Operating Systems and Networks

Tomas Olovsson

Computer Science & Engineering

Networking slides from Kurose & Ross, ”Computer Networking”

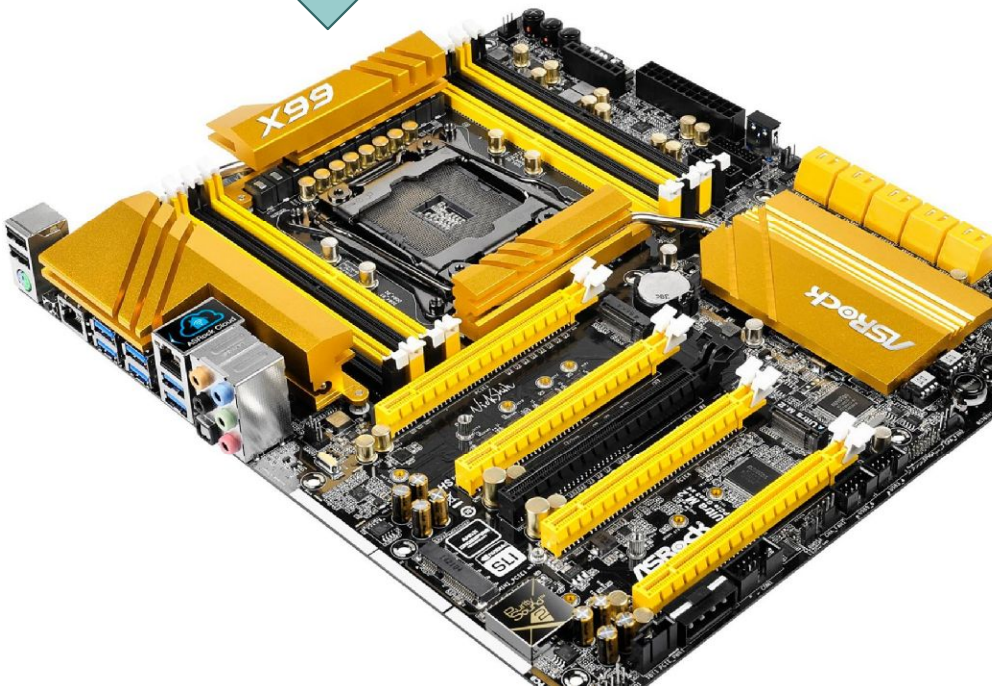
Roadmap

- Operating Systems
  - What is an Operating System
  - OS evolution
  - OS details
- Networking
  - The Internet
  - Network protocols
  - Security

How do I run a program?

```
{
    var mode = String (orient);
    var p = subMenu.offsetParent;
    var subMenuWidth = cmGetWidth (subMenu);
    var horiz = cmGetHorizontalAlign (obj, mode, p, subMenuWidth);
    if (mode.charAt (0) == 'h')
    {
        if (mode.charAt (1) == 'b')
            subMenu.style.top = (cmGetYAt (obj, p) + cmGetHeight (
        else
            subMenu.style.top = (cmGetYAt (obj, p) - cmGetHeight (
        if (horiz == 'r')
            subMenu.style.left = (cmGetXAt (obj, p)) + 'px';
        else
            subMenu.style.left = (cmGetXAt (obj, p) + cmGetWidth (
    }
    else
    {
        if (horiz == 'r')
            subMenu.style.left = (cmGetXAt (obj, p) + cmGetWidth (
        else
            subMenu.style.left = (cmGetXAt (obj, p) - subMenuWidth
        if (mode.charAt (1) == 'b')
            subMenu.style.top = (cmGetYAt (obj, p)) + 'px';
        else
            subMenu.style.top = (cmGetYAt (obj, p) + cmGetHeight (
    }
```

Some help is needed!



How do I run a program?

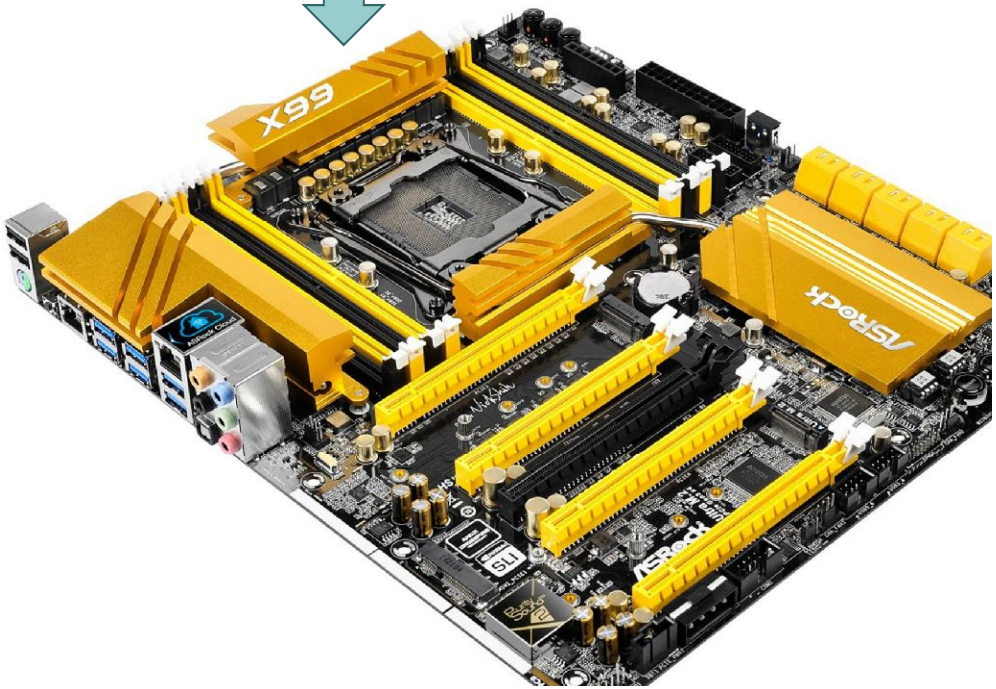
```
00000000
00000001
00000003
00000007
00000008
0000000C
0000000F
00000011
00000014
00000016
00000019
0000001B
0000001D
0000001F
00000022
00000025
```

```
push    ebp
mov     ebp, esp
movzx   ecx, [ebp+arg_0]
pop     ebp
movzx   dx, cl
lea     eax, [edx+edx]
add     eax, edx
shl     eax, 2
add     eax, edx
shr     eax, 8
sub     cl, al
shr     cl, 1
add     al, cl
shr     al, 5
movzx   eax, al
retn
```

```
{
    var mode = String (orient);
    var p = subMenu.offsetParent;
    var subMenuWidth = cmGetWidth (subMenu);
    var horiz = cmGetHorizontalAlign (obj, mode, p, subMenuWidth);
    if (mode.charAt (0) == 'h')
    {
        if (mode.charAt (1) == 'b')
            subMenu.style.top = (cmGetYAt (obj, p) + cmGetHeight (
        else
            subMenu.style.top = (cmGetYAt (obj, p) - cmGetHeight (
        if (horiz == 'r')
            subMenu.style.left = (cmGetXAt (obj, p)) + 'px';
        else
            subMenu.style.left = (cmGetXAt (obj, p) + cmGetWidth (
    }
    else
    {
        if (horiz == 'r')
            subMenu.style.left = (cmGetXAt (obj, p) + cmGetWidth (
        else
            subMenu.style.left = (cmGetXAt (obj, p) - subMenuWidth
        if (mode.charAt (1) == 'b')
            subMenu.style.top = (cmGetYAt (obj, p)) + 'px';
        else
            subMenu.style.top = (cmGetYAt (obj, p) + cmGetHeight (
    }
```

