

# Distributed computing

---

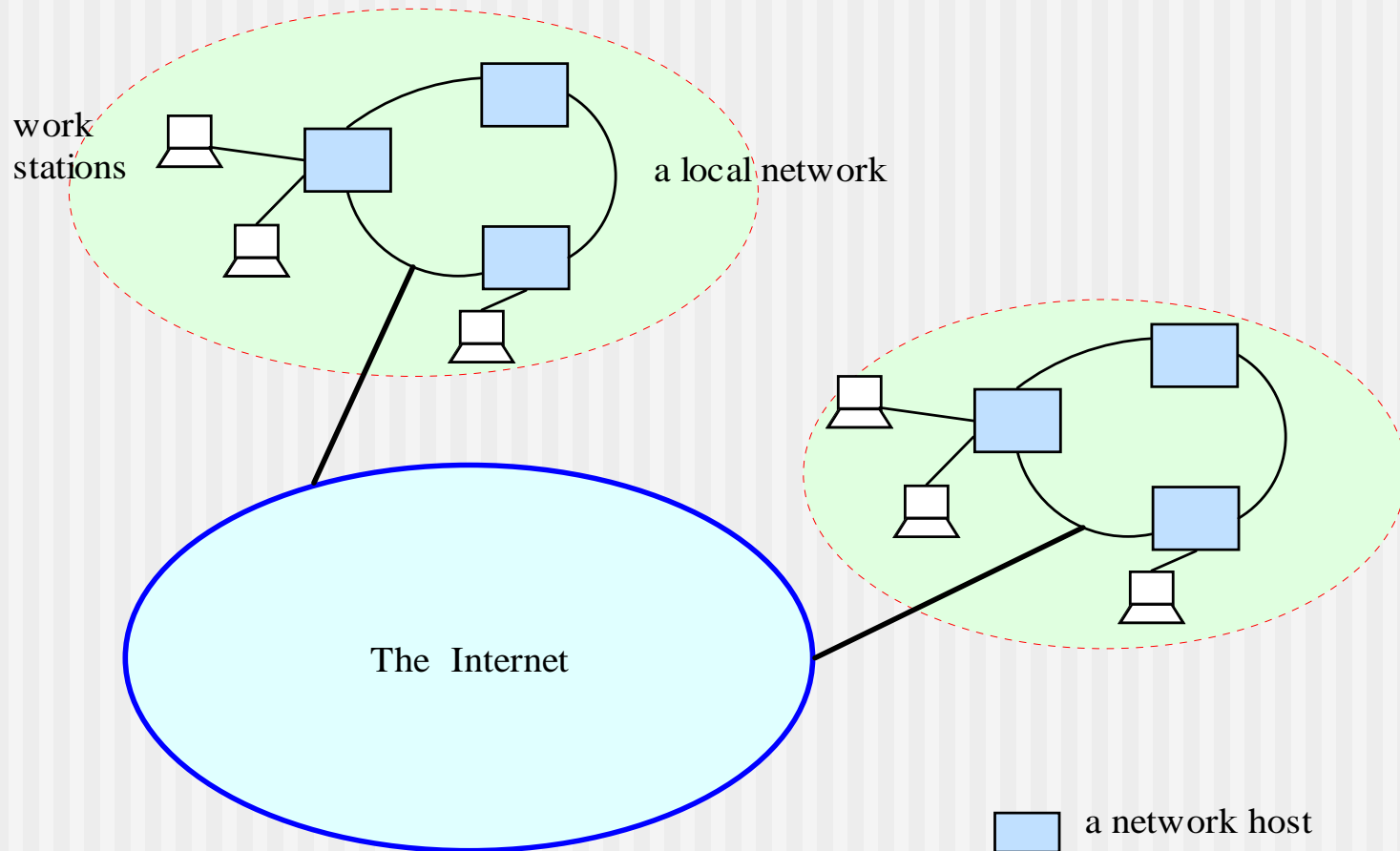
Mei-Ling Liu

# Distributed system, distributed computing

---

- Early computing was performed on a single processor. Uni-processor computing can be called *centralized computing*.
- A *distributed system* is a collection of independent computers, interconnected via a network, capable of collaborating on a task.
- *Distributed computing* is computing performed in a distributed system.

# Distributed Systems



# Examples of Distributed systems

---

- Network of workstations (NOW): a group of networked personal workstations connected to one or more server machines.
- The Internet
- An intranet: a network of computers and workstations within an organization, segregated from the Internet via a protective device (a firewall).

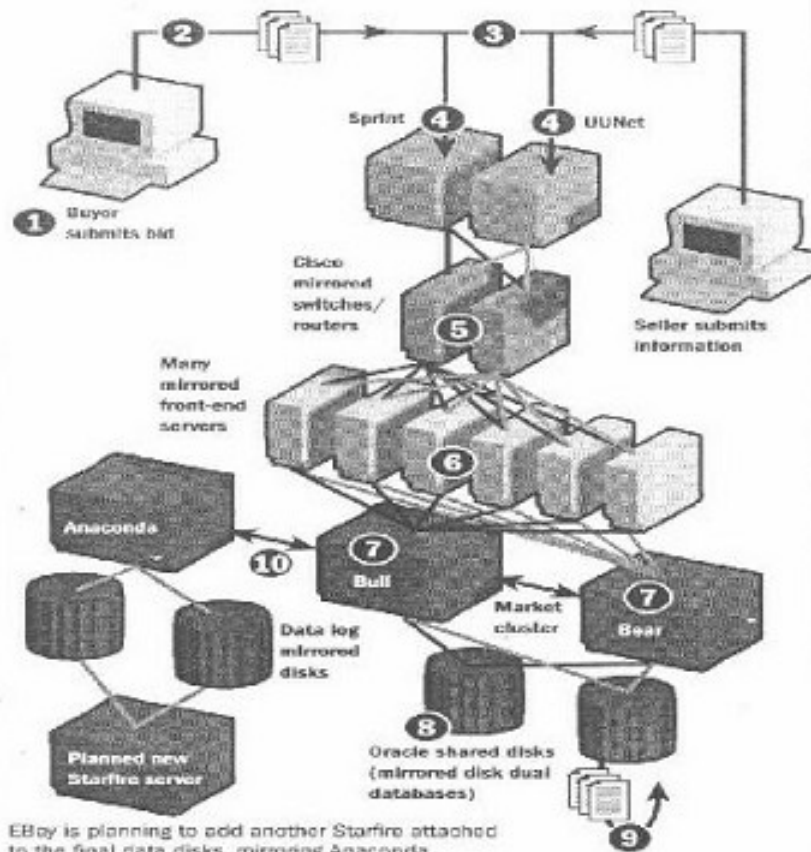
# Example of a large-scale distributed system – eBay (Source: Los Angeles Times.)

## Where It Goes

EBay users rarely think about the bidding process—until the site crashes. Behind the scenes, the online auctioneer has a number of safeguards that rely increasingly on duplicated, or mirrored, technologies in case one piece of machinery or software fails. But the information must still pass through many different companies and types of equipment for everything to work properly.

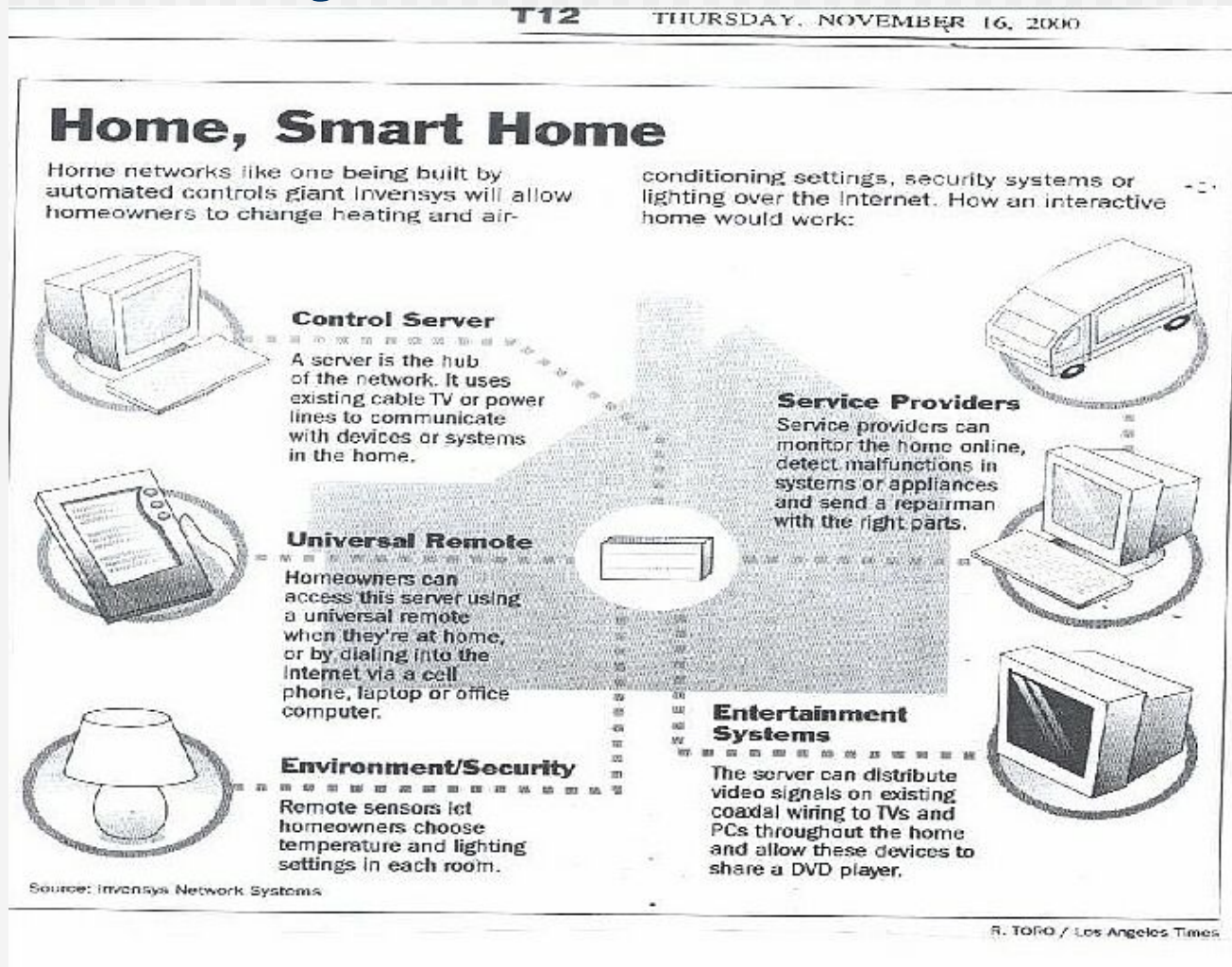
- 1** Bidder at home registers and submits an electronic bid from a personal computer.
- 2** The bid travels from the consumer's Internet service provider, through switches and routers, to the ISP company's servers.
- 3** The bid is sent through the Internet backbone.
- 4** The bid travels to one of EBay's ISPs, most likely Sprint or UUNet, and through pipes to EBay.
- 5** The bid passes through EBay's Cisco switches and routers.
- 6** The information reaches one of about 200 front-line Compaq servers running on Windows NT. The servers are mirrored, so that if any one fails, the others pick up the slack.
- 7** The bid is passed along to one of Sun Microsystems Starfire servers, named Bull and Bear, that mirror each other.
- 8** The bid is added to two information-storage databases running Oracle software, where it is matched with the seller's information.
- 9** The information flow is reversed back out of EBay, into e-mails sent to both the seller and potential buyers who are outbid. Confirmation is also sent to the bidder.
- 10** From Bull, the bid amount and other details are sent to another Starfire server, called Anaconda, and recorded on mirrored data storage disks.

Source: Times Staff Writer



# An example small-scale distributed system

(Source: Los Angeles Times.)



# Computers in a Distributed System

---

- Workstations: computers used by end-users to perform computing
- Server machines: computers which provide resources and services
- Personal Assistance Devices: handheld computers connected to the system via a wireless communication link.



# The power of the Internet

(Source: the Usability Professional Association's site.)

