Topic 3: Information and Data

What is Information?

What is Data?

How do Computers Represent Information?

What is Information?

What is Information?

- Etymology: Latin, “to give form to” or “to form an idea of”
- Definition: The state of being of an object or system of interest

What is Data?

Information Processing

- A change of information in any manner detectable by an observer
- Using a computer?
  - Encode information into data
  - Process the data
  - Translate data back into information

Storing Data

- All data in a computer is either a 0 or 1
  - Called a bit
  - Electrically, this is a switch that is either open or closed
- Encoding schemes translate integers, real numbers, letters, pictures, … into bits

Boolean Data

- Has two possible values
  - True
  - False
- Easily encoded using a single bit
  - 0: False
  - 1: True

Integer Data

- How do we represent the numbers 5, 24, or 367 using only ones and zeros?

Number Systems

- Decimal (Base 10)
  - 10 distinct symbols
  - Each digit is a factor of 10 larger than the digit to its right

Number Systems

- Binary (Base 2)
  - 2 distinct symbols
  - Each digit is a factor of 2 larger than the digit to its right

Counting in Binary

Binary Numbers

- Consider the base 2 number 1001101_2
  - What base 10 number does it represent?

Binary to Decimal

- Convert 1111_2 to base 10:
- Convert 100010_2 to base 10:
- Convert 0_2 to base 10:

Decimal to Binary

- What sequence of bits represents the decimal number 12?

The Division Algorithm

- Allows us to convert from Decimal to Binary
  - Let Q represent the number to convert
  - Repeat
    - Divide Q by 2, recording the Quotient, Q, and the remainder, R
  - Until Q is 0
  - Read the remainders from bottom to top

Decimal to Binary

- What sequence of bits represents the decimal number 12?

Decimal to Binary

- Convert 191_{10} to Binary:

Decimal to Binary

- Convert 222 Base 10 to Base 2:

Integer Data

- Base 10 integers can be represented using sequences of bits
  - Common sizes:
    - 8 bits (referred to as a byte)
    - 32 bits (referred to as a word)
    - 64 bits (referred to as a double word / long)
    - 16 bits (referred to as a half word / short)

Negative Numbers

- How can we represent negative numbers?
  - Choose an encoding where we choose that some bit patterns represent positive numbers and others represent negative numbers
  - Simple Idea:
    - Left most bit is the sign – 0: positive, 1: negative
    - Rest of the bits represent the number
  - Other ideas:
    - One's Complement, Two's Complement, Base -2, Excess N, …

Other Bases

- A number system can have any base
  - Decimal: Base 10
  - Binary: Base 2
  - Octal: Base 8
  - Hexadecimal: Base 16
  - Vigesimal: Base 20
  - Base 6
  - Any other number we choose…

Hexadecimal

- Convert 507_{10} to base 16:

Hexadecimal

- Convert 180_{10} to base 16:

Hexadecimal

- Convert 0xA1 to decimal:
- Convert 44 base 16 to decimal:
- Convert $CAFE_{16}$ to base 10:

Utility of Hexadecimal

- Common to have groups of 32 bits
  - 32 bits is cumbersome to write
  - easy to make mistakes
- Use hexadecimal as a shorthand
  - 8 hex digits instead of 32 bits
  - Group bits from the right
  - Memorize mapping from binary to hex for values between 0 and F

Utility of Hexadecimal

- Convert 0xF51A to binary
- Convert 10010010101011010100 from binary to hex

Representing Characters

- What characters do we need to be able to represent?

Representing Characters

- Standard encoding scheme called ASCII
  - American Standard Code for Information Interchange
    - 7 bits per character
  - Includes printable characters
  - Includes “control characters” that impact formatting (tab, newline), data transmission (mostly obsolete)
  - Layout seems arbitrary, but actually contains some interesting patterns