Operating system

iOS



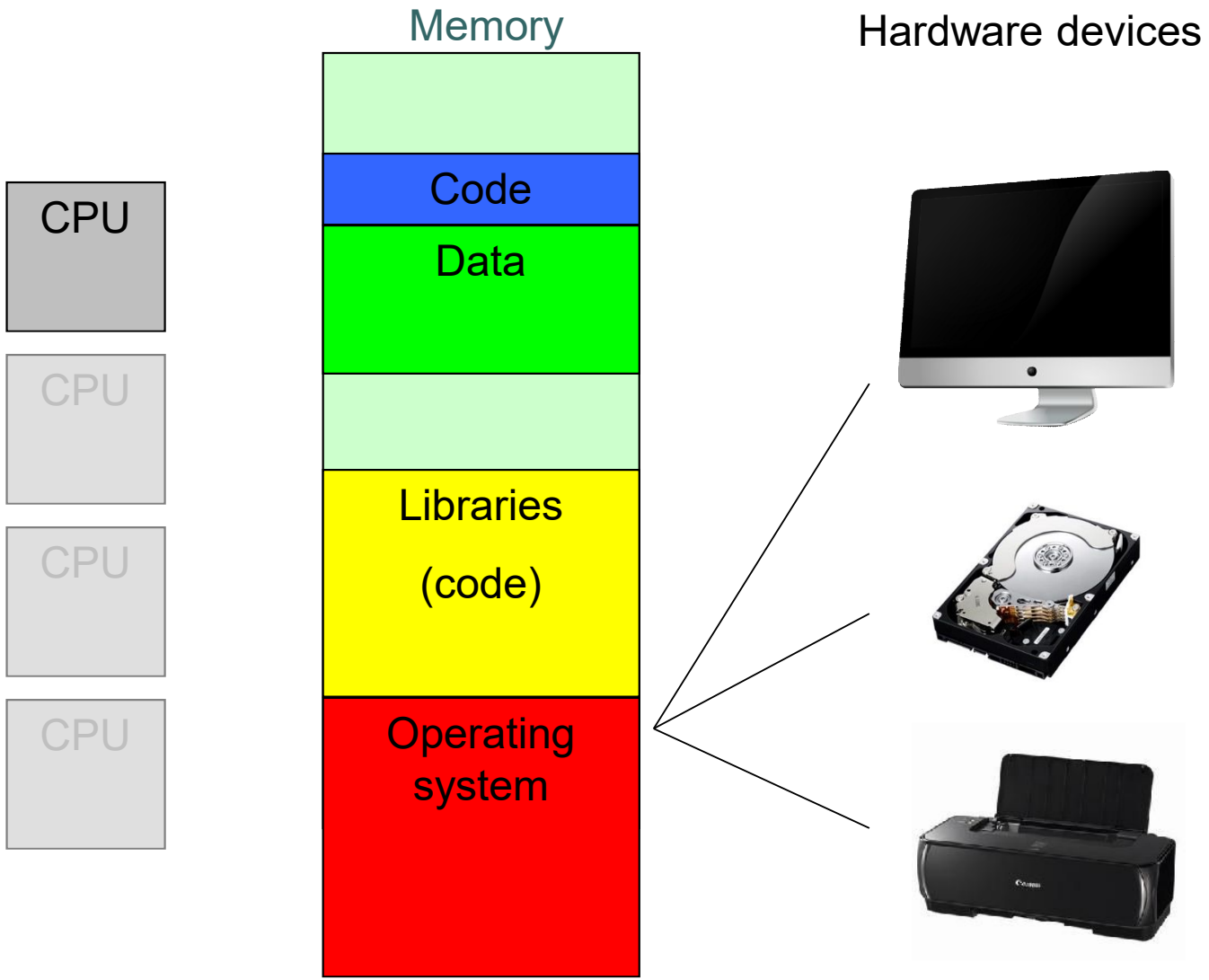
What is an Operating System?

- Intermediary between the user and the hardware
- Controls the execution of application programs
- Is an interface between applications and hardware

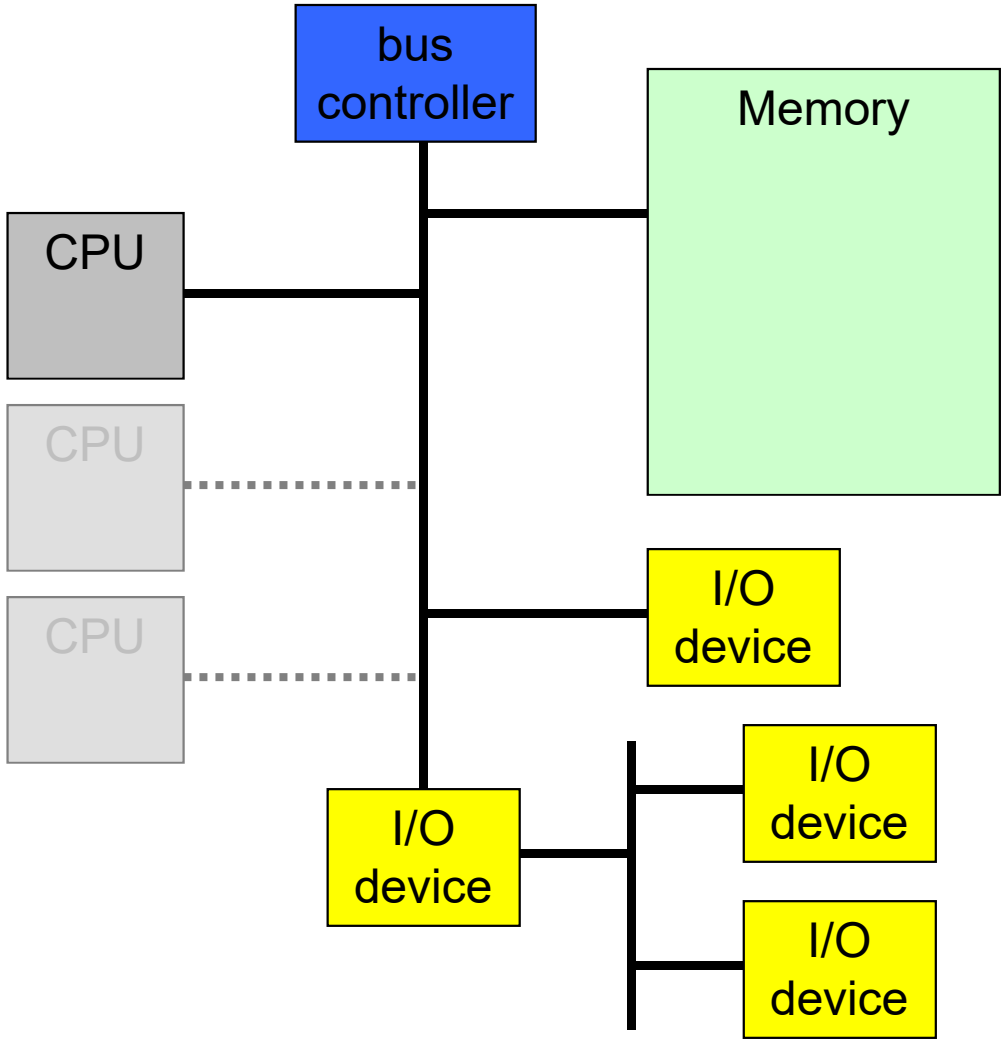
- Operating system goals:
  - Execute user programs
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient and device-independent manner

The Computer:
End-user's view

The Computer: Application programmer's view

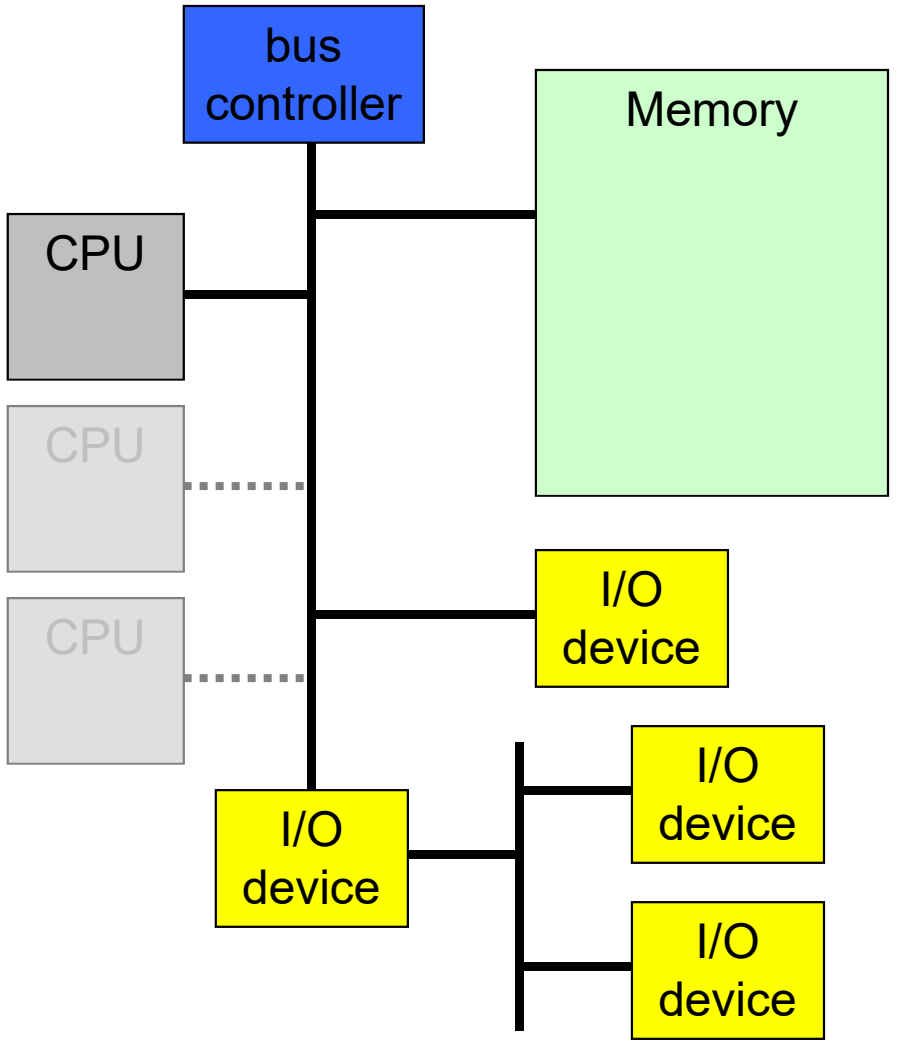


The Computer:
OS programmer's view



Computer Hardware

- Processors
- Main Memory (RAM)
  - Volatile
- I/O devices
  - secondary memory devices
  - displays, keyboards, …, communications devices
- System bus
  - communication among processors, memory, and I/O modules