- 60 million American households use computers. (*The New York Times*, 5/28/98)
- The number of computer users in the workplace has increased from 600,000 in 1976 to 80 million today. (*San Francisco Examiner*, 3/29/98)
- 84% of Internet users say that the Web is indispensable. Nearly the same percentage find e-mail indispensable. 85% use the Internet every day. (*GVU*, 1997)



# The Power of the Internet – 2

(Source: [www.cisco.com](http://www.cisco.com))

- **BACKBONE CAPACITY:** The capacity of the Internet backbone to carry information is doubling every 100 days. ([U.S. Internet Council](#), Apr. 1999).
- **DATA TRAFFIC SURPASSING VOICE:** Voice traffic is growing at 10% per year or less, while data traffic is conservatively estimated to be growing at 125% per year, meaning voice will be less than 1% of the total traffic by 2007. ([Technology Futures, Inc](#) March 2000).
- **DOMAIN NAMES:** There are 12,844,877 unique domain names (e.g. Cisco.com) registered worldwide, with 428,023 new domain names registered each week. ([NetNames Statistics](#) 12/28/1999).

# The Power of the Internet – 3

(Source: [www.cisco.com](http://www.cisco.com))

- DOMAIN NAMES: There are 12,844,877 unique domain names (e.g. Cisco.com) registered worldwide, with 428,023 new domain names registered each week. ([NetNames Statistics](#) 12/28/1999).
- HOST COMPUTERS: In July 1999 there were 56.2 million "host" computers supporting web pages. In July 1997 there were 19.5 million host computers, with 3.2 million hosts in July 1994, and a mere 80,000 in July 1989. ([Internet Software Consortium – Internet Domain Survey](#)).
- TOTAL AMOUNT OF DATA: 1,570,000,000 pages, 29,400,000,000,000 bytes of text, 353,000,000 images, and 5,880,000,000,000 bytes of image data. ([The Censorware Project](#), Jan. 26, 1999).

# The Power of the Internet – 4

(Source: [www.cisco.com](http://www.cisco.com))

- EMAIL VOLUME: Average U.S. consumer will receive 1,600 commercial email messages in 2005, up from 40 in 1999, while non-marketing and personal correspondence will more than double from approximately 1,750 emails per year in 1999 to almost 4,000 in 2005 ([Jupiter Communications](#), May 2000).
- 159 million computers in the U.S., 135 million in EU, and 116 million in Asia Pacific (as of April 2000).
- WEB HITS/DAY: U.S. web pages averaged one billion hits per day (aggregate) in October 1999. (eMarketer/Media Metrix, Nov. 1999).

# NUMBER OF AMERICANS ONLINE – HISTORICAL

(Source: [www.cisco.com](http://www.cisco.com))

- 1993 – 90,000 ([U.S. Internet Council](#), Apr. 1999).
- 1997 – 19 million ([Stratis Group](#), Apr. 1999).
- 1998 – 68 million in 1998. ([Strategis Group](#), Nov. 1999).
- 1998 – 84 million from home or work ([Stratis Group](#), Apr. 1999).
- 1998 – 37 million DAILY ([Stratis Group](#), Apr. 1999).
- 1999, Nov. – 118.4 million ([Cyberatlas/Nielsen Net Ratings](#), Dec. 1999).
- 1999, Nov. – 74 million actually went online ([Cyberatlas/Nielsen Net Ratings](#), Dec. 1999).

# PERCENTAGE OF AMERICANS ONLINE

(Source: [www.cisco.com](http://www.cisco.com))

---

- 1998 – 28% ([IDC](#), Oct. 1999).
- 1998 – 42% of the U.S. adult population.  
([Stratis Group](#), Apr. 1999)
- 2003 – 62% ([IDC](#), Oct. 1999).
- 2003 – 67% ([Yankee Group](#), 1999).
- 2005 – 91% ([Strategy Analytics](#), Dec. 1999).

# The Power of the Internet – 5

(Source: [www.cisco.com](http://www.cisco.com))

- NEW USERS Q1 2000: More than 5 million Americans joined the online world in the first quarter of 2000, which averages to roughly 55,000 new users each day, 2,289 new users each hour, or 38 new users each minute. ([CyberAtlas / Telecommunications Reports International](#), May 2000).
- US INTERNET USAGE: Average US Internet user went online 18 sessions, spent a total of 9 hours, 5 minutes and 24 seconds online and visited 10 unique sites per month. ([Nielsen NetRatings](#), June 2000).

# The Power of the Internet – 6

(Source: [www.cisco.com](http://www.cisco.com))

- E-MAIL 1998: The U.S. Postal Service delivered 101 billion pieces of paper mail in 1998. Estimates for e-mail messages sent in 1998 range from 618 billion to 4 trillion. ([U.S. Internet Council](#), Apr. 1999).
- E-MAIL 1999: There are 270 million e-mailboxes in the U.S. -- roughly 2.5 per user. ([eMarketer/ Messaging Online](#), Nov. 1999).
- HOURS ONLINE (Veronis, Suhler & Associates, Nov. 1999):
  - 1997 – 28 hours per capita
  - 1998 – 74 hours per capita
  - 2003 – 192 hours per capita



# ONLINE WORLDWIDE

(Source: [www.cisco.com](http://www.cisco.com))

- 1998 – 95.43 million people. ([eMarketer eStats](#) 1999).
- 1998, Dec. – 144 million (IDC, Dec. 1999).
- 1999, Dec. – 240 million (IDC, Dec. 1999).
- 2002 – over 490 million ([Computer Industry Almanac](#), Nov. 1999).
- 2005 – over 765 million ([Computer Industry Almanac](#), Nov. 1999).
- U.S. -- 136 million (36% of world's total) ([eMarketer](#), May 2000) – followed by Japan (27 M), UK (18M), and China (16 M).

# Wireless access to the Internet

(Source: [www.cisco.com](http://www.cisco.com))

- U.S. WIRELESS USERS: 61.5 million Americans will be using wireless devices to access the Internet in 2003, up from 7.4 million in the US today (728% increase). ([IDC Research](#), Feb. 2000).
- MOBILE DATA: Almost 80% of the US Internet population will access data from mobile phones in a year's time, up from the current figure of 3%. ([Corechange, Inc & Cap Gemini USA](#), Apr. 2000).

# “The network really is the computer.”

---

Tim O'Reilly, in an address at 6/2000 Java One:

“By now, it's a truism that the Internet runs on open source. Bind, the Berkeley Internet Name Daemon, is the single most mission critical program on the Internet, followed closely by Sendmail and Apache, open source servers for two of the Internet's most widely used application protocols, SMTP and HTTP.”

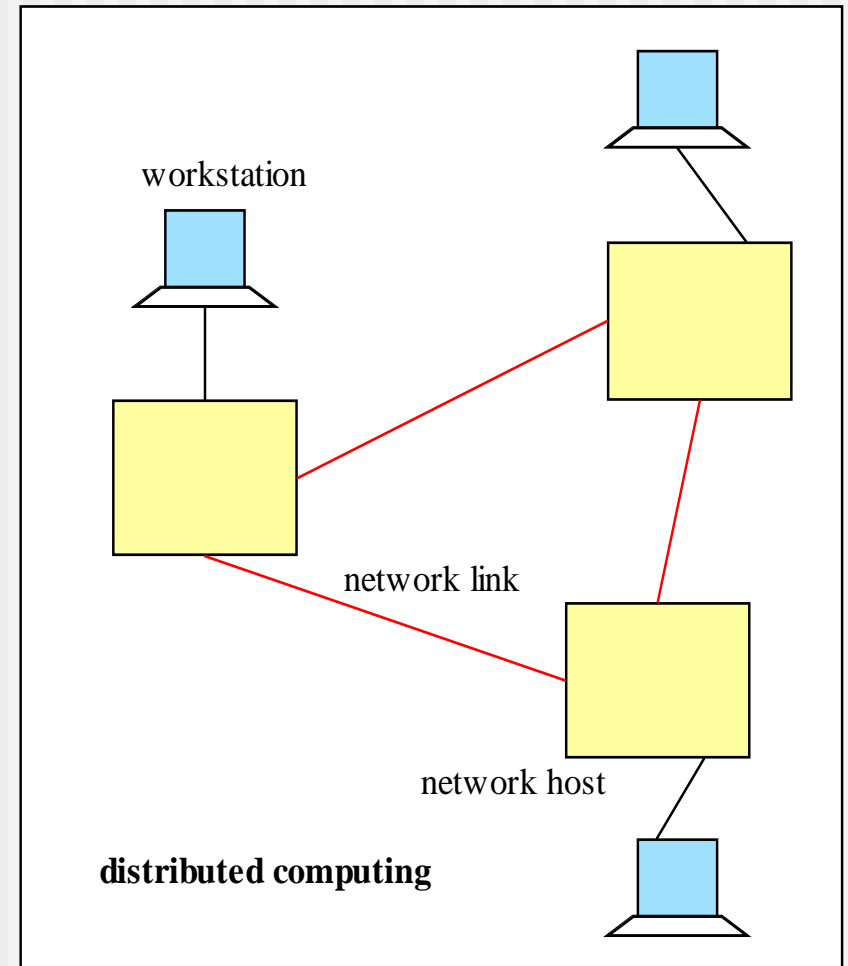
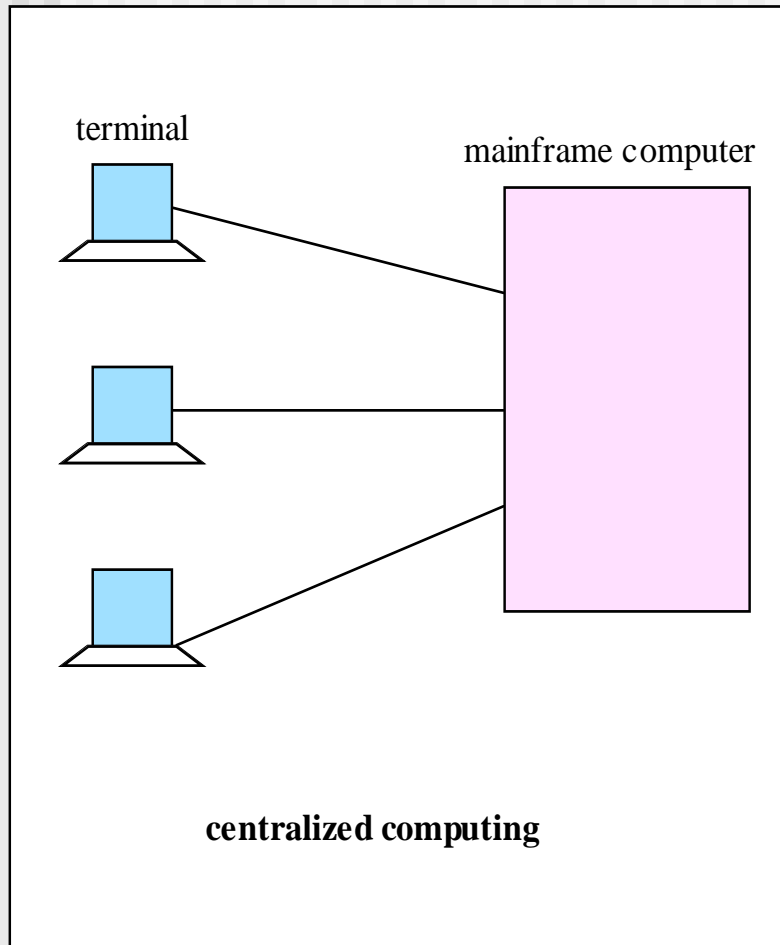
## **Early “killer apps”:**

- usenet: distributed bulletin board
- email
- talk

## **Recent “killer apps”:**

- the web
- collaborative computing

# Centralized vs. Distributed Computing



# Monolithic mainframe applications vs. distributed applications

based on <http://www.inprise.com/visibroker/papers/distributed/wp.html>

## ■ **The monolithic mainframe application architecture:**

- Separate, single-function applications, such as order-entry or billing
- Applications cannot share data or other resources
- Developers must create multiple instances of the same functionality (service).
- Proprietary (user) interfaces

## ■ **The distributed application architecture:**

- Integrated applications
- Applications can share resources
- A single instance of functionality (service) can be reused.
- Common user interfaces

# Evolution of paradigms

---

- Client-server: Socket API, remote method invocation
- Distributed objects
- Object broker: CORBA
- Network service: Jini
- Object space: JavaSpaces
- Mobile agents
- Message oriented middleware (MOM): Java Message Service
- Collaborative applications

# Cooperative distributed computing projects

---

Cooperative distributed computing projects (also called distributed computing in some literature): these are projects that parcel out large-scale computing to workstations, often making use of surplus CPU cycles. Example: [seti@home](#): project to scan data retrieved by a radio telescope to search for radio signals from another world.



