Endianness
Activities

This Week
- Lab 9 – Network Programming

Next Week (THURS)
- Start MIPS Assembly Programming
  (lecture for 1+ day)

Assignments Due

Lab 9
- Due by NOV 9th 5:00am
Endianness

- In typical computer memory, each address (location) stores one byte
- If we have a one-byte integer, how is that stored in memory?
- If we have a two-byte integer, how is that stored in memory?
- If we have a four-byte integer, how is that stored in memory?

Endianness = Byte Ordering
Endianness Example

- 32-bit hexadecimal number
  0x12345678

- Composed of 4 bytes:
  0x12 0x34 0x56 0x78
  (MSB) (LSB)

- Two possible arrangements:

<table>
<thead>
<tr>
<th>Address</th>
<th>“Option A”</th>
<th>“Option B”</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0x12</td>
<td>0x78</td>
</tr>
<tr>
<td>1</td>
<td>0x34</td>
<td>0x56</td>
</tr>
<tr>
<td>2</td>
<td>0x56</td>
<td>0x34</td>
</tr>
<tr>
<td>3</td>
<td>0x78</td>
<td>0x12</td>
</tr>
</tbody>
</table>
32-bit hexadecimal number
0x12345678

Composed of 4 bytes:
0x12 0x34 0x56 0x78

Two possible arrangements:
- Big Endian
- Little Endian

<table>
<thead>
<tr>
<th>Address</th>
<th>Big Endian (MSB)</th>
<th>Little Endian (LSB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0x12</td>
<td>0x78</td>
</tr>
<tr>
<td>1</td>
<td>0x34</td>
<td>0x56</td>
</tr>
<tr>
<td>2</td>
<td>0x56</td>
<td>0x34</td>
</tr>
<tr>
<td>3</td>
<td>0x78</td>
<td>0x12</td>
</tr>
</tbody>
</table>
How is $\text{DEADBEEF}_{16}$ stored in little and big endian formats at address $21C_{16}$?

**Little endian**
- $21C_{16} = \text{EF}_{16}$
- $21D_{16} = \text{BE}_{16}$
- $21E_{16} = \text{AD}_{16}$
- $21F_{16} = \text{DE}_{16}$

**Big endian**
- $21C_{16} = \text{DE}_{16}$
- $21D_{16} = \text{AD}_{16}$
- $21E_{16} = \text{BE}_{16}$
- $21F_{16} = \text{EF}_{16}$
Big Endian – vs – Little Endian

**Big-Endian CPU**

- Most significant byte (MSB) comes first (stored in lower memory address)

- Examples
  - Motorola 68000
  - Java virtual machine
  - IBM PowerPC (by default, can also be little endian)

**Little-Endian CPU**

- Least significant byte (LSB) comes first (stored in lower memory addresses)

- Examples
  - Intel x86/x86-64
  - DEC Alpha
  - ARM (by default, also can be big endian)
When do I need to care that some computers are big-endian and others are little endian?

What happens if I open big-endian data on a little-endian computer?

Endianness must be considered whenever you are sharing data between different computer systems

Reading/writing data files to disk

Reading/writing data files to network
Best Practices

- **Pick one format and stick with it!**
  - Example: Data sent over the network will always be in *big-endian* format regardless of who sends it
  - *Networks are big-endian “by tradition”*
  - Example: Data written to disk will always be in *little-endian* format regardless of who writes it

- **Convert between data storage/transfer format and internal representation as needed**
  - Example: Little-endian machines convert to big-endian before sending data onto the network (and convert back upon receiving data from the network)
### Examples in Industry

#### Table 2- Common file formats

<table>
<thead>
<tr>
<th>Little-Endian Format</th>
<th>Big-Endian Format</th>
<th>Variable or Bi-Endian Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMP</td>
<td>PSD</td>
<td>DXF (AutoCAD*)</td>
</tr>
<tr>
<td>GIF</td>
<td>IMG</td>
<td>PS (Postscript*, 8 bit</td>
</tr>
<tr>
<td>FLI</td>
<td>JPEG, JPG</td>
<td>interpreted text, no</td>
</tr>
<tr>
<td>PCX</td>
<td>MacPaint</td>
<td>Endian issue)</td>
</tr>
<tr>
<td>QTM</td>
<td>SGI</td>
<td>POV (Persistence of</td>
</tr>
<tr>
<td>RTF</td>
<td>Sun Raster</td>
<td>Visionraytracer*)</td>
</tr>
<tr>
<td></td>
<td>WPG</td>
<td>RIFF (WAV &amp; AVI*)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TIFF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XWD (X Window Dump*)</td>
</tr>
</tbody>
</table>

#### Bus Protocols

<table>
<thead>
<tr>
<th>Infiniband</th>
<th>PCI Express</th>
<th>PCI-32/PCI-64</th>
<th>USB</th>
</tr>
</thead>
</table>

#### Network Protocols

<table>
<thead>
<tr>
<th>TCP/IP</th>
<th>UDP</th>
</tr>
</thead>
</table>

#### Bus Protocols

| GMII | (8 bit wide bus, no Endian issue) |