

# CS354: Machine Organization and Programming

Lecture 40  
Monday the December 7<sup>th</sup> 2015

Section 2  
Instructor: Leo Arulraj

© 2015 Karen Smoler Miller  
© Some examples, diagrams from the CSAPP text by Bryant and O'Hallaron

## Section 2 Class Announcements

Practice Final Exam has been posted.  
Good luck with the finals.

Wednesday's (12/09) lecture by Adalbert (Instructor for next semester).

Friday's (12/11) and Monday's (12/14) lectures by Jason (Instructor for Section 1).

**Course evaluation:** Login to <https://aefis.wisc.edu> using Net ID and password between Dec 8<sup>th</sup> through Dec 16<sup>th</sup> to give course feedback.

## Section 2 Lecture Overview

1. Internet Domain Names
2. OSI Model
3. Internet connection
4. Web Basics
5. HTTP Transactions

The following will be handled by Adalbert on Wednesday's lecture.

1. Sockets Interface
2. Echo Client, Server
3. Overview of Tiny Web Server

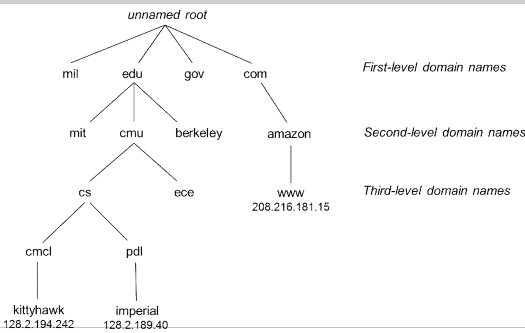
## Internet Domain Names

Since large integers are difficult to remember, a human friendly *domain name* is used along with a mechanism that maps domain names to IP addresses.

A domain name is a sequence of words separated by periods. E.g. cs.wisc.edu

The set of domain names form a hierarchy and each domain name encodes its position in the hierarchy.

## Subset of the Internet Domain Name Hierarchy



## Internet Domain Names

First level domain names defined by a nonprofit organization called ICANN (Internet Corporation for Assigned Names and Numbers).

Second level domain names are assigned on a first come first server basis by authorized agents of ICANN.

The mapping between domain names and the set of IP addresses is maintained in a distributed world-wide database known as DNS (Domain Name System).

## Internet Domain Names

- The Internet maintains a mapping between IP addresses and domain names in a huge worldwide distributed database called **DNS**
- Conceptually, programmers can view the DNS database as a collection of millions of *host entries*.
  - Each host entry defines the mapping between a set of domain names and IP addresses.
  - In a mathematical sense, a host entry is an equivalence class of domain names and IP addresses.

## Internet Domain Names

Each Internet host has the locally defined domain name *localhost* which always maps to the loopback address *127.0.0.1*

Example program that displays the host entry for a input hostname or dotted decimal address.

```
linux> ./hostinfo cs.wisc.edu
```

## Internet Connections

Internet clients and servers communicate by sending and receiving streams of bytes over connections.

A connections is:

- **point-to-point**: connects pair of processes
- **full-duplex**: data can flow in both directions
- **reliable**: except for some catastrophic failures, the stream of bytes sent by the source is eventually received by the destination in the same order that it was sent.

## Internet Connections

A socket is an endpoint of a connection and has a socket address that is denoted by “address:port” consisting of:

1. An Internet address
2. 16 bit integer port

The port in the server’s socket is typically a *well-known port* that is associated with the service.

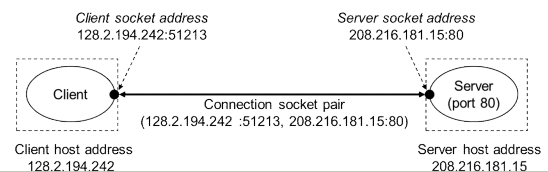
The port in the client’s socket is *ephemeral port* and is assigned automatically by the kernel when the client makes a connection request.