# Why distributed computing?

---

- Economics: distributed systems allow the pooling of resources, including CPU cycles, data storage, input/output devices, and services.
- Reliability: a distributed system allow replication of resources and/or services, thus reducing service outage due to failures.
- The Internet has become a universal platform for distributed computing.

# The Weaknesses and Strengths of Distributed Computing

---

In any form of computing, there is always a tradeoff in advantages and disadvantages

Some of the reasons for the popularity of distributed computing :

- **The affordability of computers and availability of network access**
- **Resource sharing**
- **Scalability**
- **Fault Tolerance**

# The Weaknesses and Strengths of Distributed Computing

---

The disadvantages of distributed computing:

- **Multiple Points of Failures:** the failure of one or more participating computers, or one or more network links, can spell trouble.
- **Security Concerns:** In a distributed system, there are more opportunities for unauthorized attack.

# Introductory Basics

---

M. L. Liu

# Basics in three areas

---

Some of the notations and concepts from these areas will be employed from time to time in the presentations for this course:

- Software engineering
- Networks
- Operating systems

# Software Engineering Basics

---

# Procedural versus Object-oriented Programming

---

In building network applications, there are two main classes of programming languages: ***procedural language*** and ***object-oriented language***.

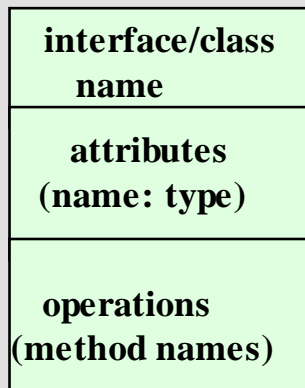
- Procedural languages, with the C language being the primary example, use procedures (functions) to break down the complexity of the tasks that an application entails.
- Object-oriented languages, exemplified by Java, use objects to encapsulate the details. Each object simulates an object in real life, carrying state data as well as behaviors. State data are represented as ***instance data***. Behaviors are represented as ***methods***.



# UML Class Diagram Notations

## Basic UML Class Diagram Notations

A class/interface is represented as follows:

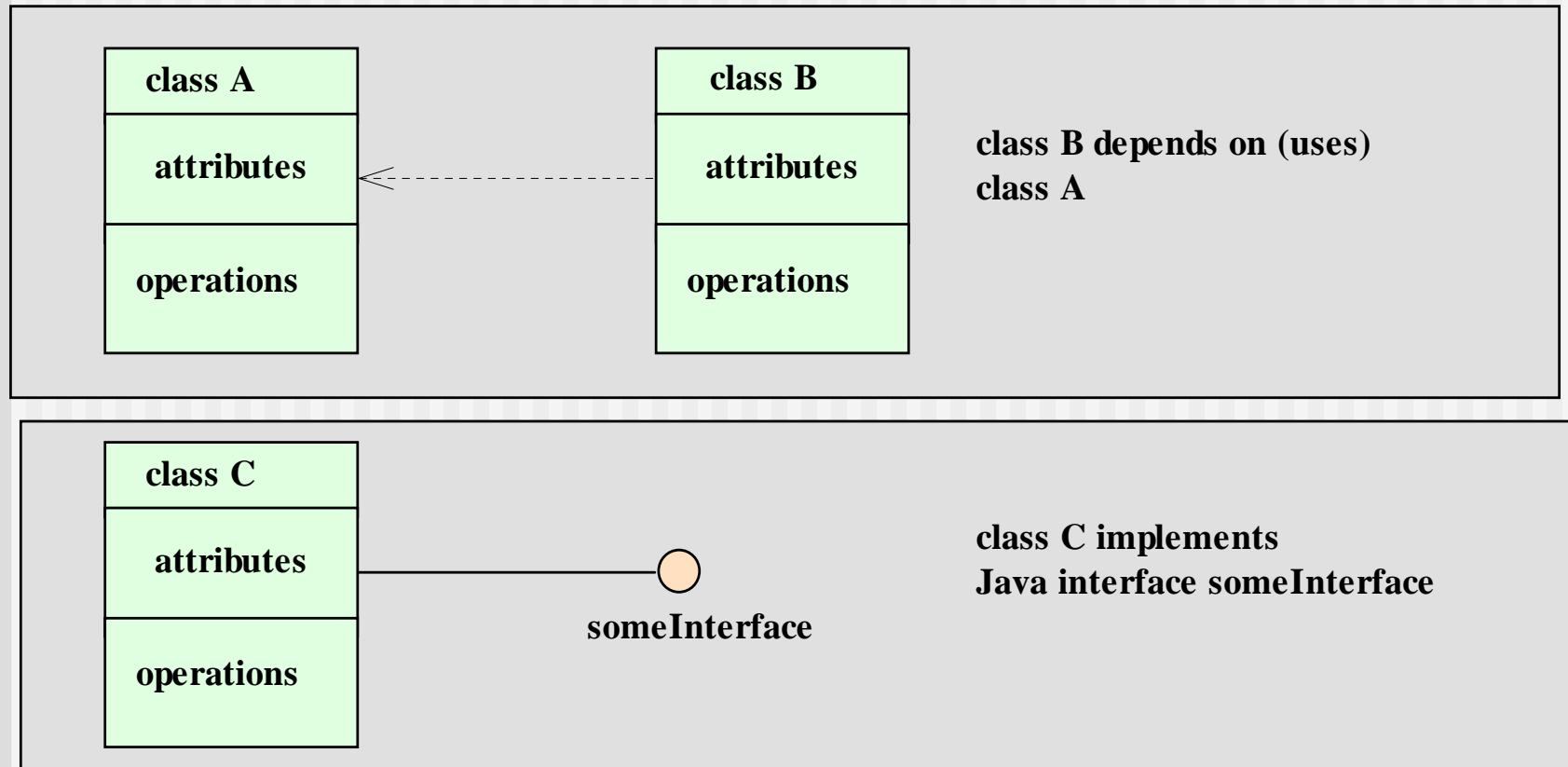


attributes are static/ instance variables/constants

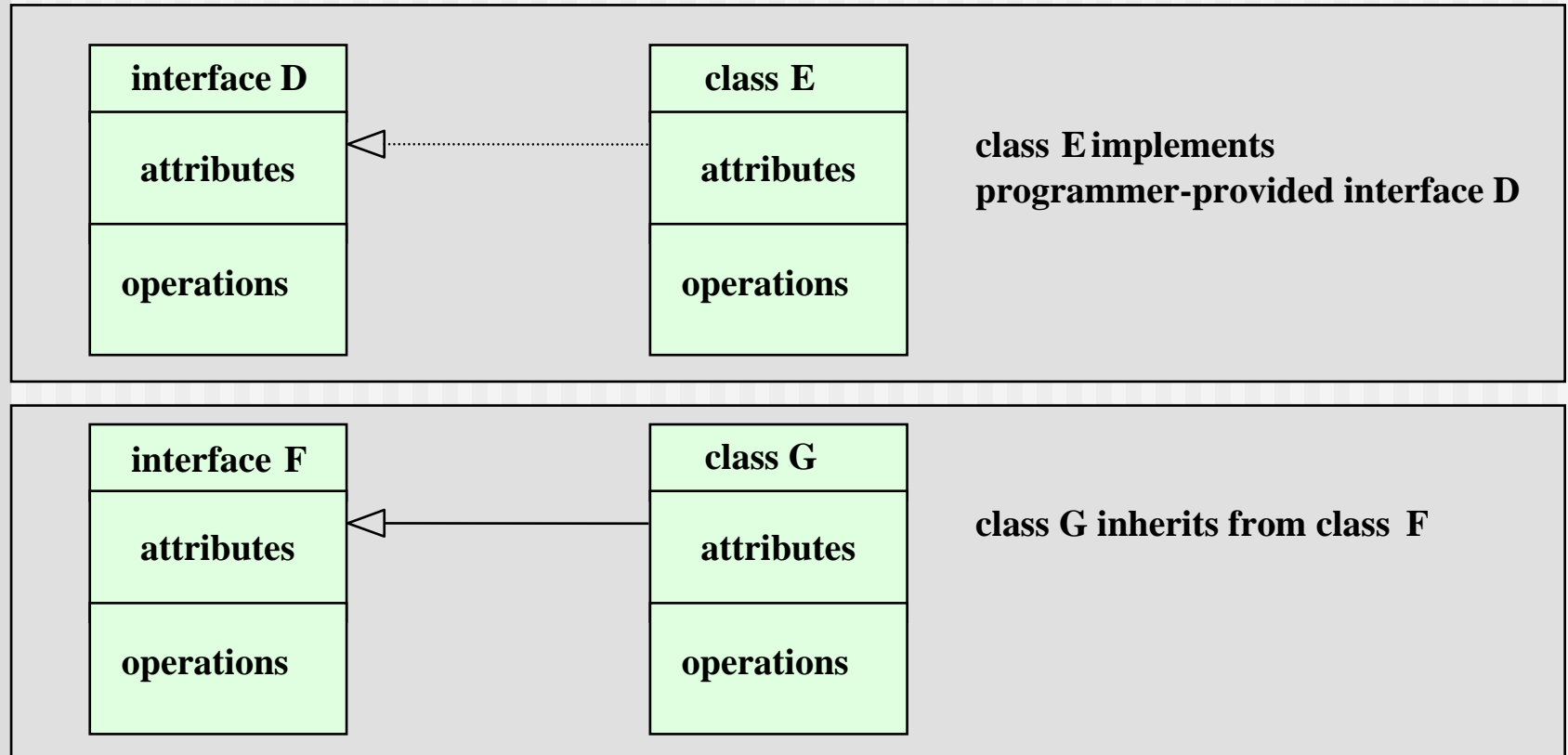
operations are static or instance methods.

**NOTE:** The shape, the style of the line (dashed or solid), the direction of the arrow, and the shape of the arrowheads (pointed, hollow, or solid) are significant.

# UML Class Diagram Notations



# UML Class Diagram Notations



# The Architecture of Distributed Applications

---

**Presentation**

**Application (Business) logic**

**Services**

---

# Network Basics

# Network standards and protocols

---

- On public networks such as the Internet, it is necessary for a common set of rules to be specified for the exchange of data.
- Such rules, called protocols, specify such matters as the formatting and semantics of data, flow control, error correction.
- Software can share data over the network using network software which supports a common set of protocols.

# Protocols

---

- **In the context of communications, a protocol is a set of rules that must be observed by the participants.**
- **In communications involving computers, protocols must be formally defined and precisely implemented. For each protocol, there must be rules that specify the followings:**
  - **How is the data exchanged encoded?**
  - **How are events (sending , receiving) synchronized so that the participants can send and receive in a coordinated order?**
- **The specification of a protocol does not dictate how the rules are to be implemented.**

