BITS & BYTES

Binary Data in Memory

© Copyright: Jasleen Kaur. 2011.

What does memory look like?

MEMORY SNAPSHOT

```
10101101101010101010101010101010
10101010101010101010111101010 1
10101010100100010101010101010 1
01010101011110101010100010101 0
01010101010101011010010111010 0
```

- All computers are binary! (0, 1)
  - Represented electronically, magnetically, etc.
- Binary is used to store everything:
  - Numbers: 0, 1, -50, …
  - Characters: a, $,), …
  - Instructions: ADD, STORE, …
  - Colors: Red, Blue, Green, …

© Copyright: Jasleen Kaur. 2011.

© Copyright: Jasleen Kaur. 2011.

Binary number system

Decimal system

5 3 1 2

2	1's	10^0
1	10's	10^1
3	100's	10^2
5	1000's	10^3
	…	

Binary system

1 0 1 1 0

0	1's	2^0
1	2's	2^1
1	4's	2^2
0	8's	2^3
1	16's	2^4
	…	

© Copyright: Jasleen Kaur. 2011.

Binary values

BIT: Binary digIT

1 bit	2 bits	3 bits	4 bits	
0 =0	00 =0	000 =0	0000 =0	1000 =8
1 =1	01 =1	001 =1	0001 =1	1001 =9
	10 =2	010 =2	0010 =2	1010 =10
	11 =3	011 =3	0011 =3	1011 =11
		100 =4	0100 =4	1100 =12
		101 =5	0101 =5	1101 =13
		110 =6	0110 =6	1110 =14
		111 =7	0111 =7	1111 =15

Each additional bit doubles the number of possible permutations

N bits represent values 0 to $2^N - 1$

© Copyright: Jasleen Kaur. 2011.

© Copyright: Jasleen Kaur. 2011.

Bytes

- Most computers use groups of 8 bits: BYTE
- Each byte
  - Can store numbers: 00000000 (0) to 11111111 (255)
  - Has a memory address: 0, 1, 2, …
- To store bigger numbers, we use several bytes:
  - 2 Bytes 0 to 65,535 (< 64 K)
  - 4 Bytes 0 to 4,294,967,295 (< 4 G)
    - Or use 1 bit for sign ±
  - 4 Bytes ± 2,147,483,648 (< 2 G)

© Copyright: Jasleen Kaur. 2011.

Storage Capacity

- Every memory device has a storage capacity, indicating the number of bytes (8 bits) it can hold
- Various units:

Unit	Symbol	Number of Bytes
kilobyte	KB	2^{10} = 1024
megabyte	MB	2^{20} = 1.048 million
gigabyte	GB	2^{30} = 1.073 billion
terabyte	TB	2^{40} = 1.099 trillion

© Copyright: Jasleen Kaur. 2011.

© Copyright: Jasleen Kaur. 2011.

Computer Science Joke

There are 10 kinds of people.

Those who understand binary and those who do not.

© Copyright: Jasleen Kaur. 2011.

COMPUTER SOFTWARE SYSTEMS

Software: Compilers, Operating Systems, …

© Copyright: Jasleen Kaur. 2011.

© Copyright: Jasleen Kaur. 2011.

Programs in Binary

- One implication of binary memory is that programs & data look like this!

```
10101101101010101010101010101010
10101010101010101010111101010 1
10101010100100010101010101010101
01010101011110101010100010101 0
01010101010101011010010111010 0
```

- Most people write programs in a human-readable form
  - And then use a program to translate to binary
- Typical translation steps:

High-level languages ⟶ Assembly code ⟶ Binary code

Compiler / Interpreter

Assembler

© Copyright: Jasleen Kaur. 2011.

Assemblers

- Assembly code:
  - Mnemonic instruction names

    01011010 ☒ ADD ☑
  - Mnemonic memory locations

    1000 ☒ SALARY ☑
- Basically, assembly code is 1-1 to binary machine code
  - Just easier to read
- Assembler:
  - Program that translates assembly code to machine code

    ADD SALARY, BONUS → 01011010 …

© Copyright: Jasleen Kaur. 2011.

© Copyright: Jasleen Kaur. 2011.

Compilers

- Higher-level source language
  - 1 source line → possibly many machine language instructions

    salary = regHours*wage + otHours*wage*1.5;
    - 3 multiply
    - 1 add
    - 1 store
- Compiler:
  - Translates high-level source to assembly code
- Need both compiler and assembler to run

  source → assembly code → binary machine code
  - Translated once (executed many times)

© Copyright: Jasleen Kaur. 2011.

Interpreters

- Translate and execute each line as it is encountered
- Pros:
  - No start-up delay
  - Unexecuted code is not translated
  - Better error handling
- Cons:
  - Slower – repeated code is translated repeatedly
- We'll see both compilation and interpretation

  Java source → byte code (intermediate) → machine code

  Compiler: Java source → byte code

  Interpreter: byte code → machine code
  - Helps in portability

© Copyright: Jasleen Kaur. 2011.

© Copyright: Jasleen Kaur. 2011.

Operating Systems: a little history

- Old days:
  - run - 30 min
  - change over program - 1 min
  - run - 30 min
  - …

  } 97 % efficiency
- Then, computers got faster; but people didn't
  - run – 1 sec
  - change over program – 1 min
  - …

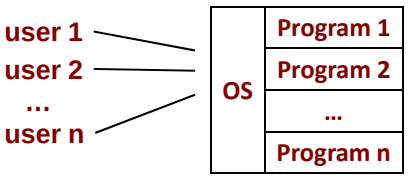
  } 1.6 % efficiency ! ☹
- Basis for operating system (OS):
  - A program to manage transition from one program to another

© Copyright: Jasleen Kaur. 2011.