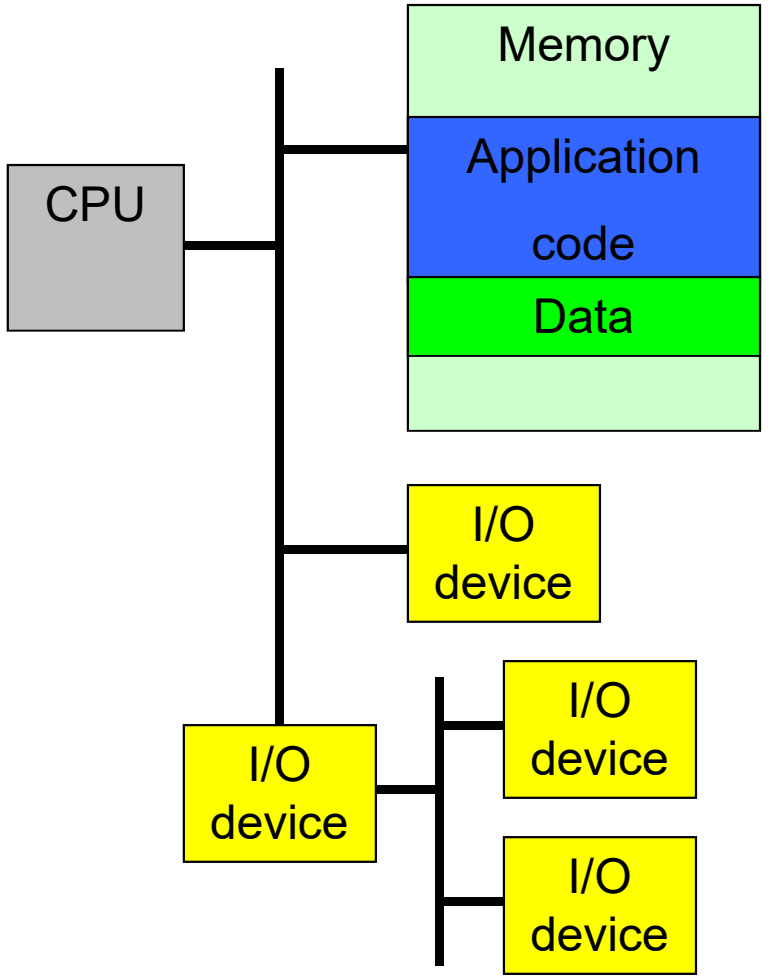


Introduction

- Operating Systems
  - What is an Operating System
  - OS evolution
  - OS details
- Networking
  - The Internet
  - Network protocols
  - Security

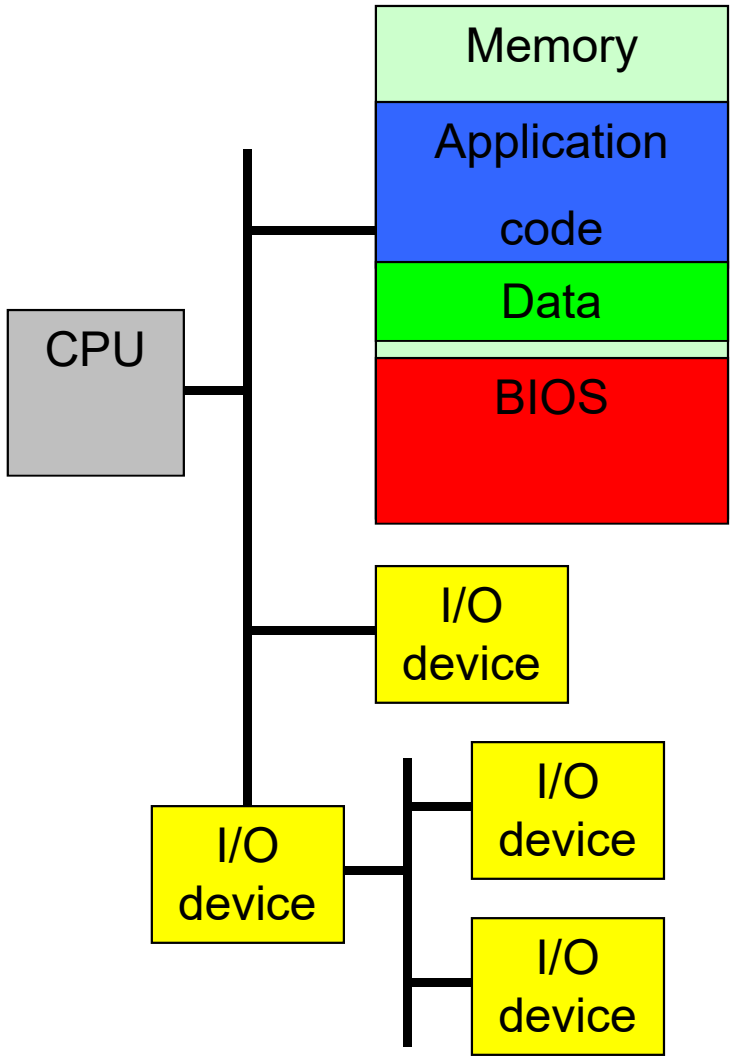
The evolution of operating systems

- The beginning
  - No OS
  - Every application had to do everything by itself
  - One program at a time
  - Help from a program loader only
- Still not so today?
  - Embedded systems and microcontrollers!



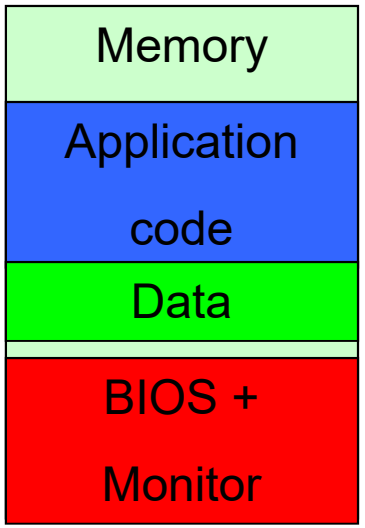
The evolution of operating systems

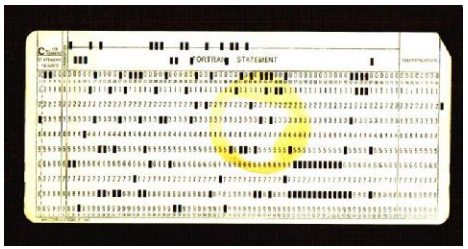
- BIOS
  - Basic Input Output System
  - In Read Only Memory (ROM)
  - Provides interface routines for accessing the hardware
  - Can load one program
- Still, only one program at a time

