CSEE 3827: Fundamentals of Computer Systems

Information Representation

Number systems: Base 10 (Decimal)

- 10 digits = {0,1,2,3,4,5,6,7,8,9}
- example: 4537.8 = $(4537.8)_{10}$

4 5 3 7 . 8

Number systems: Base 10 (Decimal)

- 10 digits = {0,1,2,3,4,5,6,7,8,9}

- example: 4537.8 = $(4537.8)_{10}$

$$\underset{10^{3}}{4} \quad \underset{10^{2}}{5} \quad \underset{10^{1}}{3} \quad \underset{10^{0}}{7} \;.\; \underset{10^{-1}}{8}$$

Number systems: Base 10 (Decimal)

- 10 digits = {0,1,2,3,4,5,6,7,8,9}
- example: 4537.8 = $(4537.8)_{10}$

4	5	3	7 .	8
$\times 10^{3}$	$\times 10^{2}$	$\times 10^{1}$	$\times 10^{0}$	$\times 10^{-1}$
4000	500	40	7	.8

Number systems: Base 10 (Decimal)

- 10 digits = {0,1,2,3,4,5,6,7,8,9}
- example: 4537.8 = $(4537.8)_{10}$

$$\begin{array}{ccccccccc} 4 & & 5 & & 3 & & 7 & . & 8 \\ \times 10^{3} & & \times 10^{2} & & \times 10^{1} & & \times 10^{0} & & \times 10^{-1} \\ \hline 4000 & + & 500 & + & 40 & + & 7 & + & .8 \end{array} = 4537.8$$

Number systems: Base 2 (Binary)

- 2 digits = {0,1}
- example: 1011.1 = $(1011.1)_2$

1 0 1 1 . 1

Number systems: Base 2 (Binary)

- 2 digits = {0,1}
- example: 1011.1 = $(1011.1)_2$

$$\begin{array}{ccccccccccc} 1 & & 0 & & 1 & & 1 & . & 1 & & \\ \times 2^{3} & & \times 2^{2} & & \times 2^{1} & & \times 2^{0} & & \times 2^{-1} & & \\ \hline 8 & + & 0 & + & 2 & + & 1 & + & .5 & = & (11.5)_{10} \end{array}$$

Number systems: Base 8 (Octal)

- 8 digits = {0,1,2,3,4,5,6,7}

- example: $(2365.2)_8$

2 3 6 5 . 2

Number systems: Base 8 (Octal)

- 8 digits = {0,1,2,3,4,5,6,7}
- example: $(2365.2)_8$

$$
\begin{array}{ccccccccc}
2 & & 3 & & 6 & & 5 & . & 2 \\
\times\ 8^{3} & & \times\ 8^{2} & & \times\ 8^{1} & & \times\ 8^{0} & & \times\ 8^{-1} \\
\hline
1024 & + & 192 & + & 48 & + & 5 & + & .25
\end{array}
= (1269.25)_{10}
$$

Number systems: Base 16 (Hexadecimal)

- 16 digits = {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}
- example: $(26BA)_{16}$ [alternate notation for hex: 0x26BA]