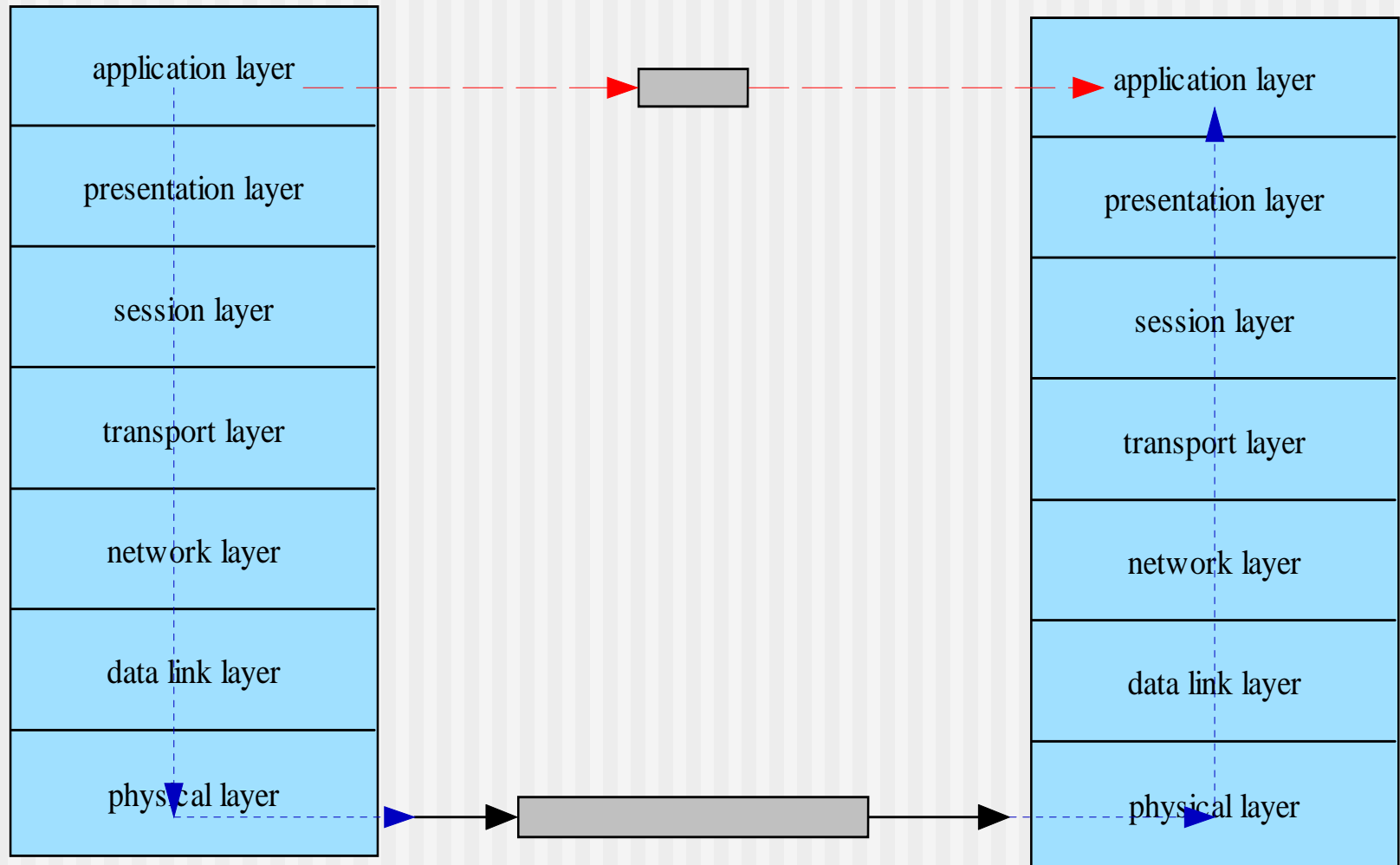


# The network architecture

---

- **Network hardware transfers electronic signals, which represent a bit stream, between two devices.**
- **Modern day network applications require an application programming interface (API) which masks the underlying complexities of data transmission.**
- **A layered network architecture allows the functionalities needed to mask the complexities to be provided incrementally, layer by layer.**
- **Actual implementation of the functionalities may not be clearly divided by layer.**

# The OSI seven-layer network architecture



# Network Architecture

---

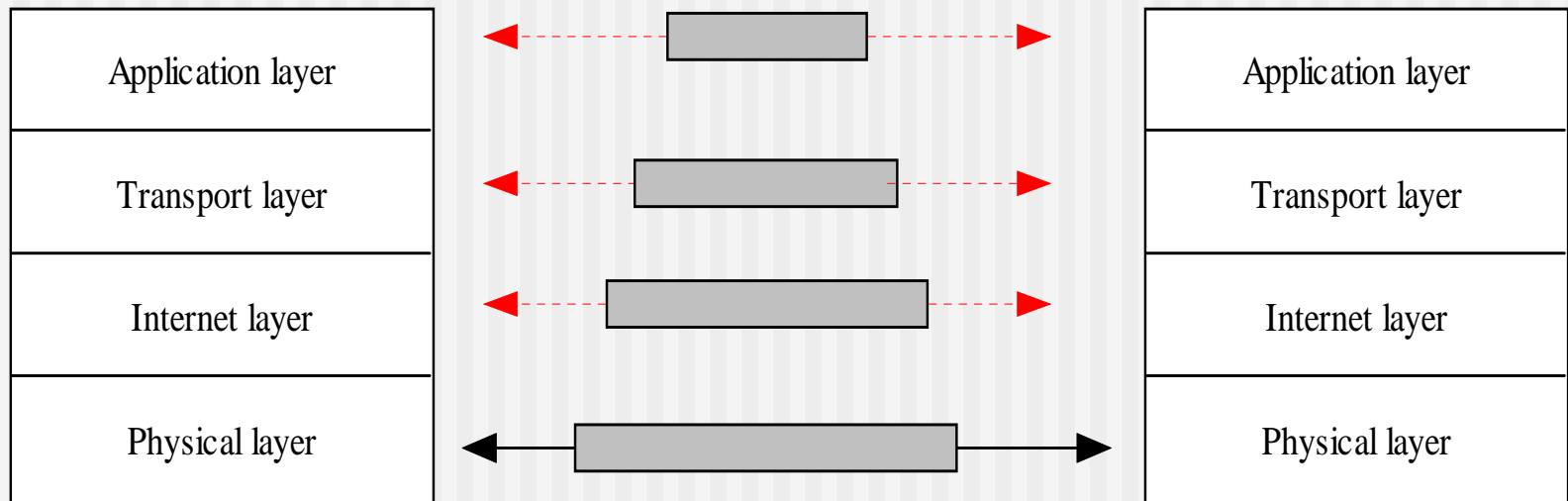
The division of the layers is **conceptual**: the implementation of the functionalities need not be clearly divided as such in the hardware and software that implements the architecture.

The conceptual division serves at least two useful purposes :

1. **Systematic specification of protocols**  
it allows protocols to be specified systematically
2. **Conceptual Data Flow**: it allows programs to be written in terms of logical data flow.

# The TCP/IP Protocol Suite

- The Transmission Control Protocol/Internet Protocol suite is a set of network protocols which supports a four-layer network architecture.
- It is currently the protocol suite employed on the Internet.



**The Internet network architecture**

# The TCP/IP Protocol Suite -2

---

- The Internet layer implements the Internet Protocol, which provides the functionalities for allowing data to be transmitted between any two hosts on the Internet.
- The Transport layer delivers the transmitted data to a specific process running on an Internet host.
- The Application layer supports the programming interface used for building a program.

## Network Resources

---

- Network resources are resources available to the participants of a distributed computing community.
- Network resources include hardware such as computers and equipment, and software such as processes, email mailboxes, files, web documents.
- An important class of network resources is **network services** such as the World Wide Web and file transfer (FTP), which are provided by specific processes running on computers.

# Identification of Network Resources

---

One of the key challenges in distributed computing is the unique identification of resources available on the network, such as e-mail mailboxes, and web documents.

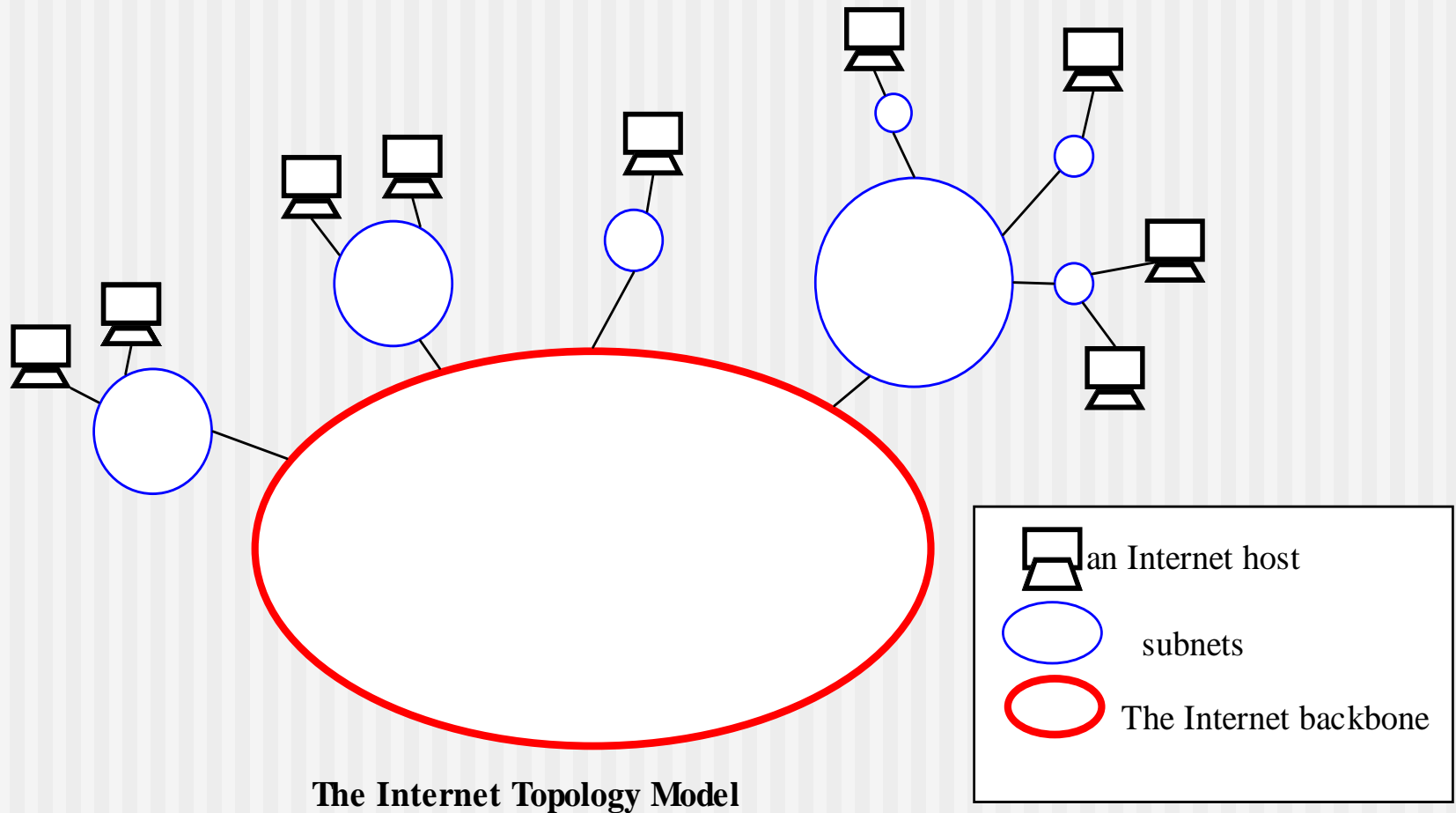
- Addressing an Internet Host
- Addressing a process running on a host
- Email Addresses
- Addressing web contents: URL

---

# Addressing an Internet Host



# The Internet Topology



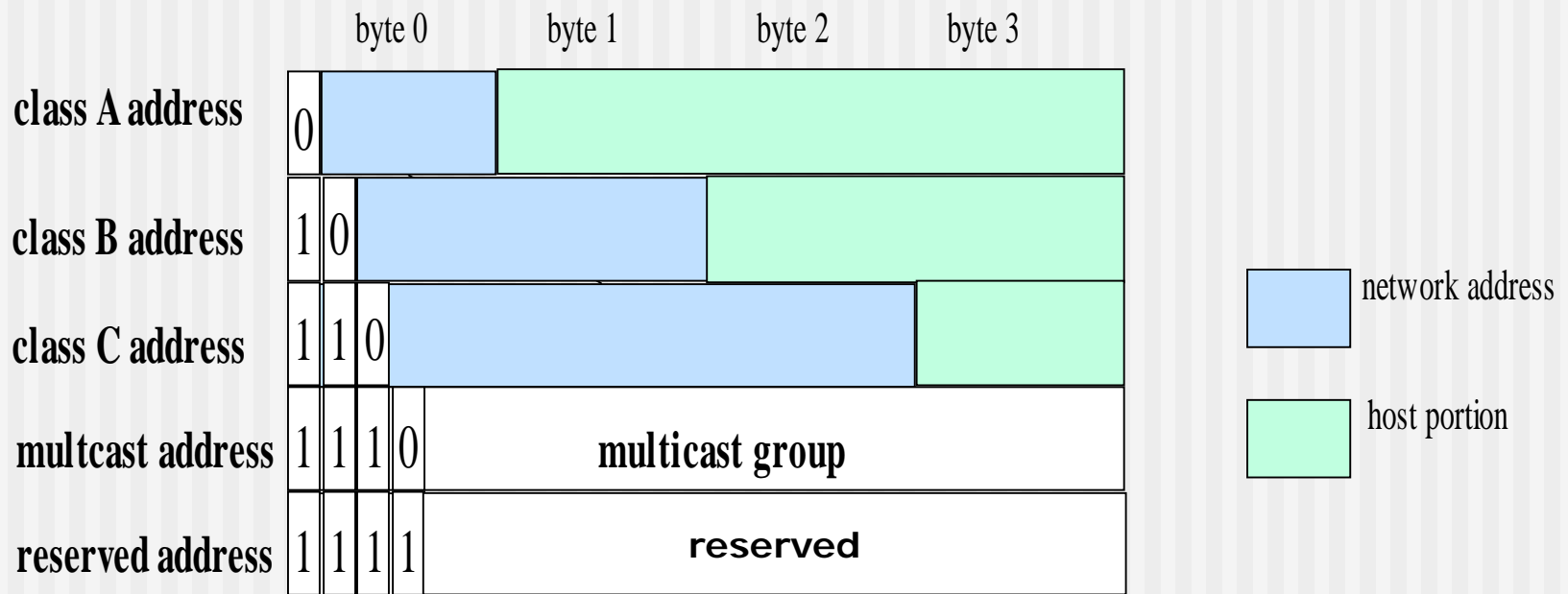
# The Internet Topology

---

- The internet consists of an hierarchy of networks, interconnected via a network backbone.
- Each network has a unique network address.
- Computers, or hosts, are connected to a network. Each host has a unique ID within its network.
- Each process running on a host is associated with zero or more **ports**. A port is a logical entity for data transmission.

