Relational Database = collection of tables consisting of rows and columns, data stored in rectangular tables called relation, similar to spreadsheets
    - A row in a relation is called a tuple, and columns in a relation are referred to as attributes because each entry in a column describes some characteristic, or attribute of the entry rep'ed by the corresponding tuple

Figure 9.3 A relation containing employee information

Empl Id	Name	Address	SSN
25X15	Joe E. Baker	33 Nowhere St.	111223333
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555
•	•	•	•
•	•	•	•
•	•	•	•

Figure 9.4 A relation containing redundancy

Empl Id	Name	Address	SSN	Job Id	Job Title	Skill Code	Dept	Start Date	Term Date
25X15	Joe E. Baker	33 Nowhere St.	111223333	F5	Floor manager	FM3	Sales	9-1-2009	9-30-2010
25X15	Joe E. Baker	33 Nowhere St.	111223333	D7	Dept. head	K2	Sales	10-1-2010	*
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999	F5	Floor manager	FM3	Sales	10-1-2009	*
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555	S25X	Secretary	T5	Personnel	3-1-1999	4-30-2010
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555	S26Z	Secretary	T6	Accounting	5-1-2010	*
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•

Figure 9.5 An employee database consisting of three relations

EMPLOYEE relation

Empl Id	Name	Address	SSN
25X15	Joe E. Baker	33 Nowhere St.	111223333
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555

JOB relation

Job Id	Job Title	Skill Code	Dept
S25X	Secretary	T5	Personnel
S26Z	Secretary	T6	Accounting
F5	Floor manager	FM3	Sales
•	•	•	•
•	•	•	•
•	•	•	•

ASSIGNMENT relation

Empl Id	Job Id	Start Date	Term Date
23Y34	S25X	3-1-1999	4-30-2010
34Y70	F5	10-1-2009	*
23Y34	S26Z	5-1-2010	*
•	•	•	•
•	•	•	•
•	•	•	•

Figure 9.6 Finding the departments in which employee 23Y34 has worked

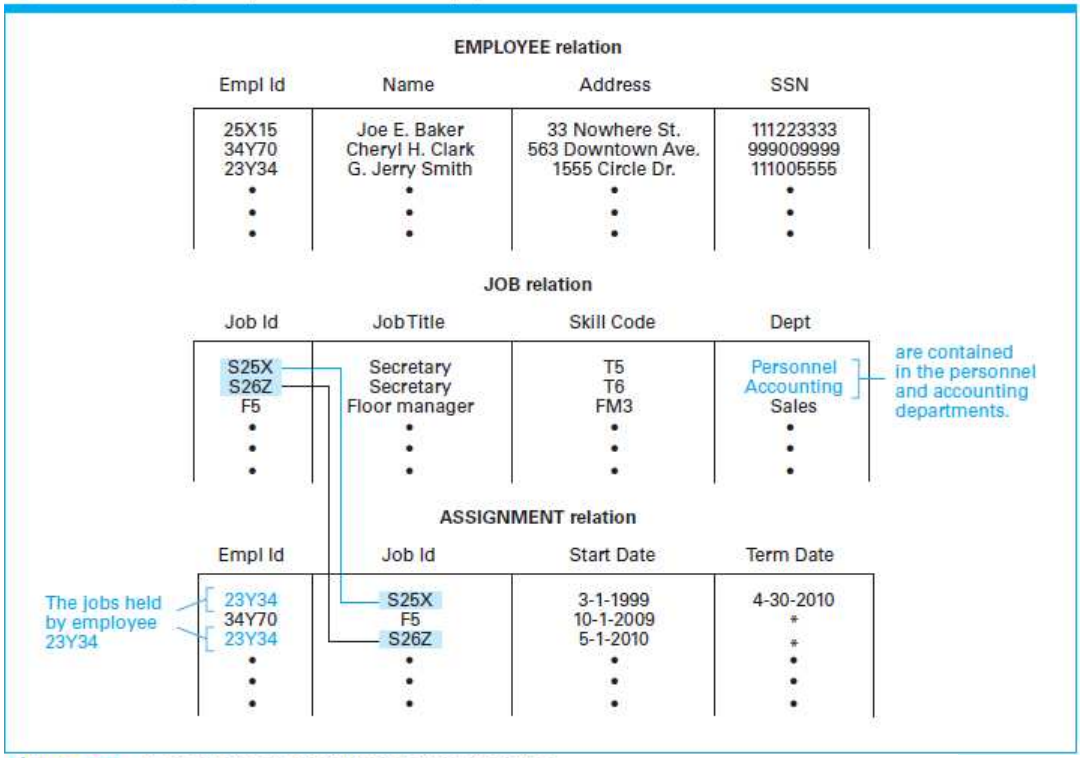
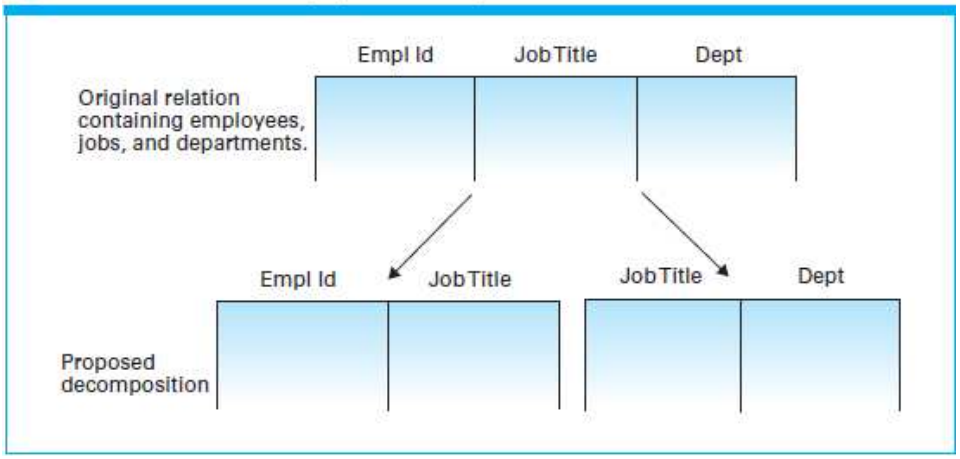