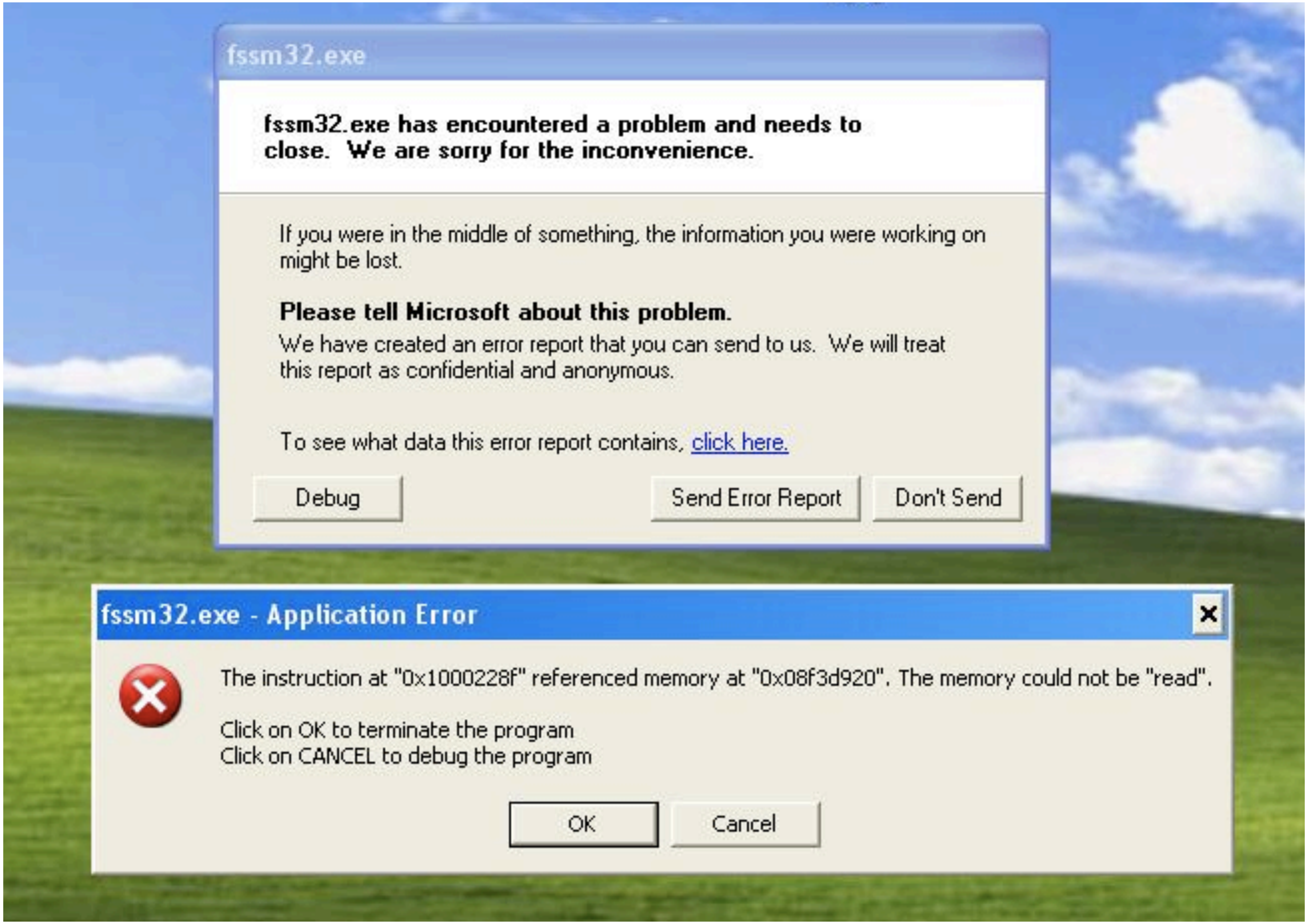
2 6 B A

Number systems: Base 16 (Hexadecimal)

- 16 digits = {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}
- example: $(26BA)_{16}$ [alternate notation for hex: 0x26BA]

$$\begin{array}{cccc} 2 & 6 & B & A \\ \times 16^{3} & \times 16^{2} & \times 16^{1} & \times 16^{0} \\ \hline 8192 + & 1536 + & 176 + & 10 \end{array} = (9914)_{10}$$

Hexadecimal (or hex) is often used for addressing



Number ranges

- Map infinite numbers onto finite representation for a computer
- How many numbers can I represent with ...

... 5 digits in decimal? 5

... 8 binary digits? 8

... 4 hexadecimal digits? 4

Number ranges

- Map infinite numbers onto finite representation for a computer
- How many numbers can I represent with ...

... 5 digits in decimal? 10^5 possible values

... 8 binary digits? 8

... 4 hexadecimal digits? 4

Number ranges

- Map infinite numbers onto finite representation for a computer
- How many numbers can I represent with ...

... 5 digits in decimal? 10^5 possible values

... 8 binary digits? 2^8 possible values

... 4 hexadecimal digits? ^{4}

Number ranges

- Map infinite numbers onto finite representation for a computer
- How many numbers can I represent with ...

... 5 digits in decimal? 10^5 possible values

... 8 binary digits? 2^8 possible values

... 4 hexadecimal digits? 16^4 possible values

Need a bigger range?

- Change the encoding.
- Floating point (used to represent very large numbers in a compact way)
  - A lot like scientific notation: 5.4×10^{5}
    exponent
    mantissa
  - Except that it is binary: $\underline{1001} \times 2^{\underline{1011}}$

What about negative numbers?