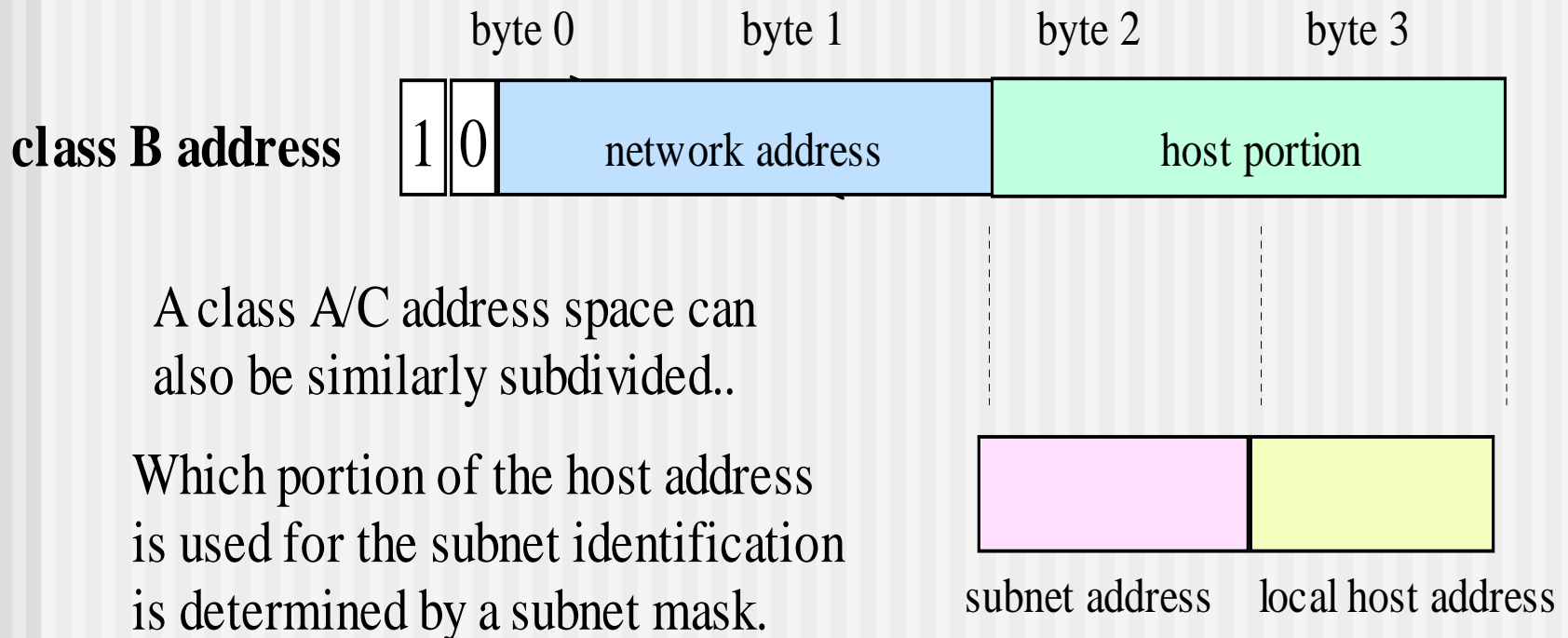
# The Internet addressing scheme

- In IP version 4, each address is 32 bit long.
- The address space accommodates  $2^{32}$  (4.3 billion) addresses in total.
- Addresses are divided into 5 classes (A through E)



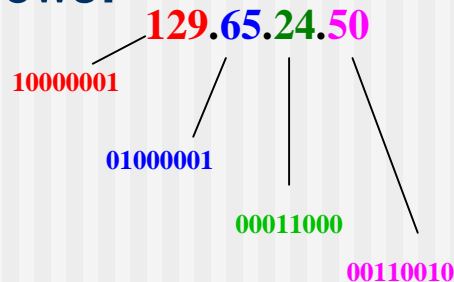
# The Internet addressing scheme - 2

## Subdividing the host portion of an Internet address:



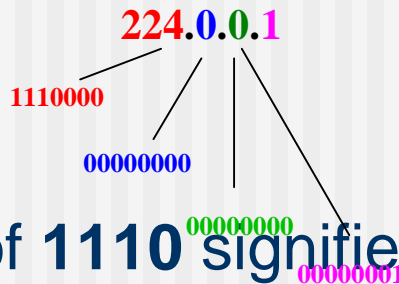
## Example:

Suppose the dotted-decimal notation for a particular Internet address is 129.65.24.50. The 32-bit binary expansion of the notation is as follows:



Since the leading bit sequence is 10, the address is a **Class B** address. Within the class, the network portion is identified by the remaining bits in the first two bytes, that is, **0000101000001**, and the host portion is the values in the last two bytes, or **0001100000110010**. For convenience, the binary prefix for class identification is often included as part of the network portion of the address, so that we would say that this particular address is at network **129.65** and then at host address **24.50** on that network.

Given the **address 224.0.0.1**, one can expand it as follows:



The binary prefix of **1110** signifies that this is class D, or multicast, address. Data packets sent to this address should therefore be delivered to the multicast group **00000000000000000000000000000001**.

# The Internet Address Scheme - 3

- For human readability, Internet addresses are written in a dotted decimal notation:

nnn.nnn.nnn.nnn, where each nnn group is a decimal value in the range of 0 through 255

# Internet host table (found in /etc/hosts file)

127.0.0.1	localhost
129.65.242.5	falcon.csc.calpoly.edu falcon loghost
129.65.241.9	falcon-srv.csc.calpoly.edu falcon-srv
129.65.242.4	hornet.csc.calpoly.edu hornet
129.65.241.8	hornet-srv.csc.calpoly.edu hornet-srv
129.65.54.9	onion.csc.calpoly.edu onion
129.65.241.3	hercules.csc.calpoly.edu hercules

## IP version 6 Addressing Scheme

---

- Each address is 128-bit long.
- There are three types of addresses:
  - Unicast: An identifier for a single interface.
  - Anycast: An identifier for a set of interfaces (typically belonging to different nodes).
  - Multicast: An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to a multicast address is delivered to all interfaces identified by that address.
- See Request for Comments: 2373  
<http://www.faqs.org/rfcs/> (link is in book's reference)

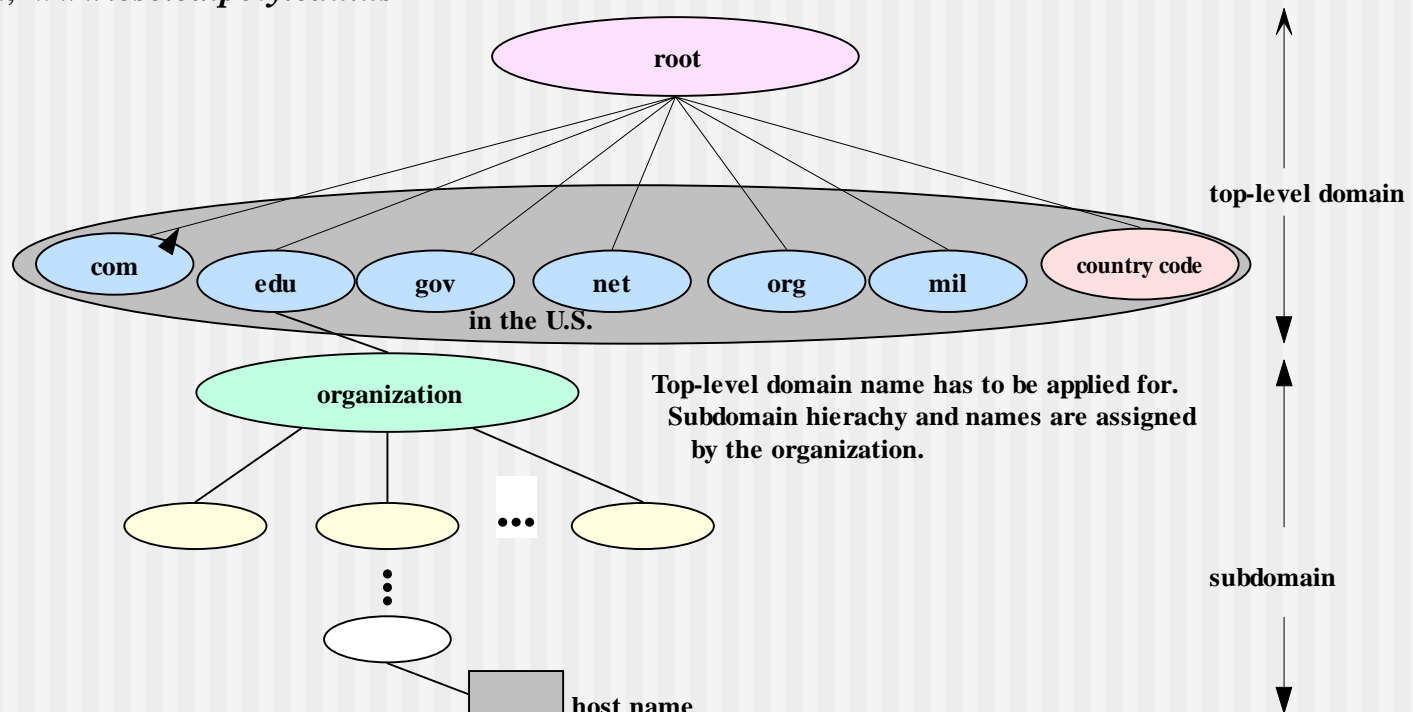


# The Domain Name System (DNS)

**For user friendliness, each Internet address is mapped to a symbolic name, using the DNS, in the format of:**

`<computer-name>.<subdomain hierarchy>.<organization>.<sector name>{.<country code>}`