Figure 9.7 A relation and a proposed decomposition



- Lossless decomposition = loss of info when dividing a relation into smaller relation
- Nonloss decomposition = No loss of info when dividing a relation into smaller relations
- Select Operation = taking tuples possessing certain characteristics and to place these selected tuples in a new relation, extracting roes from a relation
- Project Operation = extracts columns form a relation
- Join Operation = used to combine different relations into one relation

Figure 9.8 The SELECT operation

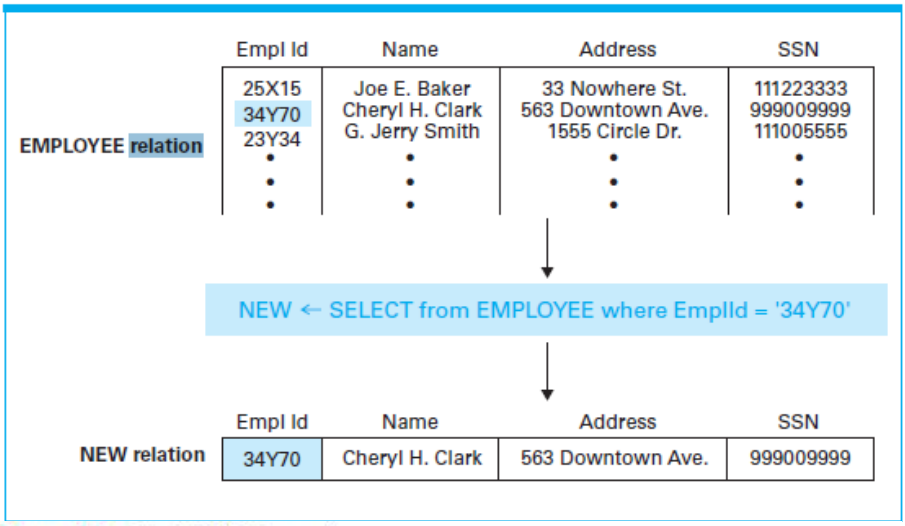


Figure 9.9 The PROJECT operation

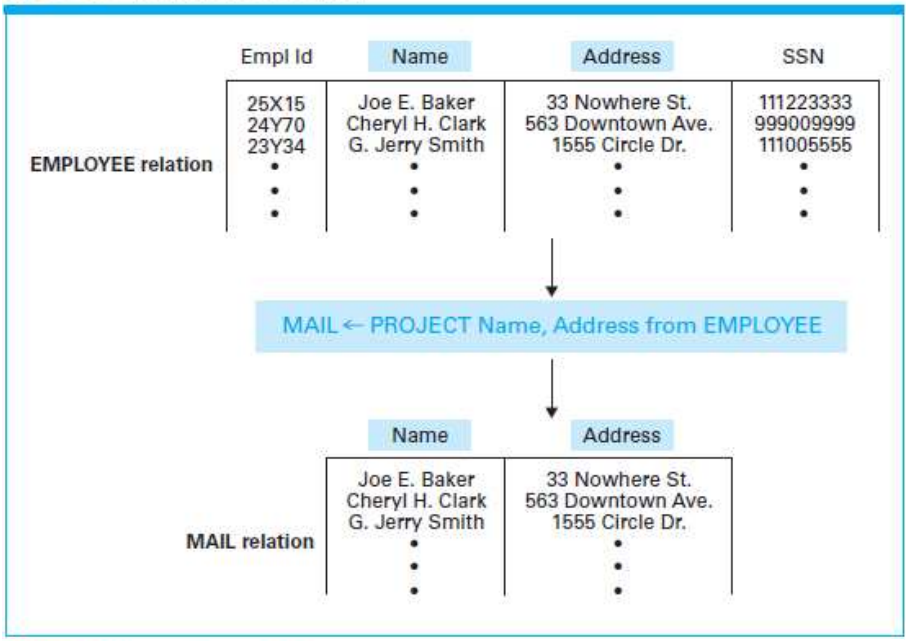


Figure 9.10 The JOIN operation

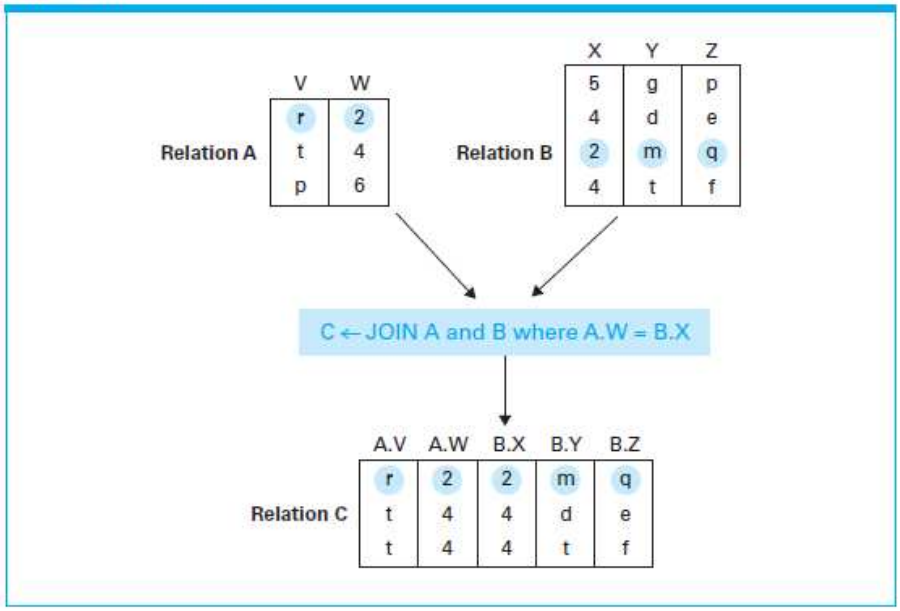


Figure 9.11 Another example of the JOIN operation

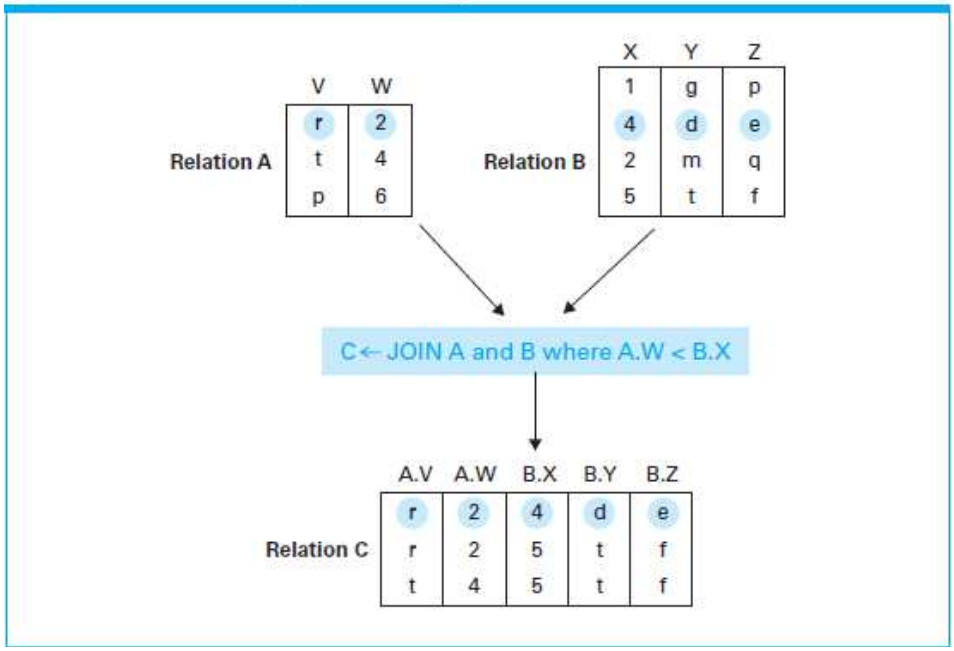
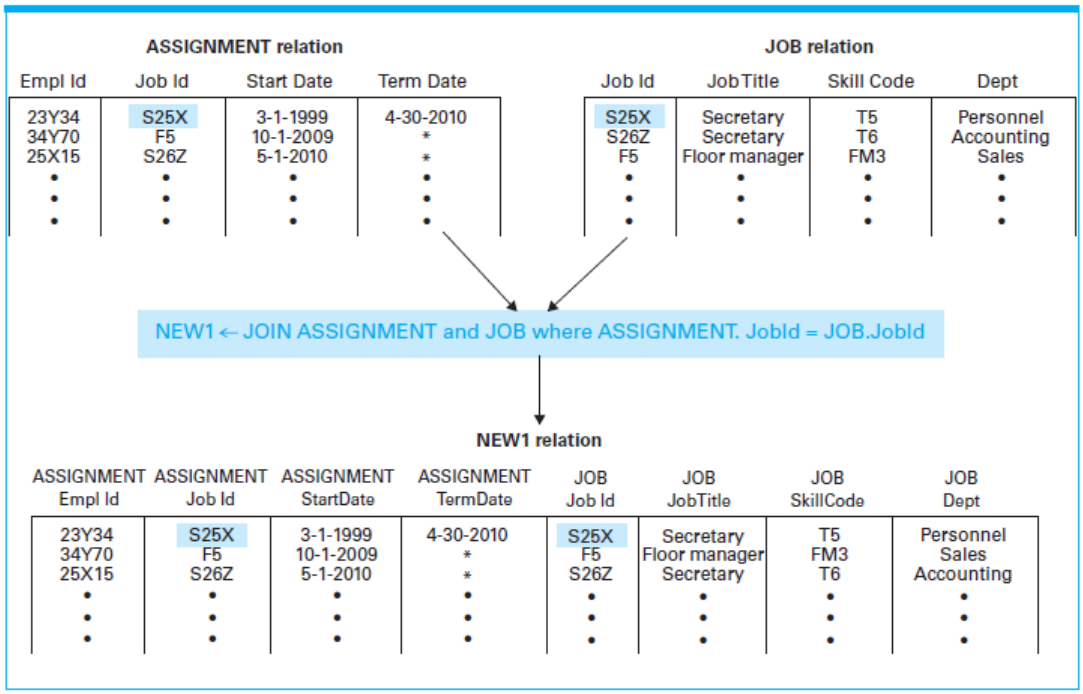


Figure 9.12 An application of the JOIN operation



Summary

In this week, we learned about what database is and the range of functions used to access data from a database.

17.1 AI & 17.2 ML

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2020-01-27 5:55 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 17.1 Artificial Intelligence 17.2 Machine Learning	Course: BSc Computer Science Class: How Computer Work [CM1030]-Lecture Date: January 27, 2020

Essential Question:

What is AI and the branch within AI called Machine Learning?

Questions/Cues:

- What is Artificial Intelligence?
- What is Computer Vision?
- What is Natural Language Processing?
- What is Computational Creativity?
- What is Narrow AI?
- How did AI work in the early days?
- What is Machine Learning?
- What is Regression?
- What is a Generative Model?
- What is Supervised Learning?
- What is Unsupervised Learning?
- What is Reinforcement Learning?
- What is the Nearest Neighbor Algorithm in ML?
- What are Support Vector Machines?
- What is Optimization?
- What are Decision Trees?
- What are Neural Networks?