- Change the encoding.
  - Sign and magnitude
  - Ones compliment
  - Twos compliment

Sign and magnitude

- Most significant bit is sign
- Rest of bits are magnitude

$$0110 = (6)_{10} \qquad 1110 = (-6)_{10}$$

- Two representations of zero

$$0000 = (0)_{10} \qquad 1000 = (-0)_{10}$$

Ones compliment

- Compliment bits in positive value to create negative value

- Most significant bit still a sign bit

$$0110 = (6)_{10} \qquad 1001 = (-6)_{10}$$

- Two representations of zero

$$0000 = (0)_{10} \qquad 1111 = (-0)_{10}$$

Twos compliment

- Compliment bits in positive value and add 1 to create negative value
- Most significant bit still a sign bit

$$0110 = (6)_{10} \qquad 1001 + 1 = 1010 = (-6)_{10}$$

- One representation of zero

$$0000 = (0)_{10} \qquad 1000 = (-8)_{10} \qquad 1111 = (-1)_{10}$$

- One more negative number than positive

$$\text{MIN: } 1000 = (-8)_{10} \qquad \text{MAX: } 0111 = (7)_{10}$$

How about letters?

How about letters?

- Change the encoding.

TABLE 1-5
American Standard Code for Information Interchange (ASCII)

<table>
<tr><th></th><th colspan="8">$B_7B_6B_5$</th></tr>
<tr><th>$B_4B_3B_2B_1$</th><th>000</th><th>001</th><th>010</th><th>011</th><th>100</th><th>101</th><th>110</th><th>111</th></tr>
<tr><td>0000</td><td>NULL</td><td>DLE</td><td>SP</td><td>0</td><td>@</td><td>P</td><td>`</td><td>p</td></tr>
<tr><td>0001</td><td>SOH</td><td>DC1</td><td>!</td><td>1</td><td>A</td><td>Q</td><td>a</td><td>q</td></tr>
<tr><td>0010</td><td>STX</td><td>DC2</td><td>"</td><td>2</td><td>B</td><td>R</td><td>b</td><td>r</td></tr>
<tr><td>0011</td><td>ETX</td><td>DC3</td><td>#</td><td>3</td><td>C</td><td>S</td><td>c</td><td>s</td></tr>
<tr><td>0100</td><td>EOT</td><td>DC4</td><td>$</td><td>4</td><td>D</td><td>T</td><td>d</td><td>t</td></tr>
<tr><td>0101</td><td>ENQ</td><td>NAK</td><td>%</td><td>5</td><td>E</td><td>U</td><td>e</td><td>u</td></tr>
<tr><td>0110</td><td>ACK</td><td>SYN</td><td>&</td><td>6</td><td>F</td><td>V</td><td>f</td><td>v</td></tr>
<tr><td>0111</td><td>BEL</td><td>ETB</td><td>'</td><td>7</td><td>G</td><td>W</td><td>g</td><td>w</td></tr>
<tr><td>1000</td><td>BS</td><td>CAN</td><td>(</td><td>8</td><td>H</td><td>X</td><td>h</td><td>x</td></tr>
<tr><td>1001</td><td>HT</td><td>EM</td><td>)</td><td>9</td><td>I</td><td>Y</td><td>i</td><td>y</td></tr>
<tr><td>1010</td><td>LF</td><td>SUB</td><td>*</td><td>:</td><td>J</td><td>Z</td><td>j</td><td>z</td></tr>
<tr><td>1011</td><td>VT</td><td>ESC</td><td>+</td><td>;</td><td>K</td><td>[</td><td>k</td><td>{</td></tr>
<tr><td>1100</td><td>FF</td><td>FS</td><td>,</td><td><</td><td>L</td><td>\</td><td>l</td><td>|</td></tr>
<tr><td>1101</td><td>CR</td><td>GS</td><td>-</td><td>=</td><td>M</td><td>]</td><td>m</td><td>}</td></tr>
<tr><td>1110</td><td>SO</td><td>RS</td><td>.</td><td>></td><td>N</td><td>^</td><td>n</td><td>~</td></tr>
<tr><td>1111</td><td>SI</td><td>US</td><td>/</td><td>?</td><td>O</td><td>_</td><td>o</td><td>DEL</td></tr>
</table>

Gray code

Binary numeric encoding where successive numbers differ by only 1 bit

value	BCD	*# bit flips*	Gray	*# bit flips*
0	0 0 0	*3*	0 0 0	*1*
1	0 0 1	*1*	0 0 1	*1*
2	0 1 0	*2*	0 1 1	*1*
3	0 1 1	*1*	0 1 0	*1*
4	1 0 0	*3*	1 1 0	*1*
5	1 0 1	*1*	1 1 1	*1*
6	1 1 0	*2*	1 0 1	*1*
7	1 1 1	*1*	1 0 0	*1*

Some definitions

- bit = a binary digit e.g., 1 or 0
- byte = 8 bits e.g., 01100100
- word = a group of bytes

  a 16-bit word = 2 bytes e.g., 1001110111000101

  a 32-bit word = 4 bytes e.g., 10011101110001010111011100010 1