e.g., *www.csc.calpoly.edu.us*



# The Domain Name System

---

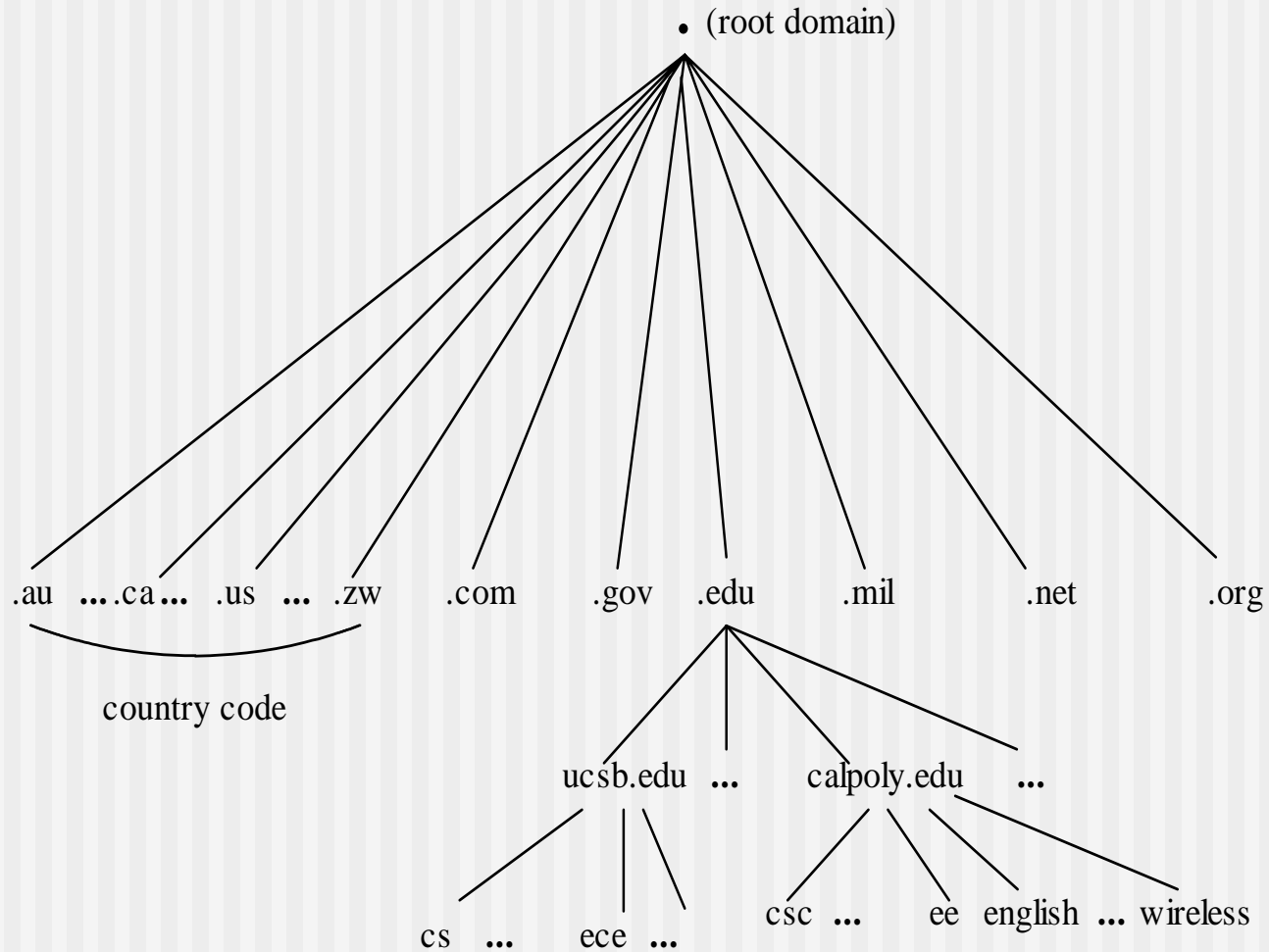
- For network applications, a domain name must be mapped to its corresponding Internet address.
- Processes known as domain name system servers provide the mapping service, based on a distributed database of the mapping scheme.
- The mapping service is offered by thousands of DNS servers on the Internet, each responsible for a portion of the name space, called a *zone*. The servers that have access to the DNS information (zone file) for a zone is said to have authority for that zone.

# Top-level Domain Names

---

- **.com:** For commercial entities, which anyone, anywhere in the world, can register.
- **.net :** Originally designated for organizations directly involved in Internet operations. It is increasingly being used by businesses when the desired name under "com" is already registered by another organization. Today anyone can register a name in the Net domain.
- **.org:** For miscellaneous organizations, including non-profits.
- **.edu:** For four-year accredited institutions of higher learning.
- **.gov:** For US Federal Government entities
- **.mil:** For US military
- **Country Codes :**For individual countries based on the International Standards Organization. For example, ca for Canada, and jp for Japan.

# Domain Name Hierarchy



# Name lookup and resolution

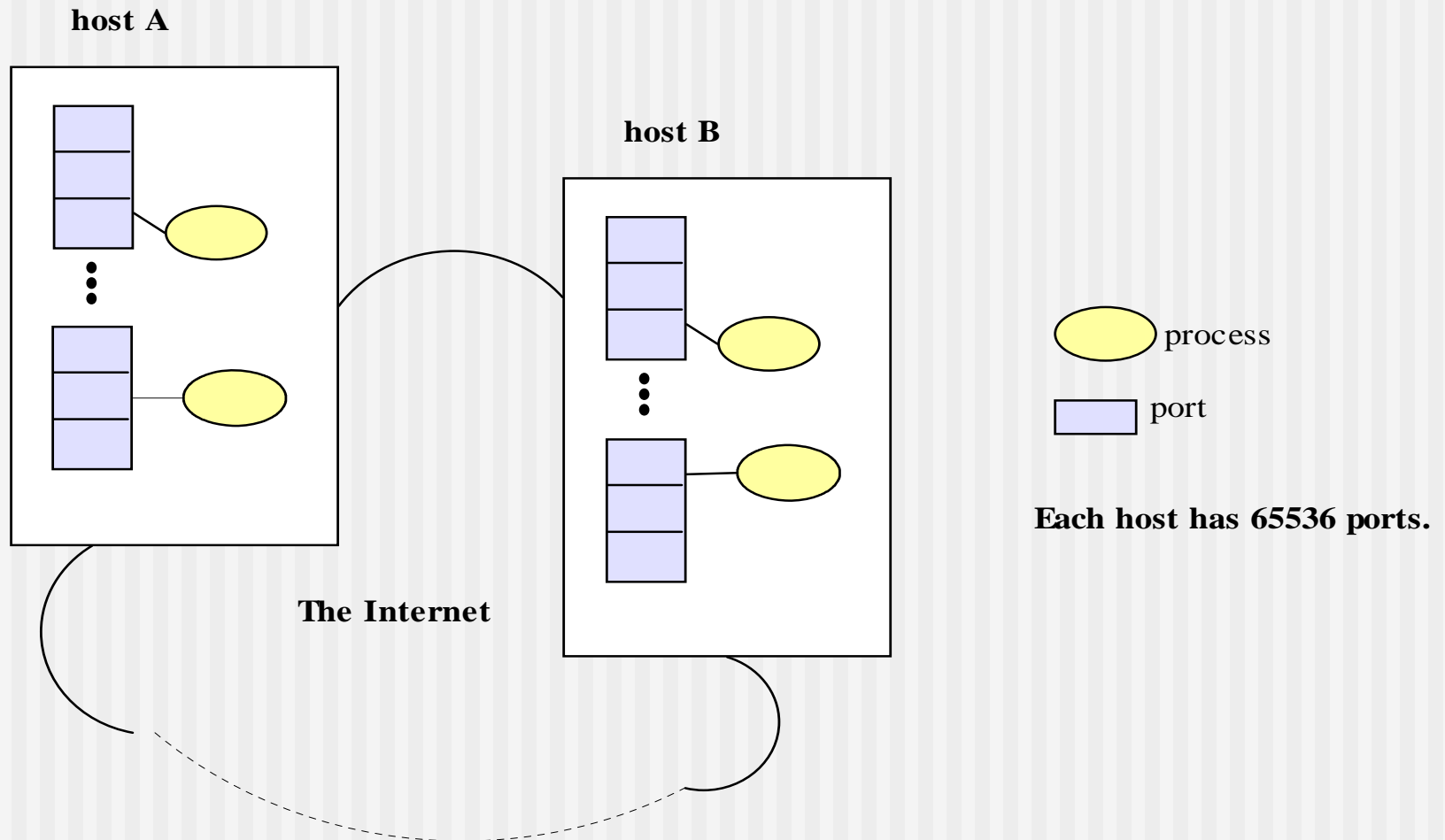
---

- If a domain name is used to address a host, its corresponding IP address must be obtained for the lower-layer network software.
- The mapping, or name resolution, must be maintained in some registry.
- For runtime name resolution, a network service is needed; a protocol must be defined for the naming scheme and for the service. Example: The DNS service supports the DNS; the Java RMI registry supports RMI object lookup; JNDI is a network service lookup protocol.

# Addressing a process running on a host

---

# Logical Ports



# Well Known Ports

---

- Each Internet host has  $2^{16}$  (65,535) logical ports. Each port is identified by a number between 1 and 65535, and can be allocated to a particular process.
- Port numbers between 1 and 1023 are reserved for processes which provide well-known services such as *finger*, *FTP*, *HTTP*, and *email*.



# Well-known ports

## Assignment of some well-known ports

Protocol	Port	Service
echo	7	IPC testing
daytime	13	provides the current date and time
ftp	21	file transfer protocol
telnet	23	remote, command-line terminal session
smtp	25	simple mail transfer protocol
time	37	provides a standard time
finger	79	provides information about a user
http	80	web server
RMI Registry	1099	registry for Remote Method Invocation
special web server	8080	web server which supports servlets, JSP, or ASP

## Choosing a port to run your program

---

- For our programming exercises: when a port is needed, choose a random number above the well known ports: 1,024- 65,535.
- If you are providing a network service for the community, then arrange to have a port assigned to and reserved for your service.

---

# Addressing a Web Document

# The Uniform Resource Identifier (URI)

---

- Resources to be shared on a network need to be uniquely identifiable.
- On the Internet, a URI is a character string which allows a resource to be located.
- There are two types of URIs:
  - URL (Uniform Resource Locator) points to a specific resource at a specific location
  - URN (Uniform Resource Name) points to a specific resource at a nonspecific location.

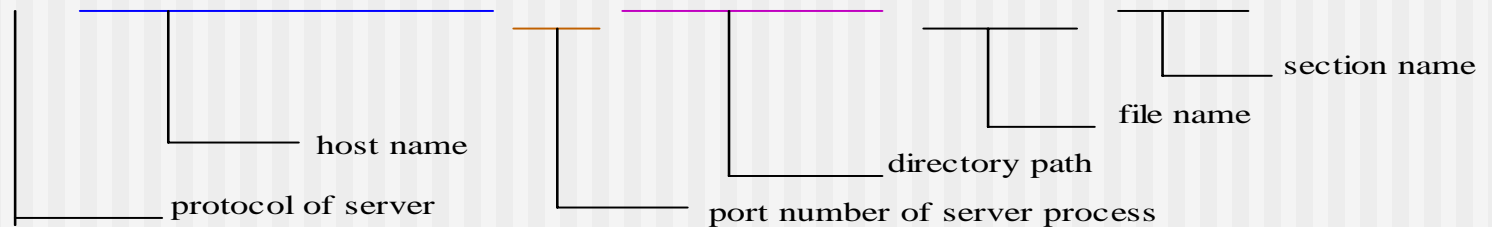
# URL

A URL has the format of:

protocol://host address[:port]/directory path/file name#section

**A sample URL:**

http://www.csc.calpoly.edu:8080/~mliu/CSC369/hw.html # hw1



*Other protocols that can appear in a URL are:*

*file*  
*ftp*  
*gopher*  
*news*  
*telnet*  
*WAIS*

# More on URL

---

- The path in a URL is relative to the document root of the server. On the CSL systems, a user's document root is `~/www`.
- A URL may appear in a document in a relative form:

`< a href="another.html">`

and the actual URL referred to will be *another.html* preceded by the protocol, hostname, directory path of the document .

# Operating Systems Basics

---

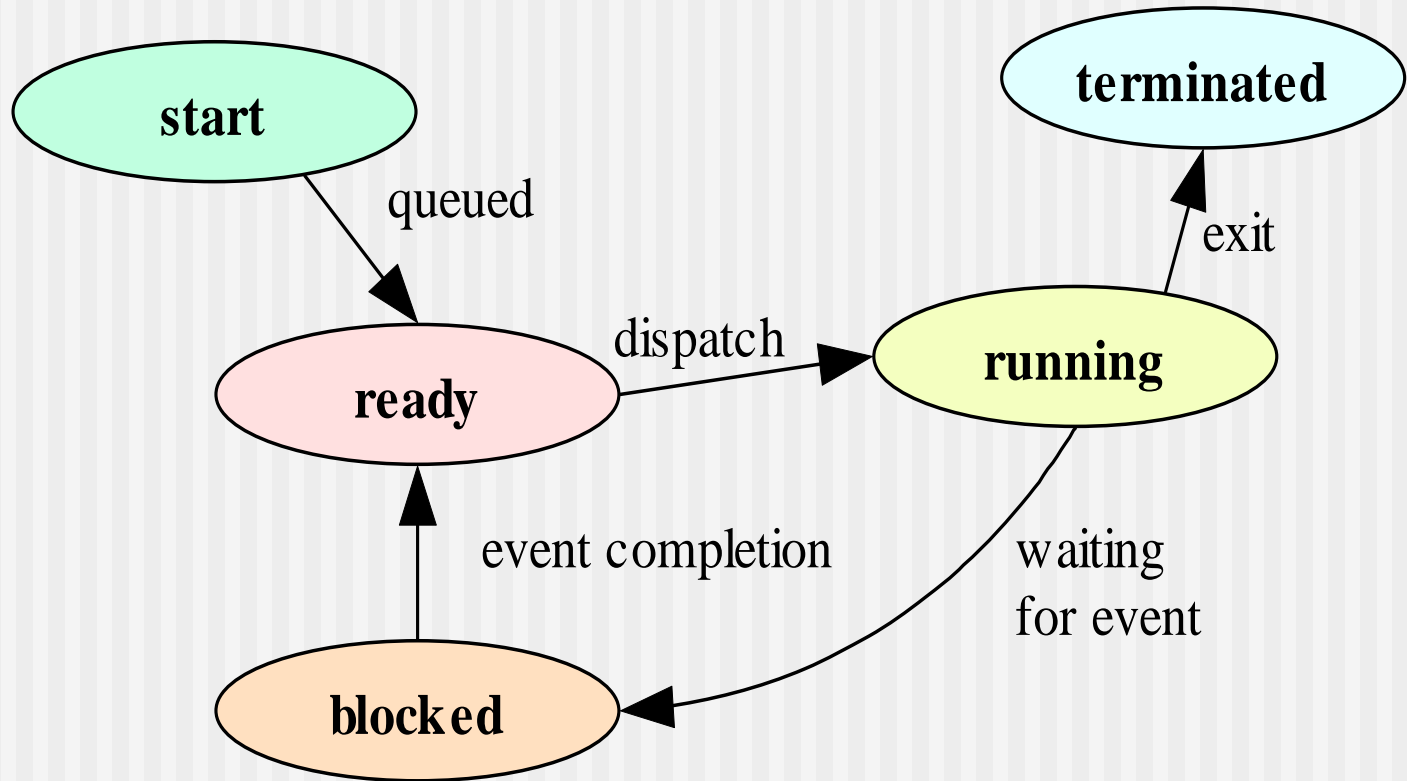
# Operating systems basics

---

- A process consists of an executing program, its current values, state information, and the resources used by the operating system to manage its execution.
- A program is an artifact constructed by a software developer; a process is a dynamic entity which exists only when a program is run.