Notes

- Artificial Intelligence (AI) = area of Comp. Sci that tries to make machines that can replicate human intelligence to do tasks that humans can do but have been historically very difficult for machines
    - AI is split in many subareas to address particular challenges that arises in development of AI like Robotics
- Computer Vision = comps that see and understand
- Natural language processing = comps that understand human lang
- Computational Creativity = comps that can make creative works of art
- Narrow AI = task-specific AI, good as individual task but doesn't have the kind of general task independent intelligence that humans have

AI in the early days used logical rules, for example, expert systems; a type of AI system that tried to do things that expert human would do (ie. diagnosing a disease). One of the big problem with AI based on logical rules was that the rules the AI would follow were very rarely certain. Gradually, AI was developed with the mathematics of probability in mind to deal with the uncertainty in the rules

- Machine Learning(ML) = Instead of programming a comp by telling it every detail of what to do, we "teach" it by giving it examples of what to do. ML uses statistical algos to learn from examples (data); comp learns from data. ML is creating statistical progs called Models; a model takes an input and gives back an output.
    - Model is created with a lot of example data. Each example includes both an input and an output. A ML algo takes the examples & uses them to train the model, this means it adapts the details of the model, so it maps the input in ex data to corresponding outputs
    - In common type of ML, output is one of a number of categories called classes, putting things into categories
- Regression = if the output of ML is a number or several numbers
- Generative Model = where ML takes the given example and creative new examples as output
- Supervised Learning = Training a model on examples of input and output
- Unsupervised Learning = giving the ML model a set of inputs and getting it to figure the categories
- Reinforcement Learning = much like training a dog, the algo does things and gets rewards for doing good things and gets punishment for doing the wrong thing
- Nearest Neighbor = classifying a picture based on its nearest example, which is called a neighbor
    - To reduce errors, we K nearest neighbors or kNN, closest examples and choosing the class that most of them have.
    - A problem with NN is that it can be slow, gotta search through every example to find the most similar
    - Solution could be to not the original examples directly, but use them to create a math function to do the classification, called a model
- Support Vector Machines = comparing with a small important examples together with a sophisticated similarity measure and etc
- Optimization = choosing the numbers that we use in our model so that they give us the best results on the training data
- Decision trees = combines many simple decisions on individual features. Algo performs one decision after another, choosing features to use based on previous decision; the features to use, the order in which they're used & threshold values are all learned from data using optimization
- Neural Networks = combine many features together into complex math functions & only make one decision based on the output of that function. The details of function are also learned from data using optimization

Summary

In this week, we learned about what AI & ML are and the different types of ML algorithms used when performing supervised classification.

17.1 AI & 17.2 ML-Reading

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2020-02-04 6:01 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 17.1 Artificial Intelligence 17.2 Machine Learning -Reading	Course: BSc Computer Science Class: How Computer Work [CM1030]-Reading Date: February 03, 2020

Essential Question:

What is AI?

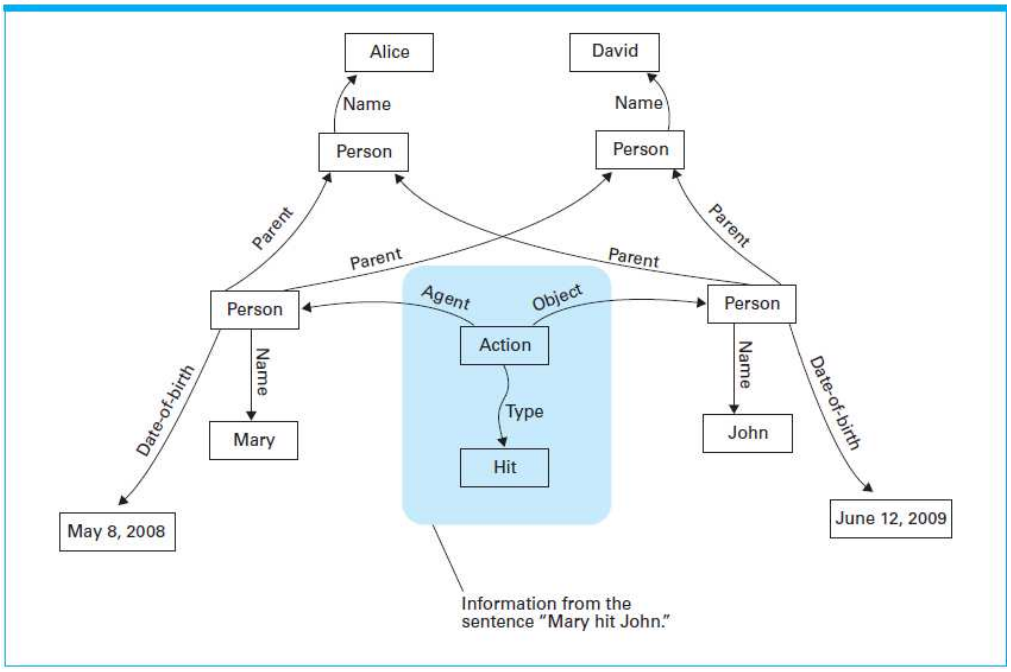
Questions/Cues:

- What is an agent?
- What is procedural knowledge?
- What is declarative knowledge?
- What is the Turing Test?
- What is image processing?
- What is image analysis?
- What is edge enhancement?
- What is region finding?
- What is Information Retrieval?
- What is Information Extraction
- What is a Semantic Net?
- What is a production system?
- What is a collection of States?
- What is a collection of Productions?
- What is a Control System?
- What is a Problem Space?
- What is a State Graph?
- What is a Search Tree?
- What is the Breadth-first manner of tree construction?
- What is the Depth-first manner of tree construction?
- What is meta-reasoning?
- What is the closed-world assumption?
- What is the frame problem?
- What is learning by Imitation?
- What is supervised training?
- What is the training set?
- What is Reinforcement Learning?
- What is a genetic algorithm?
- What is Evolutionary Programming?

Notes

- Agent = "device" that responds to stimuli from its environment. Most agents have sensors by which they receive data from their environment (microphones, cameras, range sensors) & actuators by which they affect their environment (wheels, legs, wings)
- Procedural Knowledge (learning "how") = involves a trial-&-error process by which agent learns appropriate actions, being punished for poor actions & rewarded for good ones.
- Declarative knowledge (learning "what") = form of expanding or altering the "facts" in an agent's store of knowledge from which rational responses to future events are determined
- Turing test =a test proposed by Alan Turing in 1950, allows a human (interrogator) to comm with test subject via a typewriter system without being told whether test subject was human or machine. In this environment, machine would be declared to behave intelligently if human was unable to distinguish it from human
- Image Processing = refers to identifying characteristics of image
- Image Analysis = refers to process of understanding what image characteristics mean
- Edge Enhancement = process of applying mathematical techniques to clarify boundaries between regions in image
- Region Finding = process of identifying those areas in image that have common properties such brightness, color, or texture
- Information retrieval = identifying docs that relate to topic at hand
- Information extraction = task of extracting info from docs so it takes a form useful in other apps
- Semantic Net = large linked data structure in which pointers are used to indicate associations among data items

