Naming

Content taken from the following:
“Distributed Systems: Principles and Paradigms” by Andrew S. Tanenbaum and Maarten Van Steen (Chapter 4)
Various online sources (including openclipart.org)
Naming

• "What's in a name?"
  
  - "That which we call a .com by any other TLD would load just as quickly."

“There are only two hard things in Computer Science: cache invalidation and naming things.”

- Phil Karlton (Netscape)
Trivia

• What is Netscape?
  - A. A web browser
  - B. A web directory
  - C. An internet service provider
  - D. A brand name
  - E. All of the above
  - F. None of the above
Addressing

- Concept of an *entity* and its *name* vs. its *address*
- Some names are true *identifiers*
  - Each identifier refers to at most one entity
  - Each entity is referred to by at most one identifier
  - Identifiers are never re-used at another time
- Name-to-address *binding*
  - *Name space*: domain of all possible names
  - Static vs. dynamic
  - Central vs. decentralized
    - *Name server*: central host responsible for maintaining bindings
Addressing

• Which of the following is the most decentralized name binding?
  - A. Mailing addresses
  - B. Discord server nicknames
  - C. Subreddits
  - D. Human nicknames
  - E. Xbox gamertags
Naming schemes

/         bin/
  eb40af8e  bash  (444, Molloy)
  c6c1904c  etc/
  0eced3a3e  passwd  (445, Sprague)
  28dec8ba  usr/
  4b6683e7  bin/
  88c9618b  nano  (456, Weikle)
  3566223f  vim  (458, Heydari)
  38b22b10  lib/
  (470, Lam)

Flat     Structured     Attribute-based
Flat naming

- Identifiers contain no location information
- Various lookup approaches
  - Broadcast / multicast
  - Forwarding pointers
  - Proximity routing
- Examples: ARP, Chord

https://geek-university.com/address-resolution-protocol-arp/
**Distributed hash tables**

- **Chord** uses an $m$-bit identifier space and modulo arithmetic
  - Key $k$ is stored at the node with the smallest id $\geq k$
- Each node maintains a **finger table** of forward shortcuts
- To look up $k$, repeatedly follow lookups in finger table
  - Goal: halve distance to destination every hop
Addressing

- Which of the following is the maximum size of the finger table for a 256-node Chord network?
  - A. 0
  - B. 1
  - C. 8
  - D. 32
  - E. 128
Structured naming

• Root vs. interior vs. leaf nodes
• Absolute vs. relative names
  – Global vs. local names
• Iterative vs. recursive resolution
• Linking and aliasing
  – Hard vs. soft (symbolic) links
• Mounting and mount points
• Examples: file systems, DNS, NFS

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>Size</th>
<th>Used</th>
<th>Avail</th>
<th>Use%</th>
<th>Mounted on</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/mapper/rhel_login01-root</td>
<td>50G</td>
<td>23G</td>
<td>28G</td>
<td>46%</td>
<td>/</td>
</tr>
<tr>
<td>/dev/sda6</td>
<td>497M</td>
<td>206M</td>
<td>292M</td>
<td>42%</td>
<td>/boot</td>
</tr>
<tr>
<td>nfs.cluster.cs.jmu.edu:/nfs/home</td>
<td>100G</td>
<td>4.6G</td>
<td>96G</td>
<td>5%</td>
<td>/nfs/home</td>
</tr>
<tr>
<td>nfs.cluster.cs.jmu.edu:/nfs/scratch</td>
<td>2.0T</td>
<td>862G</td>
<td>1.2T</td>
<td>43%</td>
<td>/scratch</td>
</tr>
</tbody>
</table>
Which of the following is an example of a **structured** (as opposed to **flat**) name binding?

- A. Mailing addresses
- B. Discord server nicknames
- C. Subreddits
- D. Human nicknames
- E. Xbox gamertags
IPv4

- IPv4: 32 bits - four octets w/ CIDR notation (/8, /16, etc.)
  - Classful addressing: Class A, Class B, Class C
  - IETF and IANA allocate addresses (32 bits - 4 billion total addresses)
  - Published in 1981; now nearly exhausted

- Notable networks
  - Private (10.0.0.0/8)
  - Loopback (127.0.0.0/8)
  - JMU (134.126.0.0/16)
  - Private (192.168.0.0/16)

An IPv4 address (dotted-decimal notation)

```
  172  .  16  .  254  .  1
  ↓    ↓    ↓    ↓
10101100 .00010000.11111110.00000001
```

One byte = Eight bits

Thirty-two bits (4 x 8), or 4 bytes

from https://en.wikipedia.org/wiki/IPv4

https://xkcd.com/195/
• What is the total number of addresses in IPv4?
  – A. $2^8$
  – B. $2^{16}$
  – C. $2^{32}$
  – D. $2^{64}$
  – E. $2^{128}$
IPv6 published in 1998
- 128 bits - $3.4 \times 10^{38}$ total addresses
- Eight groups of 16 bits (4 hex chars)
- 64-bit routing prefix, 64-bit host/interface identifier

An IPv6 address (in hexadecimal)

```
2001:0DB8:AC10:FE01:0000:0000:0000:0000
```

Zeroes can be omitted

0010000000000000:0000110110111000:10101100001000:1111111000000001:
0000000000000000:0000000000000000:0000000000000000:0000000000000000

from https://en.wikipedia.org/wiki/IPv6
• What is the total number of addresses in IPv6?
  - A. $2^8$
  - B. $2^{16}$
  - C. $2^{32}$
  - D. $2^{64}$
  - E. $2^{128}$
• The IPv6 name space is far larger than you think!
  - In fact, there is NO WAY to draw the two address spaces to scale. If IPv4 were a 1.6-inch square, IPv6 would be a square the size of the solar system!
  - $2^{128} \approx 10^{38} \gg$ the number of drops of water in all the world’s oceans $(10^{25})$ or the number of stars in the observable universe $(10^{23})$
  - “If we had been assigning IPv6 addresses at a rate of 1 billion per second since the earth was formed, we would have by now used up less than one trillionth of the address space.”
  - “We could assign an IPv6 address to every atom on the surface of the earth – and have enough addresses left over for another hundred earths.”

Sources:
• http://waitbutwhy.com/2014/11/1000000-grahams-number.html
• http://www.tcpipguide.com/free/t_IPv6AddressSizeandAddressSpace-2.htm
• http://www.brucebnews.com/2010/10/ipv6-and-really-large-numbers/
Why haven’t we transitioned?

• Advantages
  – Solves IP naming problem pseudo-permanently
  – Deals with explosion of Internet of Things (IoT) devices requiring an address
  – Increasing cost to acquire IPv4 addresses

• Obstacles
  – Network Address Translation (NAT) allows multiple hosts to use a single public IP address
  – IPv4 blocks have become more “fluid”
  – Lack of expertise managing multi-protocol networks

Informed by discussion with Mike Ripley, JMU Information Technology
Attribute-based naming

• Human-friendly resource identifiers
• Storage of (key, value) pairs
• Often implemented with distributed hash tables
  - Centralized vs. decentralized lookups
  - You implemented this in P3!
• Semantic overlay networks
  - Nodes maintain explicit links to "semantically proximate" nodes
  - Most useful in distributed peer-to-peer networks
  - Exploit small-world effect
Which of the following is the best example of a semantic overlay network?

- A. Mailing addresses
- B. Discord server nicknames
- C. Subreddits
- D. Human nicknames
- E. Xbox gamertags