0	0000 0000	NUL	32	0010 0000		64	0100 0000	@	96	0110 0000	`
1	0000 0001	SOH	33	0010 0001	!	65	0100 0001	A	97	0110 0001	a
2	0000 0010	STX	34	0010 0010	"	66	0100 0010	B	98	0110 0010	b
3	0000 0011	ETX	35	0010 0011	#	67	0100 0011	C	99	0110 0011	c
4	0000 0100	EOT	36	0010 0100	$	68	0100 0100	D	100	0110 0100	d
5	0000 0101	ENQ	37	0010 0101	%	69	0100 0101	E	101	0110 0101	e
6	0000 0110	ACK	38	0010 0110	&	70	0100 0110	F	102	0110 0110	f
7	0000 0111	BEL	39	0010 0111	‘	71	0100 0111	G	103	0110 0111	g
8	0000 1000	BS	40	0010 1000	(	72	0100 1000	H	104	0110 1000	h
9	0000 1001	TAB	41	0010 1001	)	73	0100 1001	I	105	0110 1001	i
10	0000 1010	LF	42	0010 1010	*	74	0100 1010	J	106	0110 1010	j
11	0000 1011	VT	43	0010 1011	+	75	0100 1011	K	107	0110 1011	k
12	0000 1100	FF	44	0010 1100	,	76	0100 1100	L	108	0110 1100	l
13	0000 1101	CR	45	0010 1101	-	77	0100 1101	M	109	0110 1101	m
14	0000 1110	SO	46	0010 1110	.	78	0100 1110	N	110	0110 1110	n
15	0000 1111	SI	47	0010 1111	/	79	0100 1111	O	111	0110 1111	o
16	0001 0000	DLE	48	0011 0000	0	80	0101 0000	P	112	0111 0000	p
17	0001 0001	DC1	49	0011 0001	1	81	0101 0001	Q	113	0111 0001	q
18	0001 0010	DC2	50	0011 0010	2	82	0101 0010	R	114	0111 0010	r
19	0001 0011	DC3	51	0011 0011	3	83	0101 0011	S	115	0111 0011	s
20	0001 0100	DC4	52	0011 0100	4	84	0101 0100	T	116	0111 0100	t
21	0001 0101	NAK	53	0011 0101	5	85	0101 0101	U	117	0111 0101	u
22	0001 0110	SYN	54	0011 0110	6	86	0101 0110	V	118	0111 0110	v
23	0001 0111	ETB	55	0011 0111	7	87	0101 0111	W	119	0111 0111	w
24	0001 1000	CAN	56	0011 1000	8	88	0101 1000	X	120	0111 1000	x
25	0001 1001	EM	57	0011 1001	9	89	0101 1001	Y	121	0111 1001	y
26	0001 1010	SUB	58	0011 1010	:	90	0101 1010	Z	122	0111 1010	z
27	0001 1011	ESC	59	0011 1011	;	91	0101 1011	[	123	0111 1011	{
28	0001 1100	FS	60	0011 1100	<	92	0101 1100	\	124	0111 1100	|
29	0001 1101	GS	61	0011 1101	=	93	0101 1101	]	125	0111 1101	}
30	0001 1110	RS	62	0011 1110	>	94	0101 1110	^	126	0111 1110	~
31	0001 1111	US	63	0011 1111	?	95	0101 1111	_	127	0111 1111	DEL

Representing More Characters

- Limitation of ASCII?
  - Only supports Latin character set
  - No support for accents, additional character sets
  - Solutions?

Representing More Characters

- UTF-8
  - Another encoding scheme for characters
    - Variable length – 1, 2, 3 or 4 bytes per character
  - Compatible with ASCII
  - Consider each byte
    - Left most bit is 0? Usual ASCII Character
    - Left most bits are 110? 2 byte character
    - Left most bits are 1110? 3 byte character
    - Left most bits are 11110? 4 byte character

UTF-8 Character Examples

Q: 0101 0001

π: 1100 1111 1000 0000

©: 1100 0010 1010 1001

𝄞: 1111 0000 1001 1101 1000 0100 1001 1110

Representing Real Numbers

- Standard Representation: IEEE 754 Floating Point
  - Express the number in scientific notation
  - -0.0002589 becomes $-2.589 * 10^{-4}$
  - Encode three pieces of information

Problems with Real Numbers

- How many real numbers are there?
- How many real numbers are there between 0 and 1?
- How many values can be represented by 32 or 64 bits?
- What's the problem?

Encoding Images

- Common Techniques
  - Vector Images
  - Raster Images

Representing Colors

- How do we represent a color as a sequence of bits?

Recall

- Inside a computer:
  - Integers are represented by bits
  - Characters are represented by bits
  - Real numbers are approximated by bits
  - …
  - Without context, the bits are just data
- Adding context transforms the data into information

Where Are We Going?

- We know:
  - Information can be encoded as data
  - Computers manipulate data
  - Data can be put into context to make it information
- Next up:
  - More ways of controlling the computer so that it will manipulate data the way we want it to