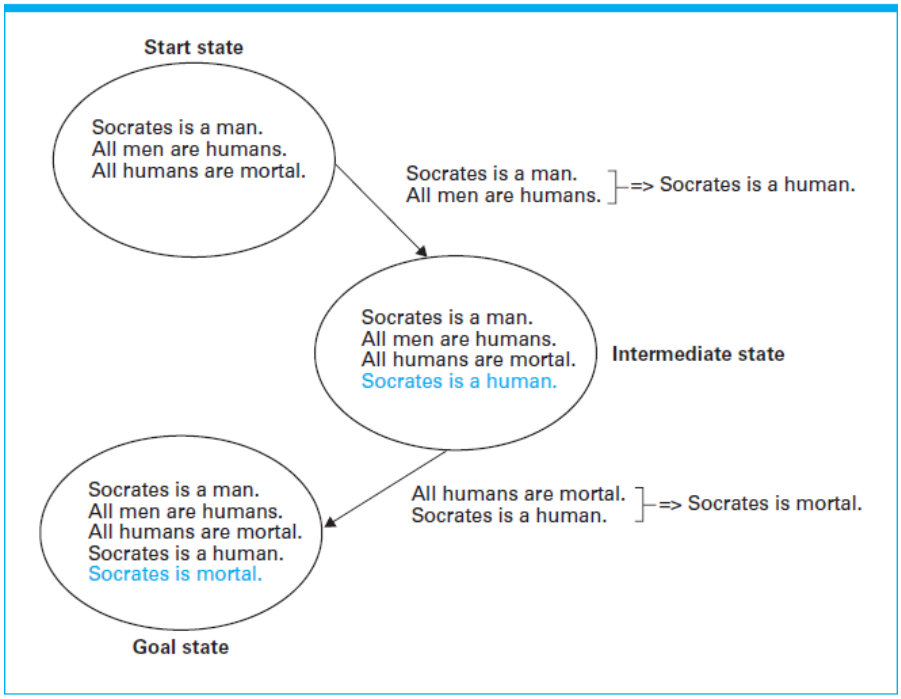
Figure 11.3 A semantic net



- Production System = large class of reasoning problems with common characteristics
  - Collection of States = Each state is situation that might occur in app environment. Beginning state is called start (initial) state; desired state (or states) called goal state
  - Collection of productions(rules or moves)= Production is operation that can be performed in app environment to move from one state to another. Each production might be associated with preconditions; conditions might exist that must be present in environment before production can be applied.
  - Control System = Control system consists of logic that solves problem of moving from start state to goal state. At each step in process control system

must decide which of those productions whose preconditions are satisfied should be applied next

Figure 11.5 Deductive reasoning in the context of a production system



- Problem Space = collection of all states, productions, & preconditions in production system
- State Graph = the conceptualized form of a problem space. Often a directed graph, a collection of locations called nodes connected by arrows. State graph consists of collection of nodes rep'ing the states in the system connected by arrows rep'ing the productions that shift system from one state to another. Two nodes connected by arrow in state graph if and only if there is production that transforms system from the state at the origin of the arrow to state at destination of arrow
- Search tree = consists of the part of the state graph that has been investigated by control system. Root node of search tree is start state & children of each node are those states reachable from parent by applying one production. Each arc between nodes in search tree reps the application of single production, & each path from the root to a leaf reps a path between the corresponding states in state graph
- Breadth-first manner = meaning tree is constructed layer by layer
- Depth first manner = tree is constructed by building vertical paths rather than horizontal layers
    - This approach is often referred to as best-first construction in recognition of the fact that the vertical path chosen for pursuit is the one that appears to offer best potential
- Meta-reasoning = reasoning about reasoning
- Closed-world assumption = assumption that a statement is false unless it can be explicitly derived from info available
- Frame problem = problem of keeping stored knowledge up-to-date in a changing environment
- Imitation-learning = in which a person directly demonstrates steps in task, perhaps by carrying out a sequence of comp operations or by physically moving a robot through a sequence of motions & comp simply records the step
- Supervised training = person identifies correct response for series of examples & then agent generalizes from those examples to develop an algorithm that applies to new cases
- Training set = series of examples
- Reinforcement learning = agent is given a general rule to judge for itself when it has succeeded for failed at a task during trial & error

    - Agent acts autonomously as it learns to improve its behavior over time
- Genetic Algorithm = will discover a solution by random behavior combined with a simulation of reproductive theory & evolutionary process of natural selection
    - GA begins by generating a random pool of trial solutions (each solution is just a guess)
    - Each trial solution called a chromosome & each component of chromosome called a gene
    - GA proceeds to generate a new pool of chromosomes whereby each chromosome is offspring(child) of two chromosomes(parents) of previous pool
    - Parents are randomly selected from pool giving a probabilistic preference to those chromosomes that appear to provide best chance of leading to solution; emulating evolutionary principle of survival of the fittest
    - Each child is random combination of genes from parents. Resulting child may occasionally be mutated in some random way.
    - Hopefully by repeating this processing over & over, better & better trial solutions will evolve until a very good one, if not the best, is discovered. But is no assurance that GA will find a solution
- Evolutionary programming = When GA applied to prog development it's called Evolutionary programming. Goal is to develop progs by allowing them to evolve rather by explicitly writing them.
    - In functional prog langs, you begin with a collection of progs that contain a rich variety of functions. The functions in starting collection form "gene pool" from which future gens of progs will be constructed. This allows evolutionary process to run for many gens, hoping that by producing each gen from best performers in previous gen, a solution to target problem will evolve.

Summary

In this week, we learned about what AI is & how "intelligence" is produced in machines.

18.1 Data Features

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2020-02-06 1:17 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 18.1 Data Features	Course: BSc Computer Science Class: How Computer Work [CM1030]-Lecture Date: February 05, 2020

Essential Question:

How does data representation affect machine learning?

Questions/Cues:

- What are features?
- What is Feature Extraction?
- What is a weight?
- What is a Neuron?
- What is a Convolutional neural?