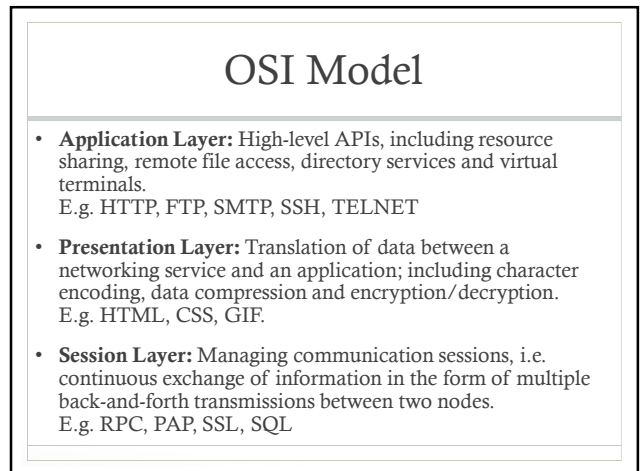
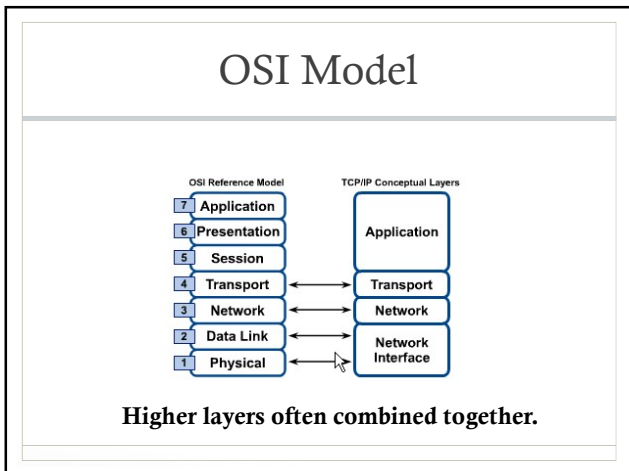
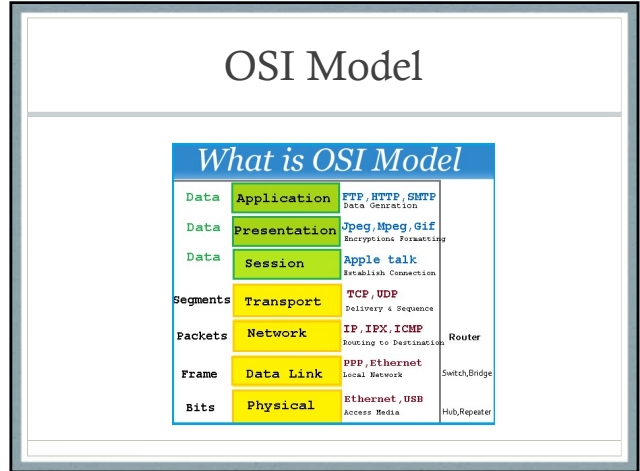
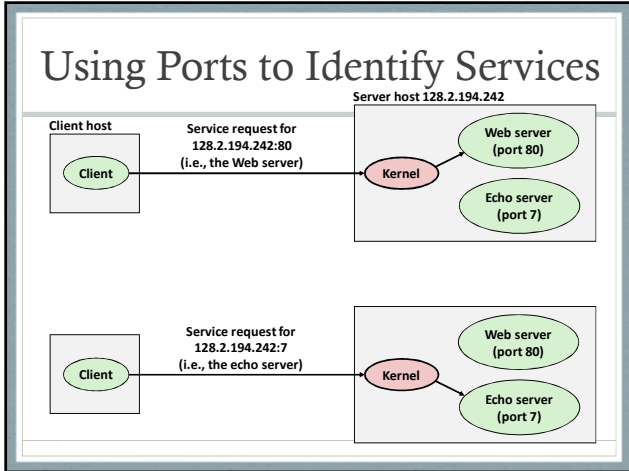
## Internet Connections

- Popular services have permanently assigned *well-known ports* and corresponding *well-known service names*:
  - echo server: 7/echo
  - ssh servers: 22/ssh
  - email server: 25/smtp
  - Web servers: 80/http
- Mappings between well-known ports and service names is contained in the file `/etc/services` on each Linux machine.

