6.033 Spring 2020

Lecture #2

• Naming in systems
• Case study: DNS
Last Time: Enforced Modularity via Client/Server Model

Client <-> internet <-> Server

load(kaws.com/buy.html?phone)

state on server

client | UID | reply

replay results from table

6.033 | spring 2020 | lacurts@mit.edu
Last Time: Enforced Modularity via Client/Server Model

Client \[\leftrightarrow\] internet \[\leftrightarrow\] Server

\text{load(\texttt{kaws.com/buy.html?phone})}

state on server

\begin{tabular}{|c|c|c|}
\hline
client & UID & reply \\
\hline
6.033 & spring 2020 & lacurts@mit.edu \\
\hline
\end{tabular}

replay results from table
Last Time: Enforced Modularity via Client/Server Model

Client  ---internet---  Server

load(kaws.com/buy.html?phone)

state on server
client | UID | reply

replay results from table
Last Time: Enforced Modularity via Client/Server Model

**Today: Naming**

allows modules to interact
Examples of Names

mit.edu
lacurts@mit.edu
lacurts
R0
main
WebBrowser
/mit/6.033/www/schedule.shtml
http://web.mit.edu/about
617-253-7341
128.30.2.121

hostname
e-mail
username
x86 register name
function name
class name
path name
URL
phone number
IP Address
why use names?
why use names?
Naming Schemes
Naming Schemes

1. Set of all possible names
Naming Schemes

1. Set of all possible names

2. Set of all possible values
Naming Schemes

1. Set of all possible *names*

2. Set of all possible *values*

3. **Look-up algorithm** to translate a name into a value (or set of values, or “none”)
Domain Name System
Domain Name System

1. **names**: hostnames (**www.mit.edu**)
Domain Name System

1. **names**: hostnames (www.mit.edu)

2. **values**: IP addresses (18.9.22.69)
Domain Name System

1. **names**: hostnames (**www.mit.edu**)

2. **values**: IP addresses (**18.9.22.69**)

   IP addresses are imbued with location information: routers can send packets to an IP address, but not to a hostname.
Domain Name System

1. **names:** hostnames (**www.mit.edu**) 

2. **values:** IP addresses (**18.9.22.69**) 
   
   IP addresses are imbued with location information: routers can send packets to an IP address, but not to a hostname.

3. **look-up algorithm:** resolves a hostname to an IP address so that your machine knows where to send data.
DNS Hierarchy
(a partial view)
DNS Look-up for web.mit.edu

query to:  

result:  

6.033 | spring 2020 | lacurts@mit.edu
DNS Look-up for web.mit.edu

query to: 198.41.0.4

result:

```
com
 ├─ berkeley
  │   └─ eecs
  │         └─ eecs
  ├── edu
  │   └─ mit
  │       └─ web
  │           └─ web
  └─ root
```
query to: 198.41.0.4  
result:
DNS Look-up for web.mit.edu

query to: 198.41.0.4

result:

198.41.0.4  root

com  

edu  root table  com. 192.5.6.30  edu. 192.41.162.30

berkeley

mit

eecs  web  www
DNS Look-up for web.mit.edu

query to: 198.41.0.4

result:

198.41.0.4 root

com

edu

root table

com. 192.5.6.30
edu. 192.41.162.30

berkeley

mit

eecs web www
DNS Look-up for web.mit.edu

query to: 198.41.0.4
result: edu. 192.41.162.30
DNS Look-up for web.mit.edu

query to: 198.41.0.4          result: edu. 192.41.162.30
DNS Look-up for web.mit.edu

query to: 192.41.162.30  
result:

198.41.0.4 → root

com 192.41.162.30 → edu 192.41.162.30

root table

edu 192.41.162.30 → com 192.5.6.30

result:

192.41.162.30
DNS Look-up for web.mit.edu

query to: 192.41.162.30  result:

198.41.0.4

root

com 192.41.162.30

edu

edu table

mit.edu. 18.72.0.3
berkeley.edu. 128.32.136.14

root table

com. 192.5.6.30
edu. 192.41.162.30

192.41.162.30

www
web

berkeley

192.41.162.30
DNS Look-up for web.mit.edu

query to: 192.41.162.30  
result:

198.41.0.4  
com 192.41.162.30  
edu  
berkeley  

root  
root table  
com. 192.5.6.30  
edu. 192.41.162.30
mit.edu. 18.72.0.3  
berkeley.edu. 128.32.136.14

mit ede

www  
web  
eecs
DNS Look-up for web.mit.edu

query to: 192.41.162.30    result: mit.edu. 18.72.0.3
DNS Look-up for web.mit.edu

query to: 192.41.162.30  
result: mit.edu. 18.72.0.3
DNS Look-up for web.mit.edu

query to: 18.72.0.3

result:
DNS Look-up for web.mit.edu

query to: 18.72.0.3

result:

com.	192.5.6.30
edu.	192.41.162.30

root table

mit.edu.	18.72.0.3
berkeley.edu.	128.32.136.14

edu table

mit.

mit table

web.mit.edu.	18.9.2.69
www.mit.edu.	18.9.22.169
eecs.mit.edu.	18.62.1.6
DNS Look-up for web.mit.edu

query to: 18.72.0.3

result:

query to: 18.72.0.3

result:
DNS Look-up for web.mit.edu

query to: 18.72.0.3

result: web.mit.edu. 18.9.2.69
DNS Look-up for web.mit.edu

query to: 18.72.0.3  
result: web.mit.edu. 18.9.2.69
DNS Hierarchy
(a partial view)
• **Modularity** (and abstraction) limit complexity. One way to enforce modularity is to use a client/server design.

• **Naming** is what allows modules — for example, a client and a server — to communicate; it is pervasive across systems.

• **DNS** maps hostnames to IP addresses. It is also a good example of **hierarchy**.
Lingering Problem

Client ↔ internet ↔ Server

load(kaws/buy.html?phone)
what if we don’t want our modules to be on entirely separate machines? how can we enforce modularity on a single machine?