Notes

- Features = name for the number we use to rep thing in ML
  - Pixels are an example of a feature for images, low-level or raw data feature that reps the image as it is but individual pixel don't carry much meaningful info
  - Sound signal is low-level feature for both music and human lang
    - Both music & human lang have high-level features like musical notes or written words which are better to use when working with ML
- Feature extraction = extracting more meaningful features from the raw data
- Weight = a numerical value multiplied in each existing feature to determine its contribution/importance in the calculation of a brand new feature. Weight is determined by optimization for best suited results
- Neuron = one unit of neural network
  - The word deep in deep learning means that you have many layers of neurons feeding into other neurons. Even the basic calculations are fairly simple, a deep neural network can learn very complex features
  - Simply feeding the output of one neuron into another is actually equivalent to simple neurons with different weights, it doesn't allow you to calculate any more complex features. You need to transform the output by a mathematical function called a nonlinearity
    - One common nonlinearity takes any output that's negative and sets it to zero
  - Simplest type of neuron just multiplies its input features by weight, adds them up and if the output is negative, sets it to zero. This simple network can learn very complex feature by choosing good weights

- Convolutional neural = basically a Photoshop filter that transforms an image, details of the filter can be learned by optimization

Summary

In this week, we learned about what features are, what weights are and what a neuron is.

18.1 Data Features-Reading

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2020-02-07 1:03 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 18.1 Data Features-Reading	Course: BSc Computer Science Class: How Computer Work [CM1030]-Lecture Date: February 05, 2020

Essential Question:

How does data representation affect machine learning?

Questions/Cues:

- What is a Artificial Neural Network?
- What is an biological neuron?
- What is associative memory?
- What are Hopfield Networks?

Notes

- Artificial Neural Network = a comp processing model that mimics networks of neurons in living biological systems
- Biological Neuron = single cell with input tentacles called dendrites and an output tentacle called the axon. Signals transmitted via cell's axon reflect whether cell is in inhibited or excited state. State is determined by combination of signals received by cell's dendrites, dendrites pick up signals from axons of other cells across small gaps known as synapses
    - A neuron in artificial neural network is software unit that mimics biological neuron
        - Artificial Neuron produces output of 1 or 0 depending on whether its effective input exceeds a given value called neuron's threshold. Effective input is weighted sum of actual input
        - Neuron receiving inputs from other axons of other neuron multiplies each by associated weight for the connection and adds products to form effective input. If this sum exceeds neuron's threshold, neuron produces an output of 1(excited state) or output of 0 (inhibited state)
        - Neurons are arranged in topology of several layers, where inputs (first-layer) and outputs are last-layers and there are hidden layers in between'

Figure 11.15 A neuron in a living biological system

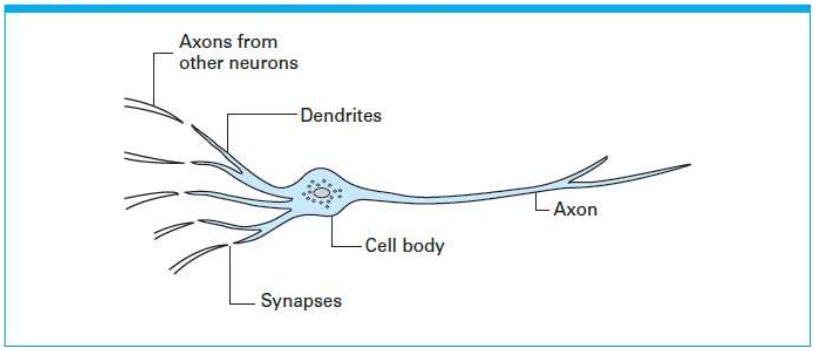


Figure 11.16 The activities within a neuron

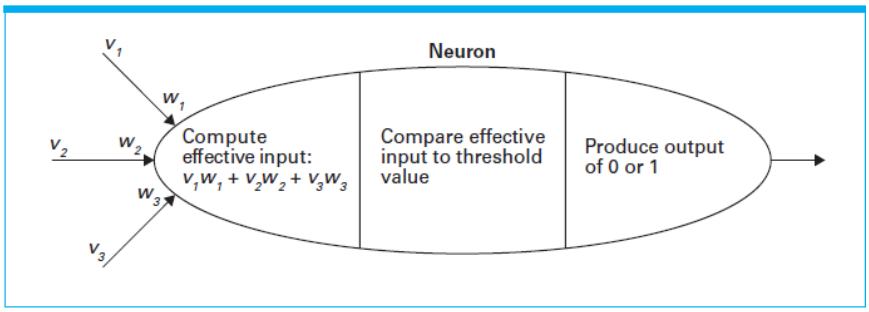
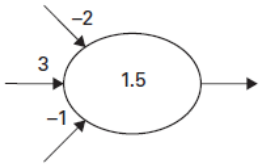


Figure 11.17 Representation of a neuron



- Associative Memory = retrieval of info that is associated with or related to info at hand
- Hopfield Network = output of each neuron is connected as inputs to each of the other neurons; in other cases, the output of a neuron may be connected only to its immediate neighbours