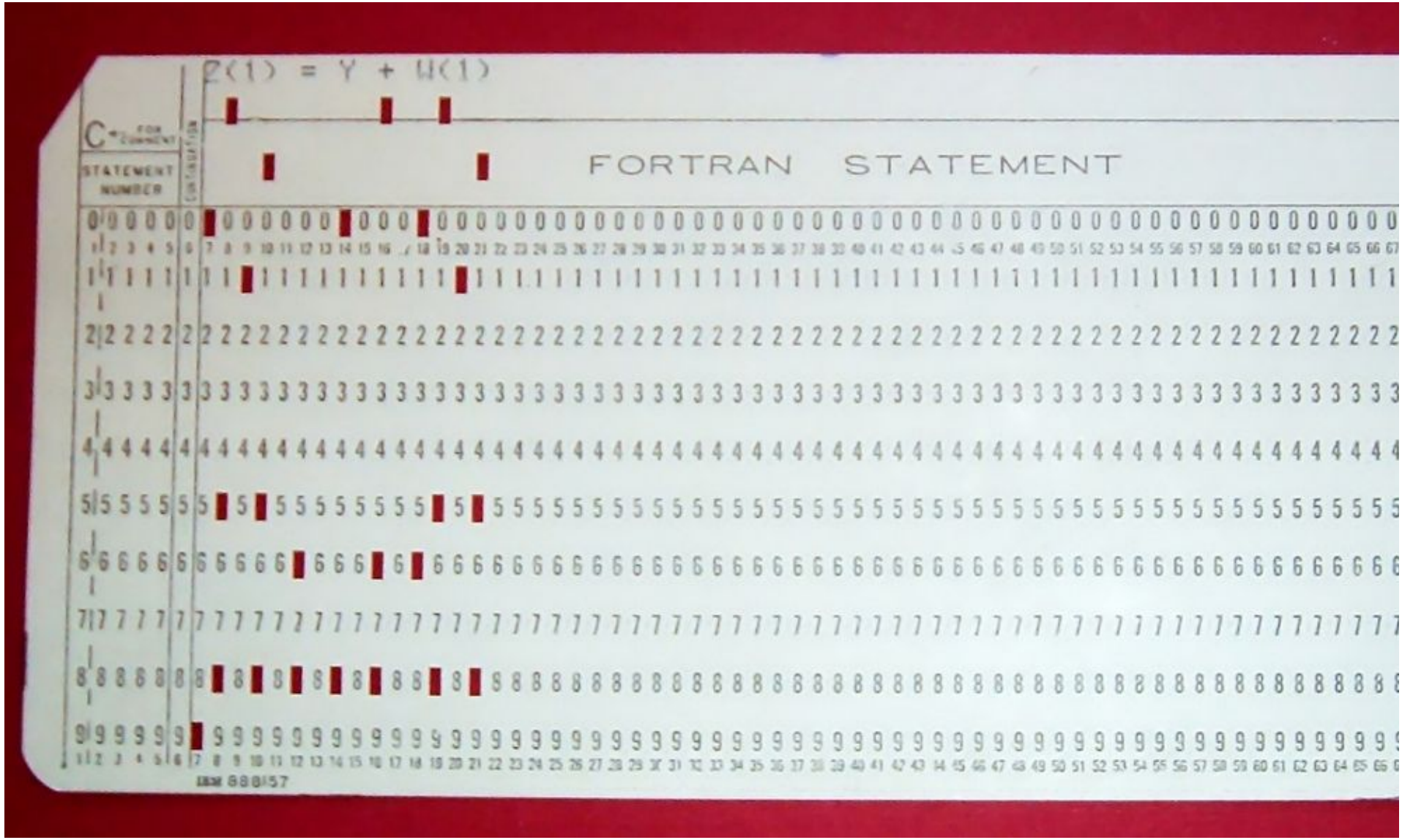


Batch processing

- In the 50s computers were expensive and rare, so efficient utilization was important
- Simple Batch Systems
  - Queue of jobs, run one at the time
  - Monitor
    - Software that controls the running programs
    - Batch jobs together
    - Program branches back to monitor when finished
    - Resident monitor is in main memory and available for execution





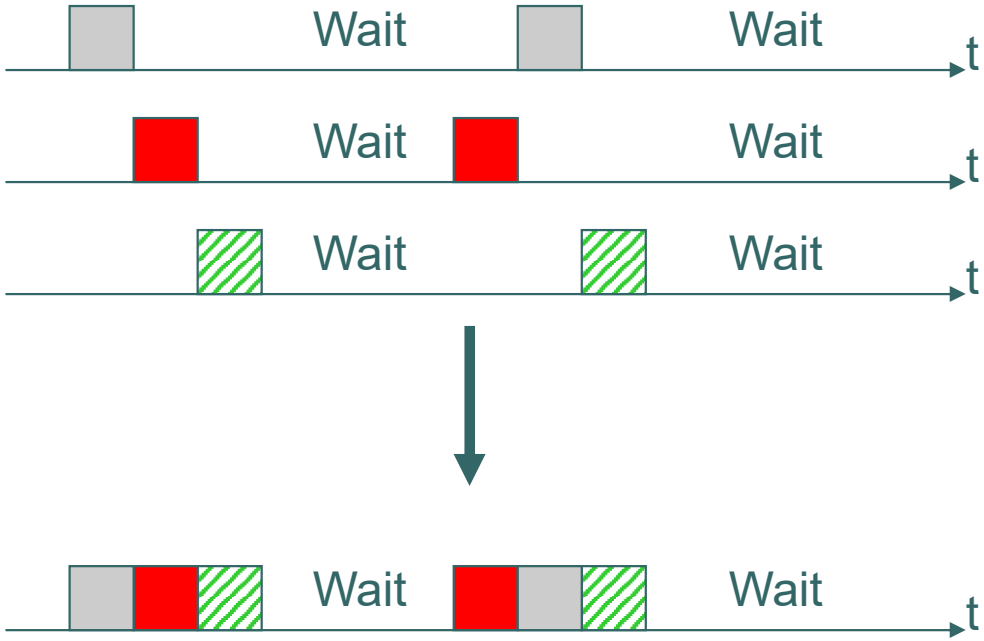


Source: Wikipedia

Single program execution

- One single program is running
- Processor must wait for I/O operations to complete before proceeding
- Leads to poor processor utilization

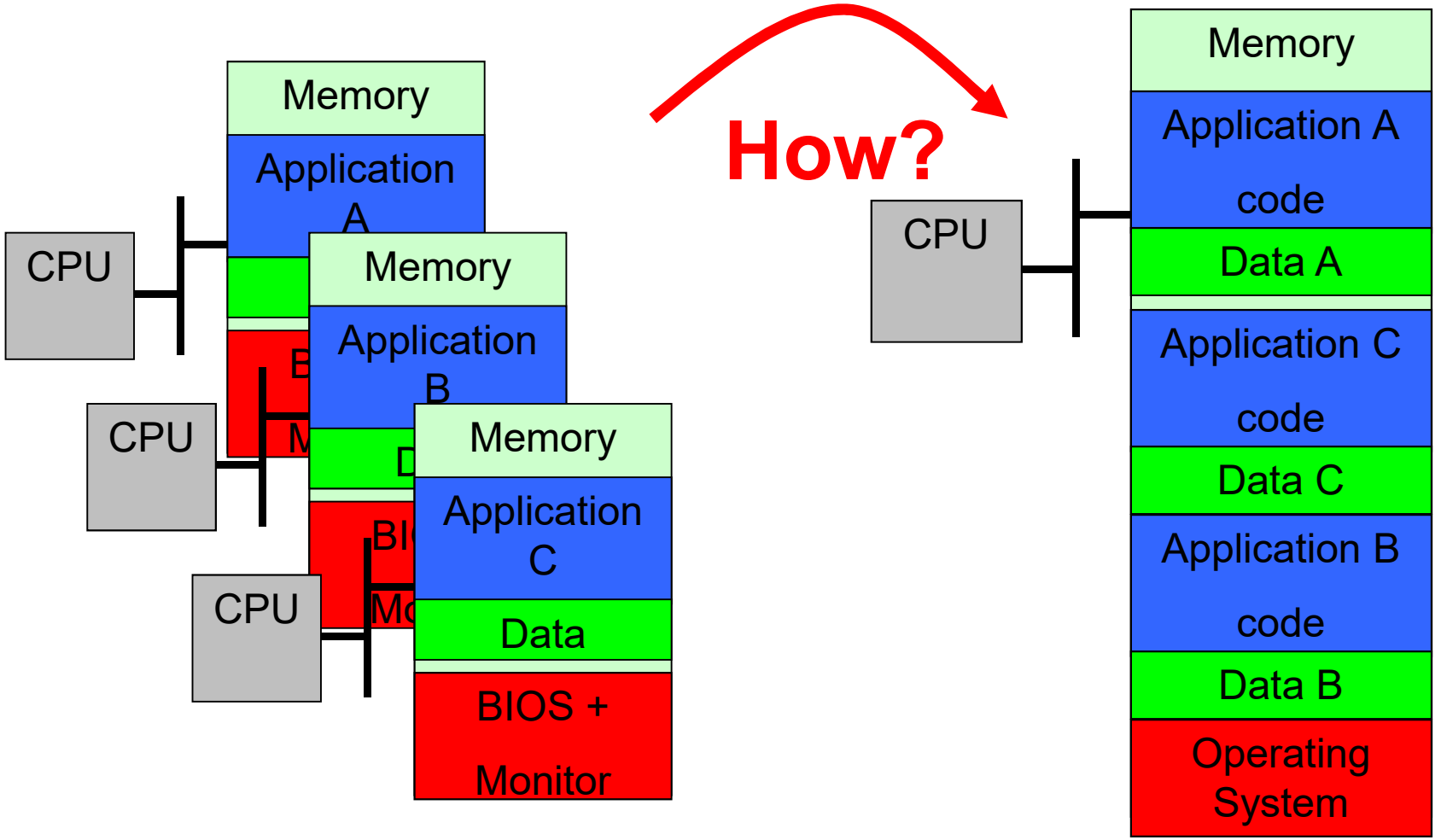
Concurrent program execution (1)
Context (or task) switching



Concurrent program execution (2)
Context (or task) switching

- (Seemingly) concurrent execution
  - Switch jobs at regular intervals
  - Benefits
    - Many applications running at the same time
    - Allows many simultaneous users
    - Interactive programs
    - “Real-time” interaction with user
    - Parallel/concurrent applications
  - Next step
    - Multiprocessor computers