## Internet Connection

A connection is uniquely identified by the socket addresses of its two end-points.





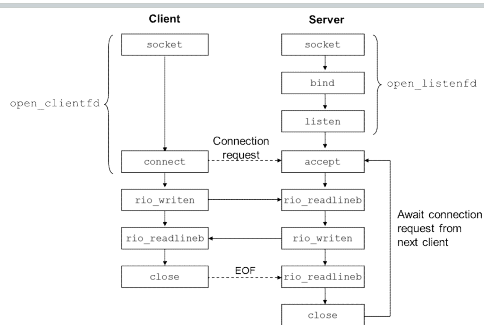
## OSI Model

- **Transport Layer:** Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing.  
E.g. TCP, UDP, NETBEUI  
Data Unit: Segments/Datagram
- **Network Layer:** Structuring and managing a multi-node network, including addressing, routing and traffic control.  
E.g. IPv4, IPv6, IPsec, AppleTalk, ICMP  
Data Unit: Packet

## OSI Model

- **Data Link Layer:** Reliable transmission of data frames between two nodes connected by a physical layer.  
E.g. PPP, IEEE 802.2, L2TP, MAC, LLDP  
Data Unit: Frame
- **Physical Layer:** Transmission and reception of raw bit streams over a physical medium Ethernet physical layer.  
E.g. DSL, USB, ISDN, DOCSIS  
Data Unit: Bit

## Overview of the sockets interface



## Web Basics

Client server interact through text based application level protocol known as HTTP (Hypertext Transfer Protocol)

Web content written in HTML language.  
(Hypertext Markup Language)

HTML contains instructions on how to display content on browser. Eg. `<b> Make me bold </b>`

HTML pages can contain pointers (hyperlinks) to content stored on any Internet host.

## Web Content

Web content is a sequence of bytes with an associated MIME type.

MIME -> Multipurpose Internet Mail Extensions

E.g. text/html -> HTML page.

text/plain -> unformatted text.

image/gif -> Binary image encoded in GIF format.

Each file has a unique name known as a URL (Universal Resource Locator)

E.g. <http://pages.cs.wisc.edu/~cs354-1/index.html>

## Web Content

Two types of web content:

**Static:** Fetch a disk file and return its contents to client.

- Fastest and most efficient way to deliver content
- More secure because no executable

**Dynamic:** Run an executable file and return its output to client.

- Suitable for displaying user specific content

## HTTP Requests

**HTTP Request** is a request line followed by zero or more request headers followed by empty text line.

Request line is of the form: <method> <uri> <version>

HTTP support many different methods: GET, POST, OPTIONS, HEAD, PUT, DELETE, TRACE

Our focus: GET method which instructs server to generate and return the content identified by the URI (Uniform resource Identifier)

URI is the suffix of the URL that contains the file name and optional arguments.

Version field specifies which HTTP version to which request conforms.

## HTTP Requests

Request headers provide additional information to the server such as:

Brand name of the browser

MIME types that the browser understands

Request headers are of the form:

<header name>: <header data>

E.g. Host: [www.aol.com](http://www.aol.com)

## HTTP Responses

**Http response** is a response line followed by zero or more response headers, followed by an empty line that terminates the headers, followed by the response body.

Response line is of the form: <version> <status code> <status message>

Status code is a three digit positive integer and some common status codes are:

200 -> OK, 404 -> Not found, 505 -> HTTP version not supported

Status message gives the English equivalent of the status code.

Common Response headers are: Content-Type: text/html, Content-Length: 42092

## Serving dynamic content

### Passing arguments to the server:

“?” separates file name and arguments.

“&” separates multiple arguments.

E.g. Http request: GET /cgi-bin/adder?150&213 HTTP/1.1

### Passing arguments to child process by Web Server:

Using environment variables.

Child sets environment variable QUERY\_STRING to “150&213” before calling execve.

Child also perform output redirection using dup2 to capture the results.