Figure 11.20 An artificial neural network implementing an associative memory

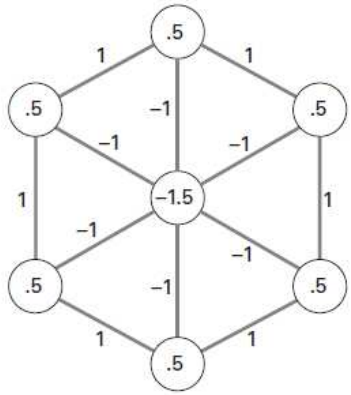
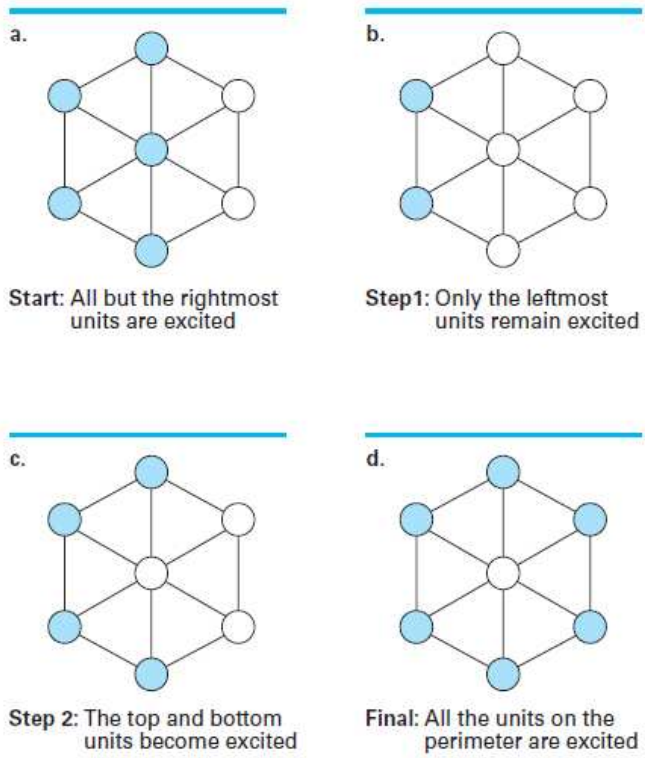


Figure 11.21 The steps leading to a stable configuration



Summary

In this week, we learned about what features are, what weights are and what a neuron is.

Week 1

Abstraction: Abstraction is the process used to break complex ideas or structures down into simpler forms that are easier for humans to understand, or easier for computers to process.

Abstraction in Data Representation: Complext forms of data are often broken down into simpler components in order to encode them in a numeric form and process them. Eg: A document might be broken down into a string of characters with some metadata.

Notional Machines: Notional machines are an abstraction of real computers which simplify their internal processes so humans may understand their operation. Eg: the turing machine

CPU: The CPU is a component of a computer that executes a program. It reads instructions from memory, and uses those instructions to manipulate data in memory.

Memory: Memory is a broad term used to describe several components of a computer which temporarily store data and instructions while they are being used. Memory is normally volitile, which is to say it only stores information while the computer is in use. It is distinct from "storage" which is used to hold data between sessions, or retain information for programs not currently in use.

Week 2

State: An abstract concept used to describe how a machine (or program) has a number of distinct configurations. Eg: A video player might have a paused or play state. A strange state is a configuration not intended by the programmer, it can often be resolved by restarting the machine or program.

Files: Files are digital containers of information. They will typically contain data as well as metadata, which is information about the data and the file itself. When a file is being used by a program it will often be locked by the operating system to prevent another program from editing it at the same time. Some operating systems also have an "executible bit" attached to a file, which tells the OS whether the contents of the file may be executed as an instruction or not.

Modularity: Modularity is a design approach which separates a system into distinct chunks called modules. Each module has a detailed specification and may be implemented separately from ther others, by another person, another team or even another organisation.

Driver: A piece of software used to control a hardware device. It presents a generic interface to other pieces of software which can interact with it like any other piece of hardware of a similar type. Eg: Libreoffice writer can use a driver for an epson 3200 just the same as a Brother 500.

Applications: An application is a program intended to be interacted with by the end user. They are comprised of code libraries and resource files.

Utility: A utility is an application developed by the OS developer and bundled with the OS.

Week 3

Networking

Networks: A system connecting multiple machines together and allowing them to communicate. Might be a physical connection (eg. Copper cable or fiber optic) or a wireless connection (eg. Bluetooth or infrared).

The Internet: A global network of networks.

Internet Backbone: A top level link between ISPs, usually running between countries.

URL: A human readable network address indicating the location of a resource (file)

IP Address: A machine readable network address indicating the location of a particular machine

DNS: Domain Name Service. An organisation linking human readable domain names to IP addresses.

Security

Malware: General term for software that is intended to cause harm

Virus: A piece of malware which propagates by inserting it's code into another piece of software. This software is then used to spread the virus between systems.

Worm: Similar to a virus except that the malware is standalone software and does not require any other software to propagate.

Trojan: A piece of malware which is copied between systems by users, by presenting itself as non-malware

Spyware: Malware which is intended to extract information from the user without their knowledge

Ransomware: Malware which seizes control of a system, only releasing control when payment is made by the user.

Encryption: A system of rendering a message unreadable to messengers, which then may be rendered readable again by the intended recipient.

Firewall: A system which examines incoming and outgoing traffic from a system and limits them according to predetermined security rules. Firewalls may govern interactions between networks (network firewalls) or individual machines (host-based firewalls).

Week 4

The Internet

Web page: A mixture of document and software which presents information via a browser

Website: A collection of related web pages, typically developed by a single organisation and located at a single domain.

The world wide web: The superset of all websites available via the internet. Specifically everything using HTTP and derivatives.

Protocol: A prearranged system for communicating information

Cookie: A small file stored by a website on an end user's computer to track their usage.

Client: The machine used by an end user

Server: A central machine which provides data and services to clients

Database: A piece of software which manages a structured array of data.

Embedding: The process of including assets from other web pages in another web page. Eg.: An advertisement banner or a youtube video

API: Application programming interface. An interface intended for use by software. Often used by websites to allow third party sites to access their data. Eg: A twitter search engine

Week 5

Data

Bit: A single binary digit, has two states (0 and 1). Can be used to represent all kinds of different information.

Byte: 8 bits. A standard unit of memory since the 1980s. Most programs will not interact with memory in units smaller than a byte.

Word: A unit of memory specific to a particular architecture. Some systems use 8 bit words, others use 16, 32 or 64 bit words. The longer a word, the more precision the system is capable of and the greater it's paralellism.

ASCII: American Standard Code for Information Interchange. An alphabetic encoding system, originally for telex machines. ASCII codes use 7 binary bits, so the remaining bit in each byte was used for different purposes depending on the system. Eg: Storing highlighting information

Unicode: Unicode is a successor to ASCII and provides support for a much wider range of alphabets, ideographics and other symbols. It does not have a fixed code length and any one glyph may be encoded using between 1 and 4 bytes.

Week 6

Compression

Compression: A system by which less bits may be used to represent the same information. With more or less loss of detail.

Dictionary encoding: The use of an index to represent patterns in a file using shorter symbols.