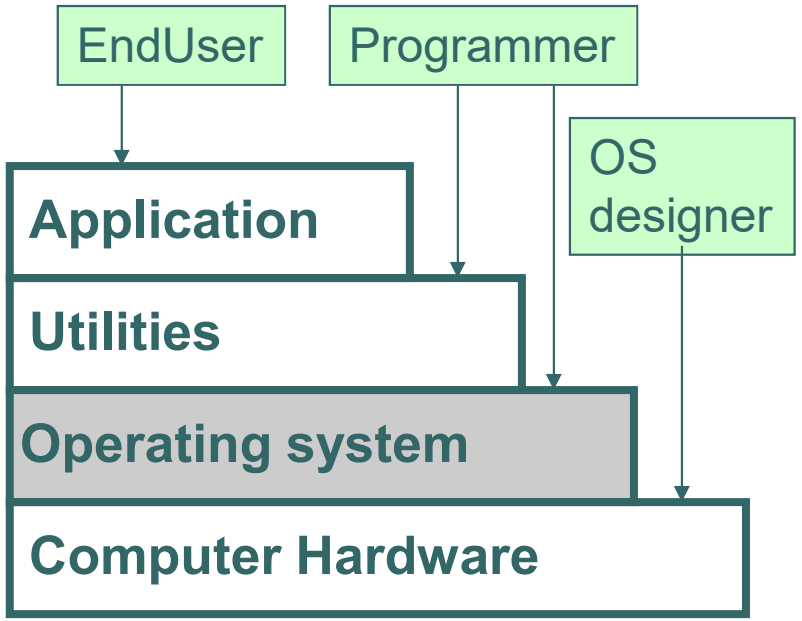
Concurrent execution – The challenge



System Architecture

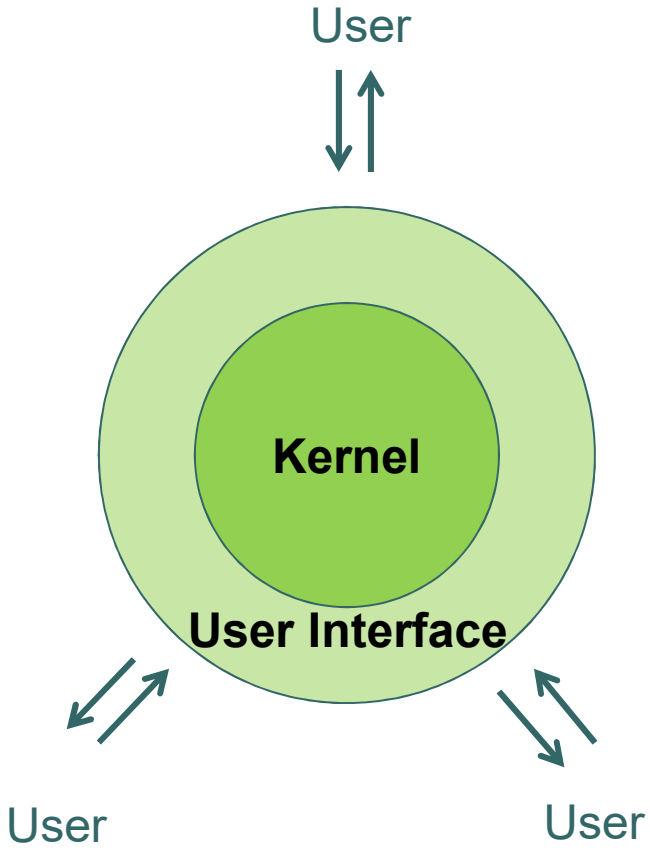
Software in the system:

- Applications
- System software
  - Operating System
    - Shell
      - GUI
      - Command line
    - Kernel
      - The core of the OS
  - Utilities
    - Compilers
    - Interpreters



Services provided by the OS

- Program execution
- Shared access to I/O devices
- Controlled access to files
- Error detection
  - Hardware errors
  - Software errors



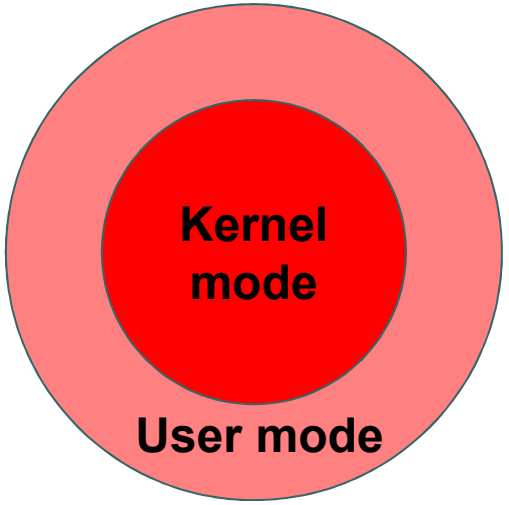
Kernel overview

- Resource control
  - CPU Scheduling
  - Memory manager
  - File manager
  - Device drivers
- BIOS still present at startup
  - Bootstrap
  - Get the operating system running at power on

Kernel Security

- Privileged mode (Kernel mode)
  - Allowed to execute all CPU instructions
  - Access to all I/O devices

- Unprivileged mode (User mode)
  - Not all CPU instructions can be executed
    - e.g. access to memory and I/O devices are restricted



Roadmap

- Operating Systems
  - What is an Operating System
  - OS evolution
  - OS details
    - Context Switching
    - Virtual Memory
    - Resource Competition and Deadlock
    - File Systems
    - Interprocess Communication
- Networking

The answer to concurrent execution:

Processes

P1: MS Word

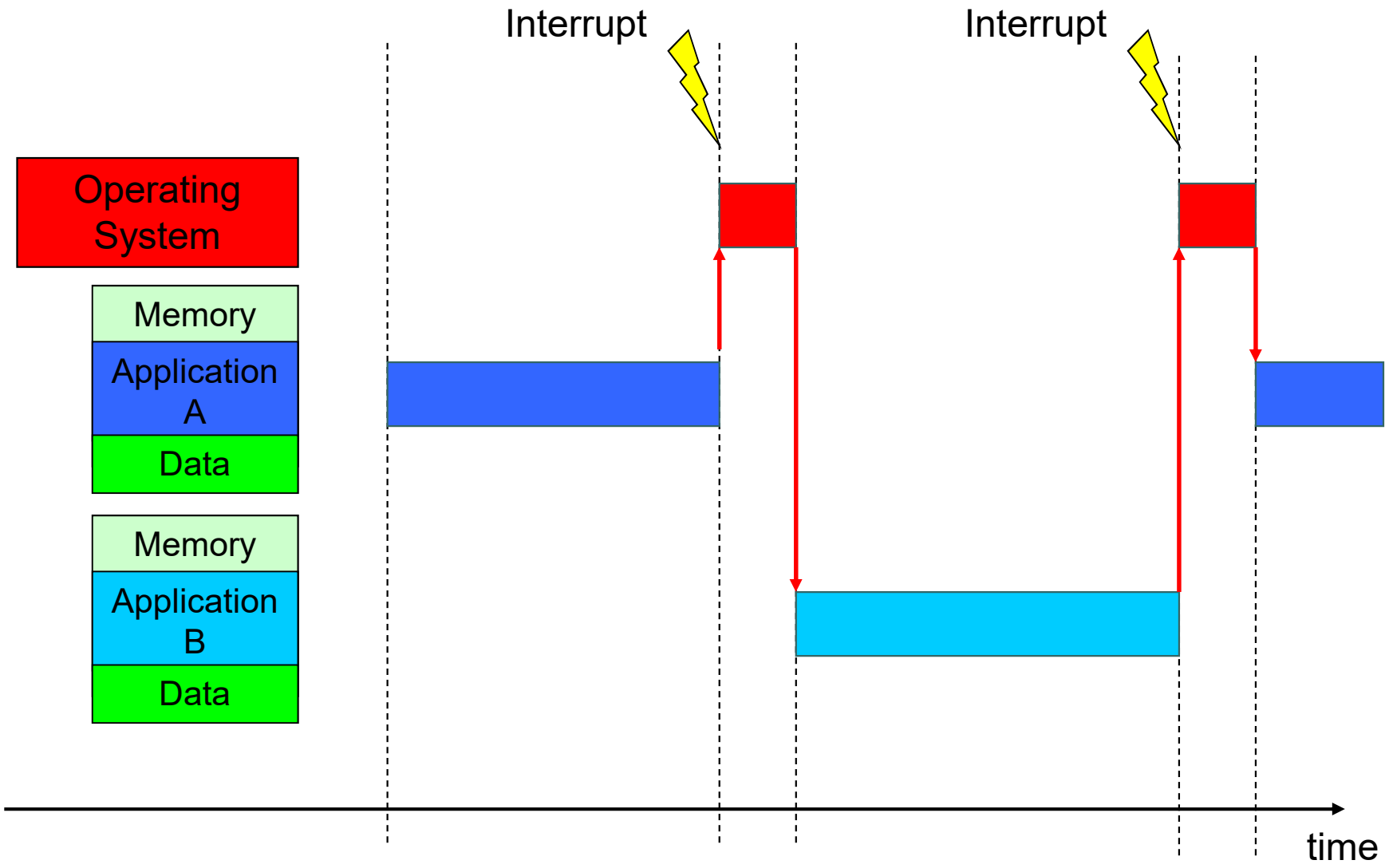
P2: Chrome

P3: …

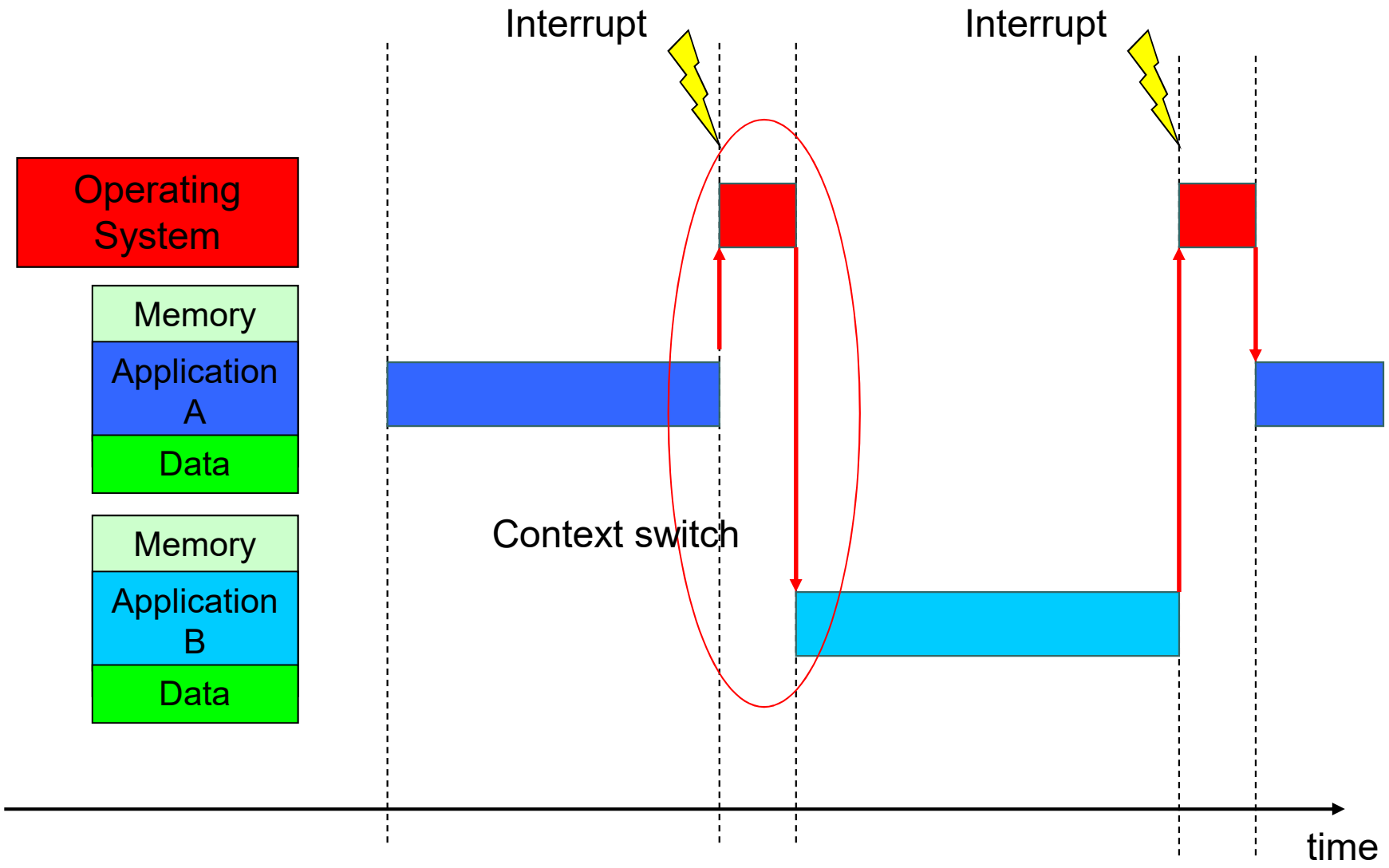
- Process
  - A program in execution
  - The OS presents a simpler ”virtual” computer for exclusive use by the program
- A process includes:
  - Program code
  - Program data and stack
    - The variables
  - State
    - for context switches