# Process State Transition Diagram



**Simplified finite state diagram for a process's lifetime**

# Java processes

---

- There are three types of Java program: applications, applets, and servlets, all are written as a class.
  - A Java application program has a main method, and is run as an independent(standalone) process.
  - An applet does not have a main method, and is run using a browser or the appletviewer.
  - A servlet does not have a main method, and is run in the context of a web server.
- A Java program is compiled into bytecode, a universal object code. When run, the bytecode is interpreted by the Java Virtual Machine (JVM).

# Three Types of Java programs

---

## ■ **Applications**

a program whose byte code can be run on any system which has a Java Virtual Machine. An application may be standalone (monolithic) or distributed (if it interacts with another process).

## ■ **Applets**

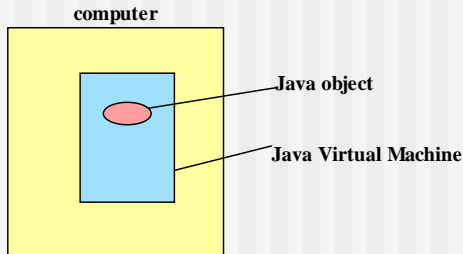
A program whose byte code is downloaded from a remote machine and is run in the browser's Java Virtual Machine.

## ■ **Servlets**

A program whose byte code resides on a remote machine and is run at the request of an HTTP client (a browser).

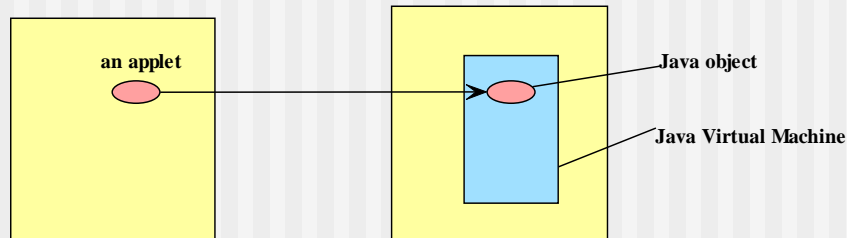
# Three Types of Java programs

A standalone Java application is run on a local machine



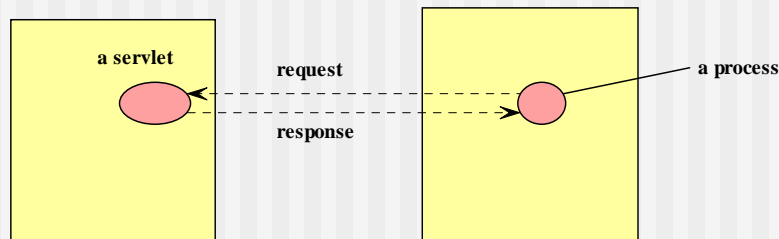
---

An applet is an object downloaded (transferred) from a remote machine, then run on a local machine.



---

A servlet is an object that runs on a remote machine and interacts with a local program using a request-response protocol



# A sample Java application

```

/*****
* A sample of a simple Java application.
* M. Liu      1/8/02
*****/

import java.io.*;

class MyProgram{

    public static void main(String[ ] args)
        throws IOException{

        BufferedReader keyboard = new
            BufferedReader(new InputStreamReader(System.in));
        String theName;
        System.out.println("What is your name?");
        theName = keyboard.readLine( );
        System.out.print("Hello " + theName );
        System.out.println(" - welcome to CSC369.\n");

    } // end main

} //end class

```

# A Sample Java Applet

```
/*  
 * A sample of a simple applet.  
 * M. Liu 1/8/02  
 */  
  
import java.applet.Applet;  
import java.awt.*;  
  
public class MyApplet extends Applet{  
  
    public void paint(Graphics g){  
        setBackground(Color.blue);  
  
        Font Claude = new Font("Arial", Font.BOLD, 40);  
        g.setFont(Claude);  
        g.setColor(Color.yellow);  
        g.drawString("Hello World!", 100, 100);  
    } // end paint  
  
} //end class
```

```
<!-- A web page which, when browsed, will run >  
<!-- the MyApplet applet>  
<!-- M. Liu 1/8/02>  
  
<title>SampleApplet</title>  
<hr>  
  
<applet code="MyApplet.class" width=500 height=500>  
</applet>  
  
<hr>  
<a href="Hello.java">The source.</a>
```

# A Sample Java Servlet

```

/*****
 * A sample of a simple Java servlet.
 * M. Liu      1/8/02
 *****/

import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class MyServlet extends HttpServlet {

    public void doGet (HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException {

        PrintWriter out;
        String title = "MyServlet Output";
        // set content type and other response header
        // fields first
        response.setContentType("text/html");
        // then write the data of the response
        out = response.getWriter();
        out.println("<HTML><HEAD><TITLE>");
        out.println(title);
        out.println("</TITLE></HEAD><BODY>");
        out.println("<H1>" + title + " </H1>");
        out.println("<P>Hello World!");
        out.println("</BODY></HTML>");
        out.close();
    } //end doGet

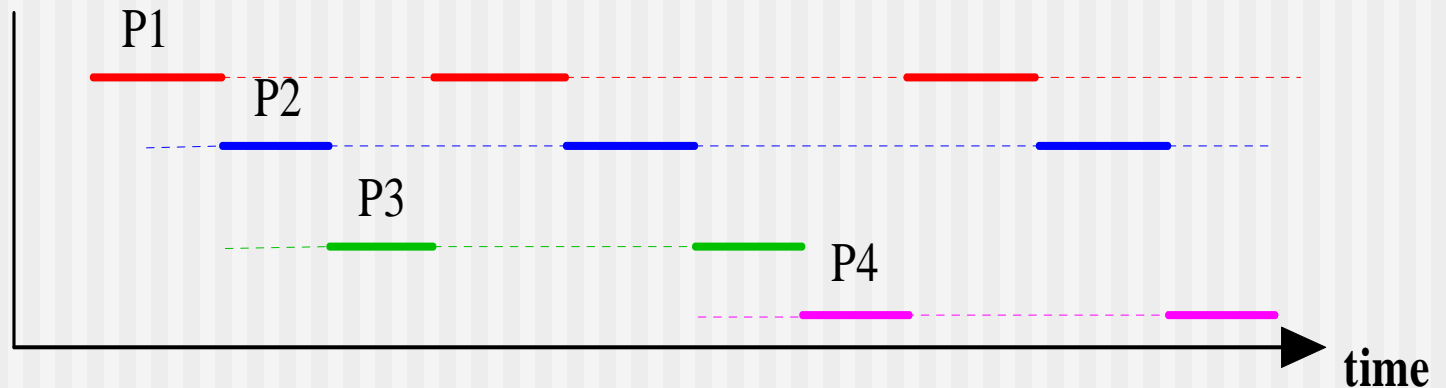
} //end class

```

# Concurrent Processing

On modern day operating systems, multiple processes *appear* to be executing concurrently on a machine by timesharing resources.

Processes

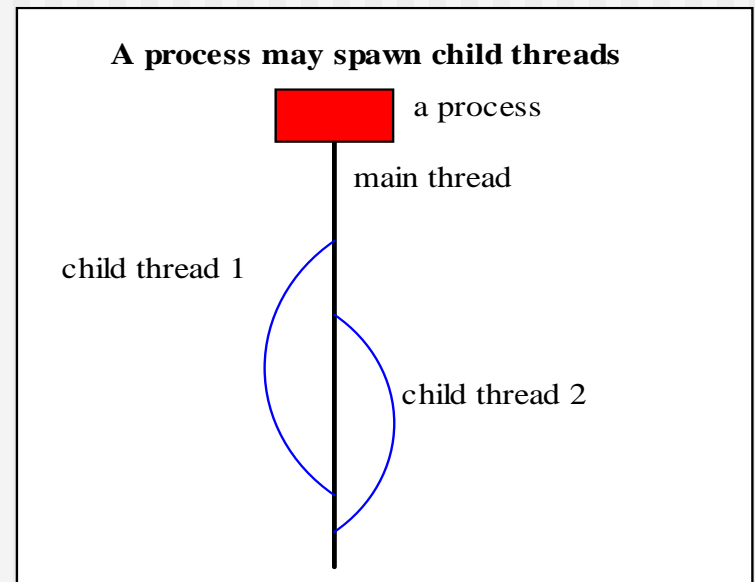
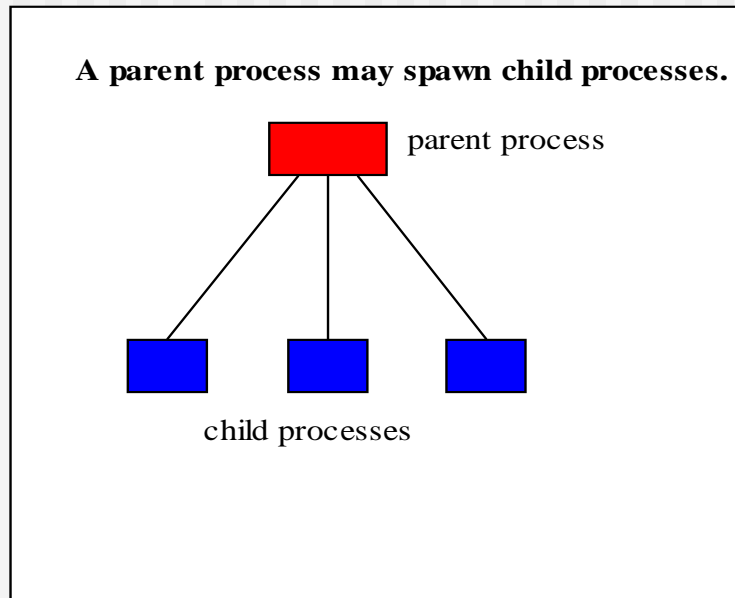


Timesharing of a resource



# Concurrent processing within a process

It is often useful for a process to have parallel threads of execution, each of which timeshare the system resources in much the same way as concurrent processes.



# Java threads

---

- The Java Virtual Machine allows an application to have multiple threads of execution running concurrently.
- Java provides a Thread class:  
    public class **Thread**  
    extends [Object](#)  
    implements [Runnable](#)
- When a Java Virtual Machine starts up, there is usually a single thread (which typically calls the method named **main** of some designated class). The Java Virtual Machine continues to execute threads until either of the following occurs:
  - The **exit** method of class **Runtime** has been called and the security manager has permitted the exit operation to take place.
  - All threads have terminated, either by returning from the call to the **run** method or by throwing an exception that propagates beyond the **run** method.

