More on sockets

Nick DeMarinis
• Snowcast milestone due 9/20 by 11:59pm EDT
  – “Lab 1”: get started with implementation
  – Design of server
• Grading server: we’re still working on testing!
Administrivia

• **Office hours**
  – Collaborative hours: come to room, TAs (or I) will circulate
  – Individual: join queue on Hours platform, talk in-person or on Zoom

• **Waitlist**
  – If you have a pending override code, you’re safe
  – Next batch: Tomorrow morning
WHAT IS NETWORK BYTE ORDER

IF WE LOOK AT THE MESSAGE AS A BYTE ARRAY IT WILL LOOK LIKE THIS

INDEX 0 1 2 3

Y Dot 1 TYPE NUMBER

WHEN A FIELD LIKE NUMBER IS 1 BYTE WE HAVE A CHOICE ON HOW TO ARRANGE THE BYTES

Ex NUMBER 00H 00H DD NUMBLY BIG ENDIAN GEO DBE BE DI NETWORK BYTE ORDER

LITTLE ENDIAN OLD DBE IN

DIFFERENT SYSTEMS MAY USE DIFFERENT CONVENTIONS SO NETWORK PROTOCOLS NEED TO STANDARDIZE

CLIENT

CONNECT (SERVER, PORT)

SERVER

LISTEN ON PORT 8888

ACCEPT NEW CONNECTIONS

KERNEL MAKES NEW SOCKET FOR THIS CONNECTION!
A simple example program for working with sockets:

**Guessing game**

**Client**
- **Guess number**

**Server**
- Picks a random number

**Response**
- Too high
- Too low
- Correct
DATA FORMAT

As protocol designer, we could make up any format we want (strings, JSON, HTTP, ...). Here let's make a binary format:

Each message is 5 bytes:

<table>
<thead>
<tr>
<th>1 byte</th>
<th>Message Type</th>
<th>UINT8</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 bytes</td>
<td>Number</td>
<td>INT32</td>
</tr>
</tbody>
</table>

If Guess:
- TYPE = 0
- NUMBER = the user's guess

If Response:
- TYPE = 1
- NUMBER =
  - -1 too low
  - 0 correct
  - 1 too high

Data sent in network byte order (like most network protocols)
**What is Network Byte Order?**

If we look at the message as a byte array, it will look like this:

<table>
<thead>
<tr>
<th>Index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Type**
- **Number**

When a field like number is > 1 byte, we have a choice on how to arrange the bytes:

Ex. Number = 0xAAABCCDD

- **Big Endian** (i.e., Network Byte Order)
  - 0x00
  - AA
  - BB
  - CC
  - DD

- **Little Endian**
  - 0x00
  - DD
  - CC
  - BB
  - AA

Different systems may use different conventions, so network protocols need to standardize!