} Snapshot of the state of the program in execution

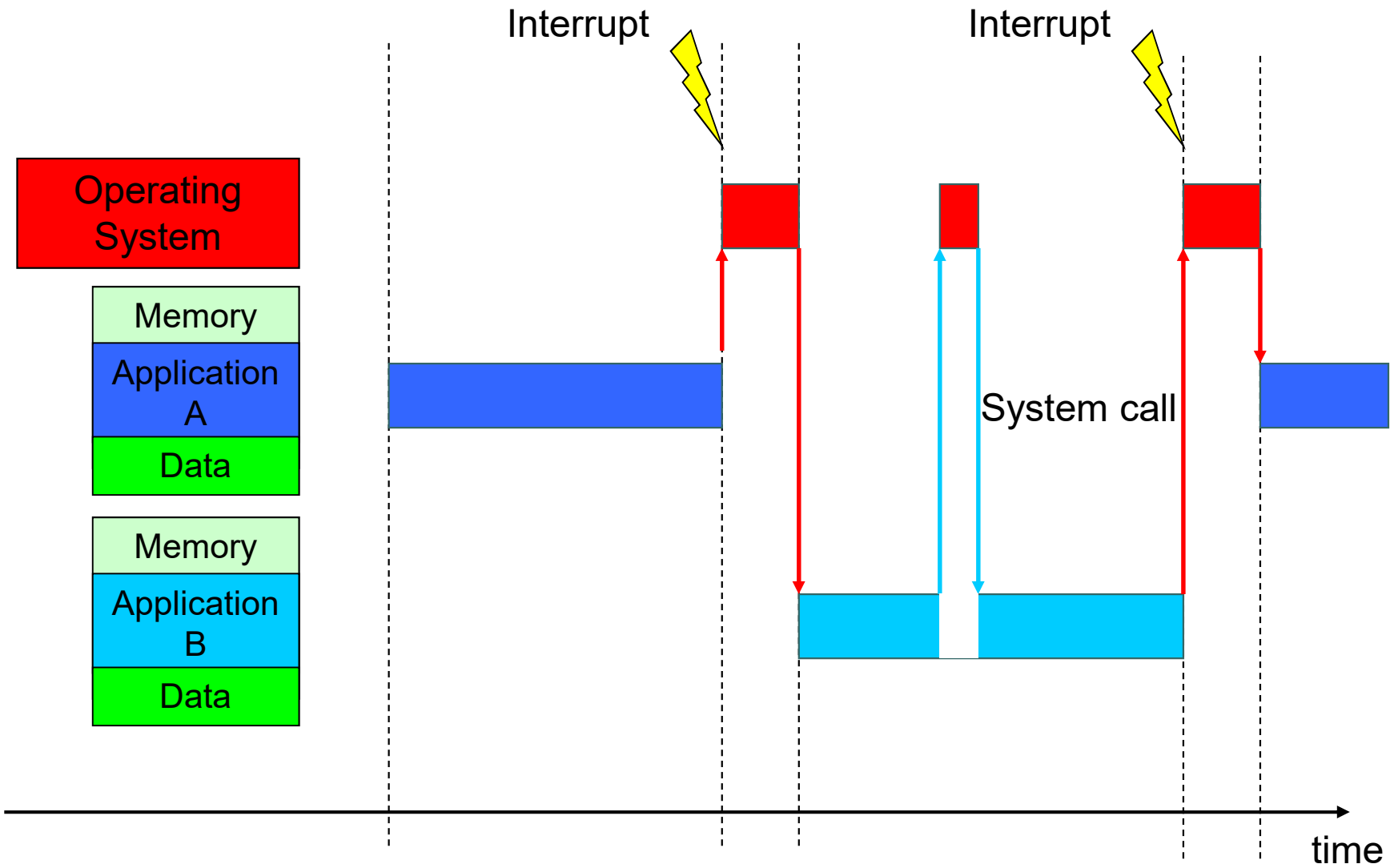
Sharing the processor



Sharing the processor



Sharing the processor



Context switch
Switching Process

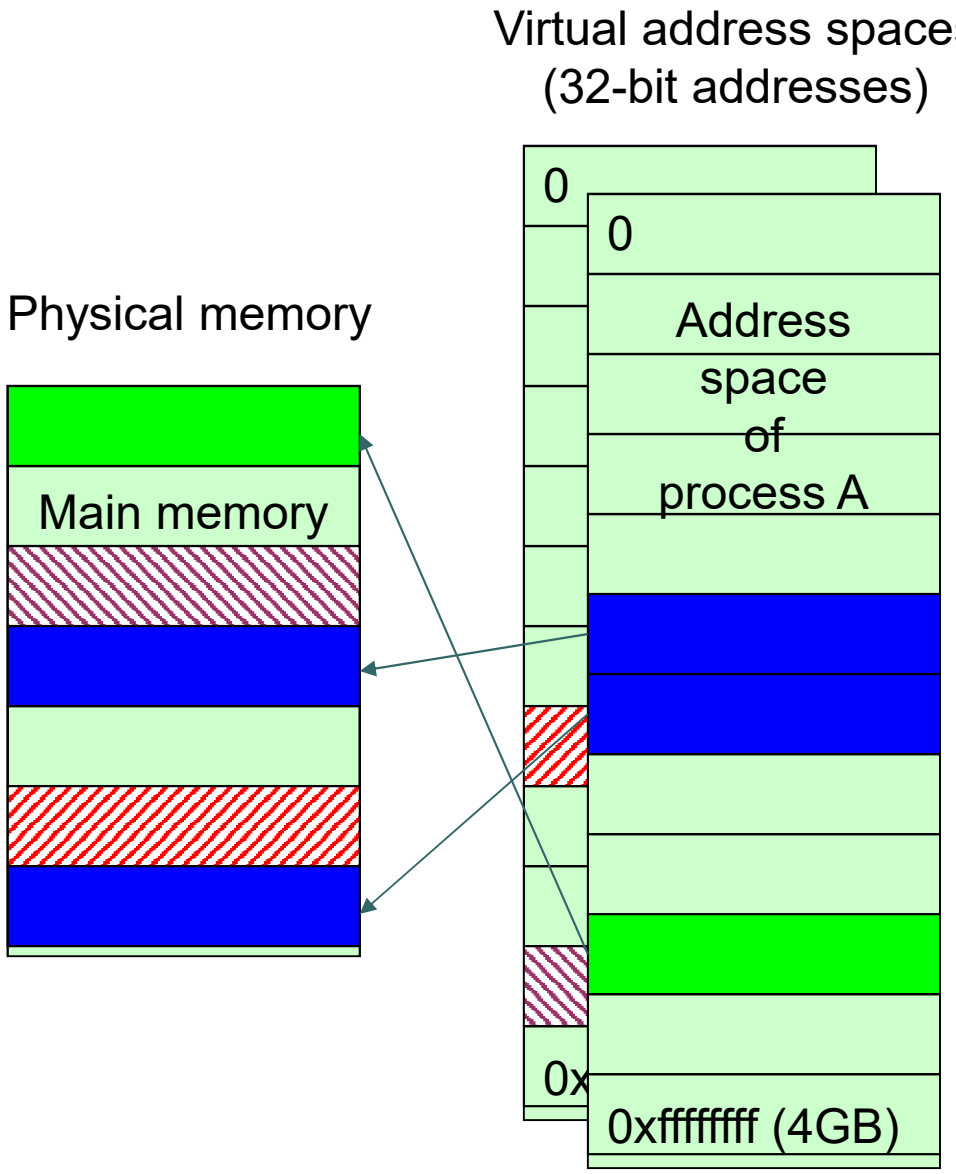
- When switching to another process
  - Save state of old process
  - Load state of new process
- Reasons for switch
  - Interrupts
  - Blocking operations
    - I/O
    - Process synchronization
- Scheduling
  - Choosing the next process to switch to

Virtual Memory

- The illusion of having almost unlimited memory
  - And all the memory for itself
  - 64-bit address => 16.8 million TB
- Main memory shared and smaller
  - OS moves parts of processes to secondary memory
- Protects each process' memory from other processes

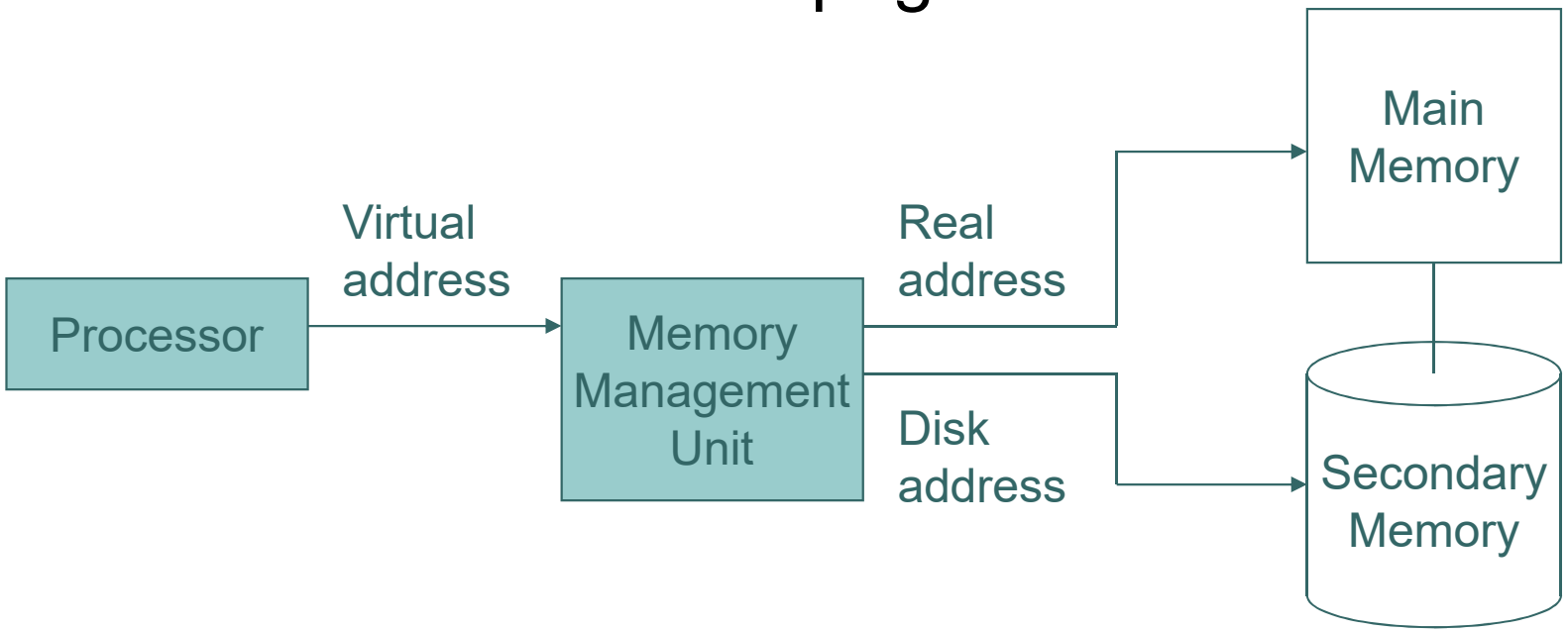
Paging

- Address space and main (physical) memory is divided into fixed-sized pages
- Each page may be located anywhere in main memory
- Or on hard disk if not currently needed



Virtual memory addressing

- A virtual address is the combination of
  - A page number
  - An offset within the page

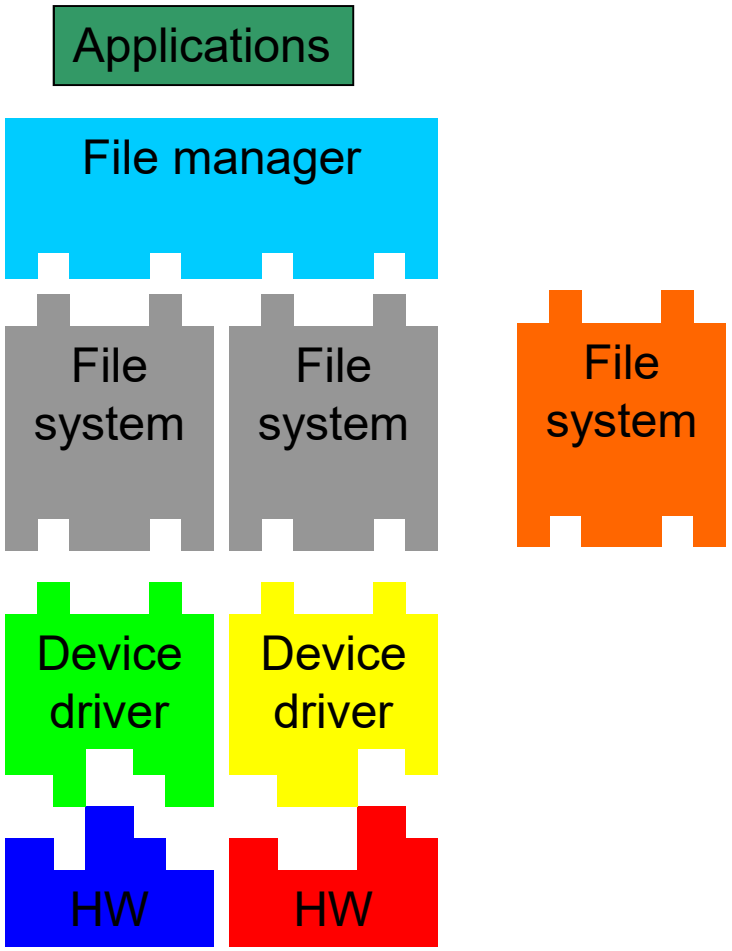


When physically memory filled

- Pages can be written to disk
  - They are ”paged out”
  - Memory becomes available for other pages
  - Chosen pages should be seldomly used
  - If a process uses paged out memory
    - Needs to be read back into main memory
    - Probably ends up in another physical location