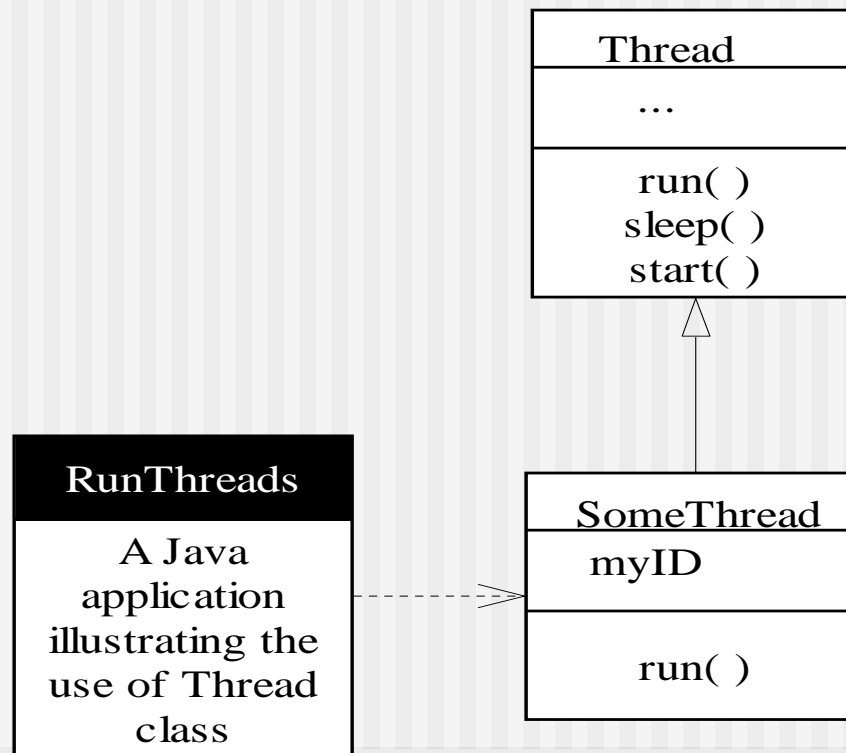
## Two ways to create a new thread of execution

---

- Using a subclass of the **Thread** class
- Using a class that implements the **Runnable** interface

## Create a class that is a subclass of the Thread class

Declare a class to be a subclass of **Thread**. This subclass should override the `run` method of class `Thread`. An instance of the subclass can then be allocated and started:



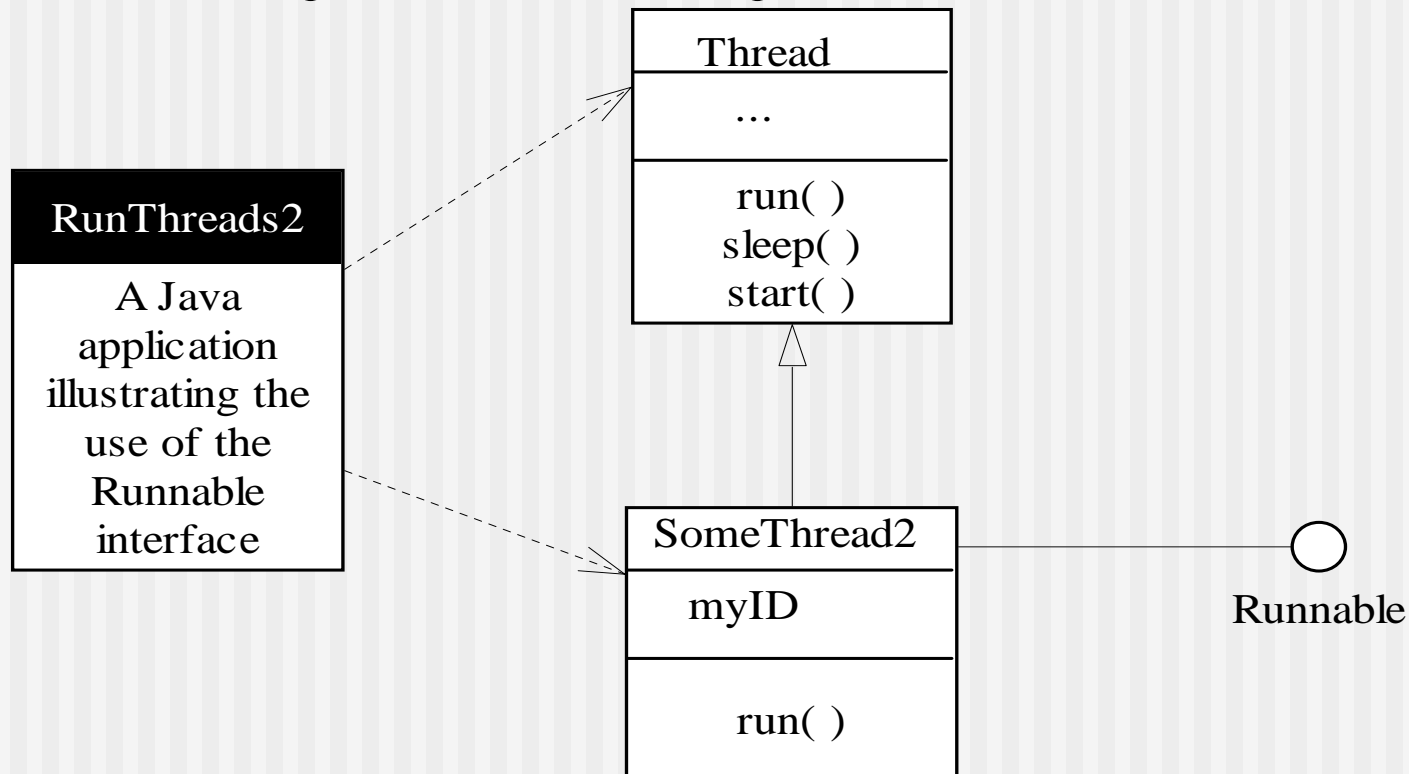
## Create a class that is a subclass of the Thread class

```
import SomeThread;  
public class RunThreads  
{  
    public static void main (String[] args)  
    {  
        SomeThread p1 = new SomeThread(1);  
        p1.start();  
  
        SomeThread p2 = new SomeThread(2);  
        p2.start();  
  
        SomeThread p3 = new SomeThread(3);  
        p3.start();  
    }  
} // end class RunThreads
```

```
public class SomeThread extends Thread {  
    int myID;  
  
    SomeThread(int id) {  
        this.myID = id;  
    }  
  
    public void run( ) {  
        int i;  
        for (i = 1; i < 11; i++)  
            System.out.println ("Thread"+myID + ": " + i);  
    }  
} //end class SomeThread
```

# Java Threads-2

The other way to create a thread is to declare a class that implements the **Runnable** interface. That class then implements the **run** method. An instance of the class can then be allocated, passed as an argument when creating **Thread**, and started.



## Create a class that implements the Runnable interface

```
public class RunThreads2
{
    public static void main (String[] args)
    {
        Thread p1 = new Thread(new SomeThread2(1));
        p1.start();

        Thread p2 = new Thread(new SomeThread2(2));
        p2.start();

        Thread p3 = new Thread(new SomeThread2(3));
        p3.start();
    }
}
```

```
class SomeThread2 implements Runnable {
    int myID;

    SomeThread2(int id) {
        this.myID = id;
    }

    public void run() {
        int i;
        for (i = 1; i < 11; i++)
            System.out.println ("Thread"+myID + ": " + i);
    }
} //end class SomeThread
```

# Program samples

---

- RunThreads.java
- SomeThread.java
- RunThreads2.java
- SomeThread2.java

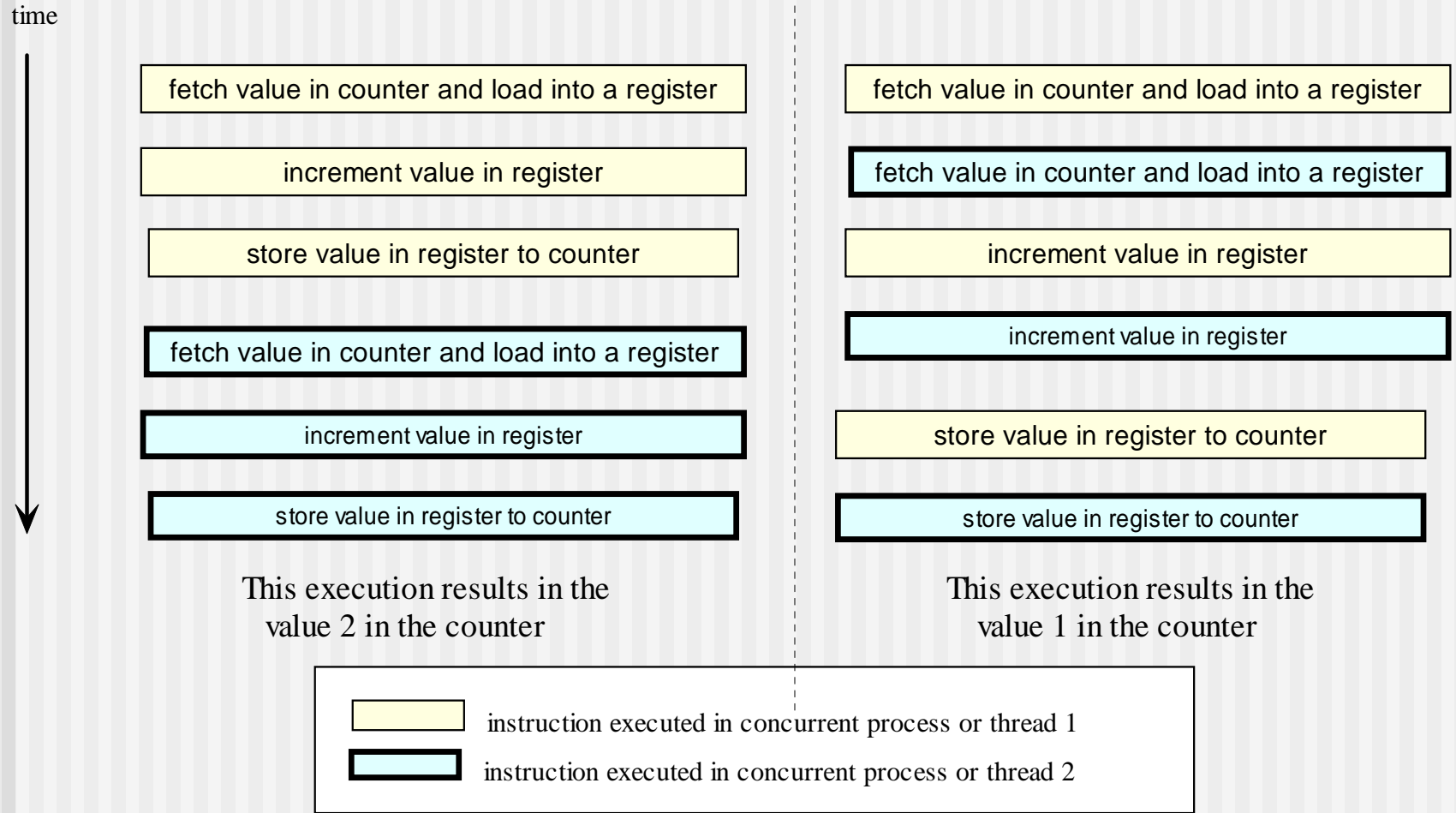


# Thread-safe Programming

---

- When two threads independently access and update the same data object, such as a counter, as part of their code, the updating needs to be synchronized. (See next slide.)
- Because the threads are executed concurrently, it is possible for one of the updates to be overwritten by the other due to the sequencing of the two sets of machine instructions executed in behalf of the two threads.
- To protect against the possibility, a **synchronized** method can be used to provide mutual exclusion.

# Race Condition



# Synchronized method in a thread

```
class SomeThread3 implements Runnable {  
    static int count=0;  
  
    SomeThread3() {  
        super();  
    }  
  
    public void run() {  
        update( );  
    }  
  
    static public synchronized void update( ){  
        int myCount = count;  
        myCount++;  
        count = myCount;  
        System.out.println("count="+count+  
            "; thread count=" + Thread.activeCount( ));  
    }  
}
```

# Summary - 1

---

We discussed the following topics:

- What is meant by distributed computing
- Distributed system
- Distributed computing vs. parallel computing
- Basic concepts in operating system: processes and threads

# Summary - 2

---

- Basic concepts in data communication:
  - Network architectures: the OSI model and the Internet model
  - Connection-oriented communication vs. connectionless communication
  - Naming schemes for network resources
    - The Domain Name System (DNS)
    - Protocol port numbers
    - Uniform Resource Identifier (URI)
    - Email addresses

# Summary-3

---

- Basic concepts in software engineering:
  - Procedural programming vs. object-oriented programming
  - UML Class diagrams
  - The three-layered architecture of distributed applications: presentation layer, application or business logic, the service layer
  - The terms toolkit, framework, and component