Operating Systems: time-sharing

- Busy executive needs information for project X
  - Asks assistant, waits for response. Now what?
  - Work on another project !

- Time-sharing:
  - Computers much faster than humans
  - Also helps share costs!



© Copyright: Jasleen Kaur. 2011.

© Copyright: Jasleen Kaur. 2011.

INTERNET & THE WEB

Client-server architecture

© Copyright: Jasleen Kaur. 2011.

Computer Networks

- Enable communication
  - Allow sending messages of <u>any length</u> <u>reliably</u> from any computer to another
- Enable information-sharing, e-commerce, …
  - e.g., emails, chat, facebook, wikipedia, amazon, …
- Enable sharing of resources:
  - Advent of PCs in 1980s $\Longrightarrow$ sharing was lost !
    - Expensive devices: printer
    - Expensive software : compiler
  - Networks restored resource sharing
    - Connect computers to printers, scanners, …

© Copyright: Jasleen Kaur. 2011.

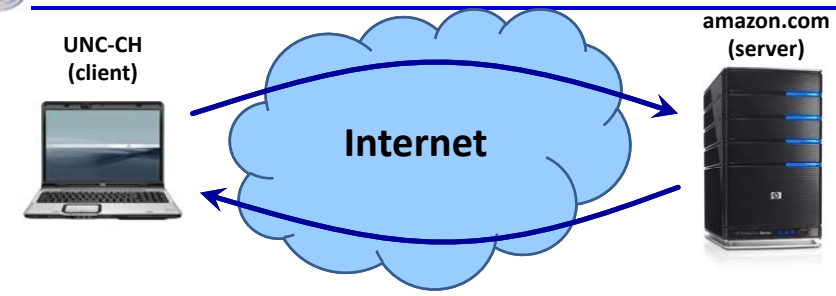
© Copyright: Jasleen Kaur. 2011.

What is WWW?

- World-wide Web (WWW)
  - A system of inter-linked *hypertext* documents
  - Accessed via the Internet
- WWW ≠ Internet
  - Internet is the *infrastructure* on which WWW is built
  - e.g., FEDEX built on top of airports, air traffic control
- WWW built using the "client-server architecture"

© Copyright: Jasleen Kaur. 2011.

Client-server architecture

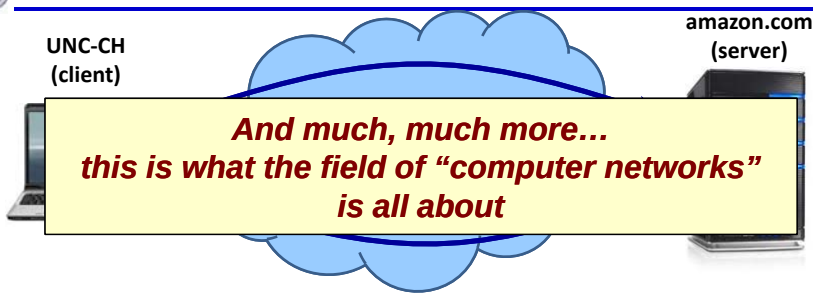


- Server: a computer running a server program
  - Waits (listens) for and responds to client initiation/requests
  - Always "on"
  - e.g., Web servers, email servers, ftp servers, gaming servers
- Client: a computer running a client program
  - Initiates contact with the server
  - Sends a request to the server (after initiation)
  - e.g., browser (web), thunderbird (email)

© Copyright: Jasleen Kaur. 2011.

© Copyright: Jasleen Kaur. 2011.

Things needed to make this work



And much, much more…
this is what the field of "computer networks" is all about

- A way for client to translate amazon.com to a server ID
  - Domain Name System (DNS)
- A path for client's initiation request to reach server
  - Routing protocols (BGP), Internet protocol (IP)
- A way for server to ensure that file delivered properly
  - Transmission Control Protocol (TCP)

© Copyright: Jasleen Kaur. 2011.

WWW client-server model

- Server: web server; has access to files
- Client: browser (Internet Explorer, Firefox, Safari, …)
- When you click / enter a link:
  - Client sends URL (Uniform Resource Locator) to server
    - Identifies the server file requested

http://www.cs.unc.edu/~jasleen/Courses/Fall11/slides/1-introduction.pdf

protocol | server | directories / folders | name | type

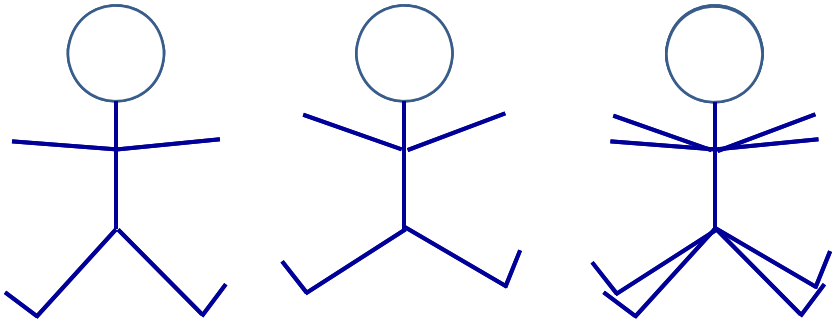
  - Server sends back the file
  - Browser displays the file

Your UNC web-page: http://www.unc.edu/~onyen !

© Copyright: Jasleen Kaur. 2011.

© Copyright: Jasleen Kaur. 2011.

Displaying motion



© Copyright: Jasleen Kaur. 2011.

© Copyright: Jasleen Kaur. 2011.