Adaptive Dictionary Encoding: Begins by recognising basic blocks the file is composed of, but then recognises larger superblocks comprised of simpler blocks. Eg: LZW encoding

Run length encoding: Describing a pattern in terms of how long it runs for, rather than recording it's entire length.

Frequency dependent encoding: An encoding system that takes into account how often a pattern occurs, and assigning a longer or shorter symbol to represent it. Also known as Huffman codes.

Relative encoding: Encoding a data stream as distinct data units. Only changes between data units are recorded. Most useful with files which can be broken down into highly repetitive chunks, such as frames from a cartoon. Also known as differential encoding.

GIF: A type of dictionary encoding intended for use with images. Begins by comparing each pixel to a fixed size pallette, then picking the closest match. LZW encoding is then employed to detect patterns.

JPEG: Another image compression format. Takes advantage of the human eye's lower number of cone cells (colour recognition) compared to rod cells (brightness recognition). Initially the image is reencoded in terms of luminence (brightness) and chrominence (colour). The chrominence values are then averaged over 2x2 pixel blocks, allowing the filesize to be substantially reduced without a significant loss in perceived quality. The resulting image is then further compressed and changes beyond those easily detected by the human eye are reduced to zero.

TIFF: Another image filetype. Includes compression techniques similar to GIF and has metadata storage.

PNG: A non patented and improved version of GIF. Intended for transferring images via networks. Consists of a header, identifying the filetype for various systems, followed by a series of "chunks," each of which contains data or metadata. Chunks may be critical (must be decoded) or ancillary (may be decoded).

MPEG-4: A type of file format intended to store video. It is a "container format," which means that a single .mp4 file can contain multiple subfiles, such as video tracks, audio tracks, subtitles, metadata etc.

MKV: Matroska, a video container format intended as an open source alternative to MPEG-4 or AVI.

MP3: A lossy audio compression format which prioritises detail for frequencies human ears are most sensitive to.

Week 7

Computer Architecture

Bus: A common pathway used by multiple components to transmit information. May be serial (information is sent as pulses along a single pathway, possibly being muxed/demuxed at either end) or parallel (information is sent side by side using several pathways).

Register: A single memory cell housed on the CPU itself (or possibly another piece of hardward). Typically stores a single word, on some architectures registers may be paired to store two words. Registers may be general use (used for storing data during calculations) or special purpose (typically used to control hardware, such as jump vectors or memory pointers).

Arithmetic Logic Unit: The component in a CPU which carries out arithmetic instructions, taking inputs from one or more registers, and using it to create an output, to be stored in a register or memory.

Control Unit: The part of the CPU that receives, orders and interprets instruction. Directing data flow and activating processing circuitry as appropriate. Internal components may include: Instruction decoder, Program counter, Instruction register.

Program Counter: The part of the control Unit that tracks which instruction is currently being executed. It contains a register called the instruction pointer, which contains the address of the current instruction, and the instruction register which contains the instruction itself. Typically once a CPU cycle is complete it will increment the instruction pointer, moving to the next instruction. It is also capable of "branching" by jumping to a completely different instruction if particular conditions are met.

Clock: Typically a quartz oscillator. A component which provides electronic pulses to the system telling it how long to spend on any one task.

Motherboard: A circuitboard providing connectivity and glue logic for most of a computer's main components.

Cache Memory: High speed ram stored on the CPU die itself. Also called "L1 (level 1) Cache."

Machine Language: The set of instructions a CPU is capable of interpreting. The lowest form of code, being run directly by the hardware with no interpreter or compiler being involved. Typically machine language would be created using a "machine language monitor." Pure machine language is rarely written directly by humans and is more commonly written in a more human readable form before being translated into machine code by an assembler.

RISC: Reduced instruction set computer. A design philosophy intended to keep machines fast by keeping them simple, and implementing a basic set of instructions which can be combined to form more complex operations. In practice there is no clear

definition of how reduced an instruction set must be to qualify, and many modern RISC architectures are highly complex/bloated.

CISC: Complex instruction set computer. The opposite design philosophy which aims to provide a comprehensive instruction set to the programmer.

MISC: Minimal instruction set computer. Aims to be even simpler than RISC. In practice these are often hobby machines intended to push the limits of turing completeness.

Machine Language

OP-Code: The part of an instruction that indicates the type of instruction to be carried out.

Operand: The part of the instruction indicating what the subject(s) of the operation are to be. Eg, a memory address or a register.

Machine cycle: The process of fetching instructions interpreting them, and executing them.

Illustrative Machine Language

Generic instruction names (from the textbook) are in black. Instruction names from the excercise are in red. These may come up on the exam!

LOAD (MEMLOAD): Retrieve a value from memory and write it to a register
STORE (MEMSTORE): Write a value from a register to memory
WRITE (VALLOAD): Writes a value directly to a register
? (STORE): ?
JUMP (JUMP): Switch the instruction pointer to a new location in memory, branches the program.
STOP: Halts the program

Coffman (1971) identified **four (4) conditions** that must hold simultaneously for there to be a deadlock.

1. Mutual Exclusion Condition
The resources involved are non-shareable.

2. Hold and Wait Condition
Requesting process hold already, resources while waiting for requested resources.

3. No-Preemptive Condition
Resources already allocated to a process cannot be preempted.

4. Circular Wait Condition
The processes in the system form a circular list or chain where each process in the list is waiting for a resource held by the next process in the list.

Mass storage:
Track: An individual circle of bits on a hard disk
Sector: A chord of a track
Zone: Tracks are organised into zones, each of which have the same number of sectors. Zones towards the edge of a disk have more sectors, zones towards the center have less.

OSI: Open Systems Interconnection model

- **Physical layer:** Governs all the actual circuitry and cable as well as standards for bitrate. Includes switches and repeaters
- **Link Layer:** Protocols for exchanging packets between networks, WiFi, Ethernet etc.
- Network Layer: IP
- **Transport Layer:** TCP, UDP etc.
- **Session layer:** Manages the sessions of individual machines on the network, joining, terminating etc.
- **Presentation Layer:** Abstracted interface of an application for interacting with other applications
- **Application Layer:** The meat interface that lets you actually interact with the network