File systems

- An abstraction that provides
  - Long-term information storage in named files and directories
  - Allows hierarchical organization of data
  - Standard interface for applications
- The implementation is layered
  - OS file manager and application interface
  - File system implementation
  - Device driver
  - Storage device



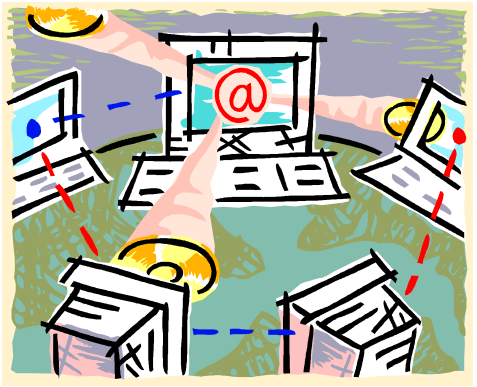
Interprocess Communication

- OS supports communication between processes
- Message-passing (Client/Server model)
  - Clean interface
  - Allows distribution
- Shared memory / shared data structures
  - Locks / Mutual exclusion
    - Acquire / Release
    - Only one process can own it at a time

Roadmap

- Operating Systems
  - What is an Operating System
  - OS evolution
  - OS details
- Networking
  - The Internet
  - Network protocols
  - Security

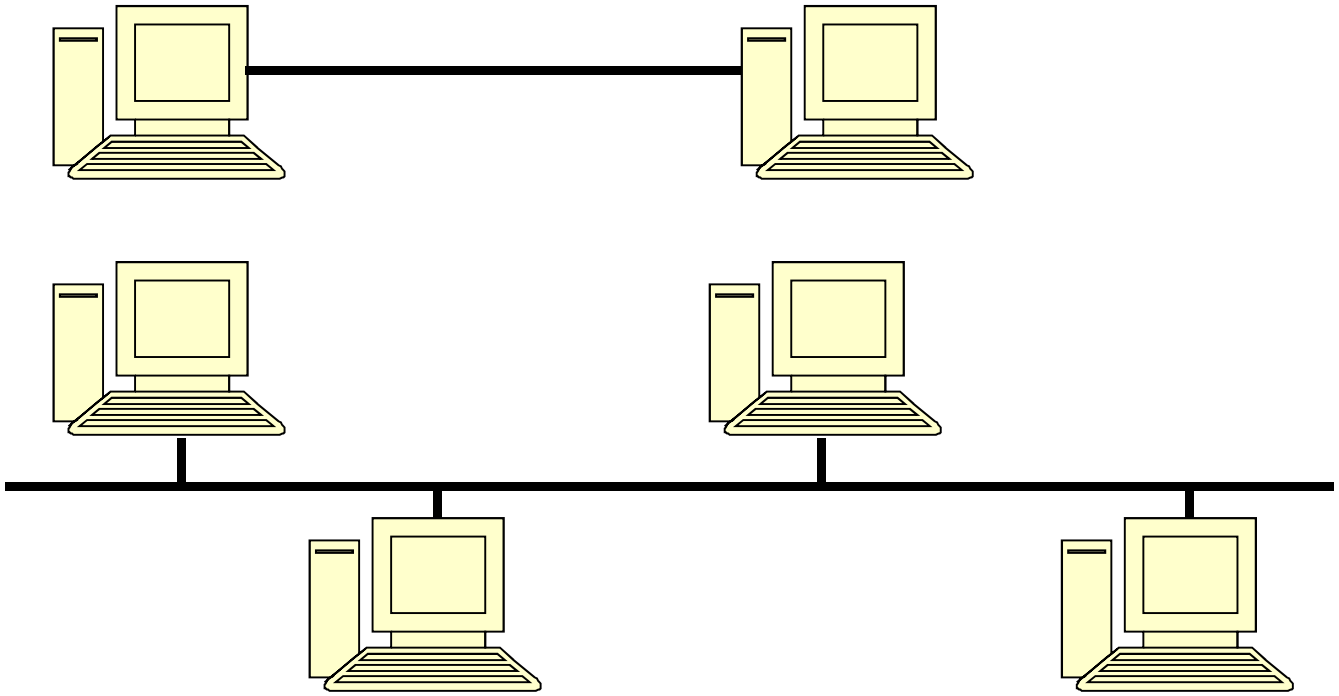
Networking



- Purpose
  - Allow applications (on different computers) to talk to each other

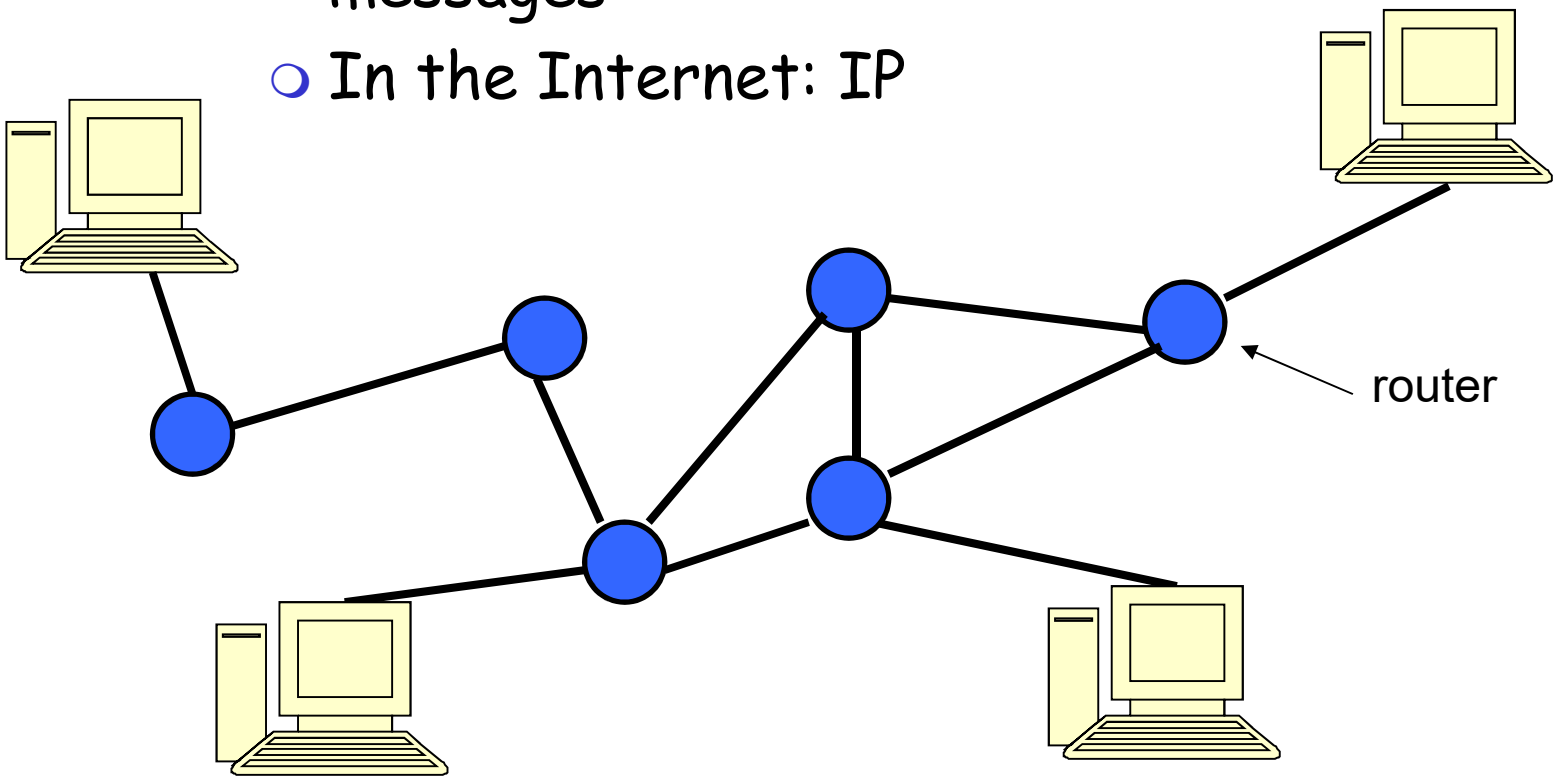
Networks – A bottom up view

- The Link Layer - Computer to computer communication
  - Point-to-point
  - Shared medium (e.g. Ethernet)



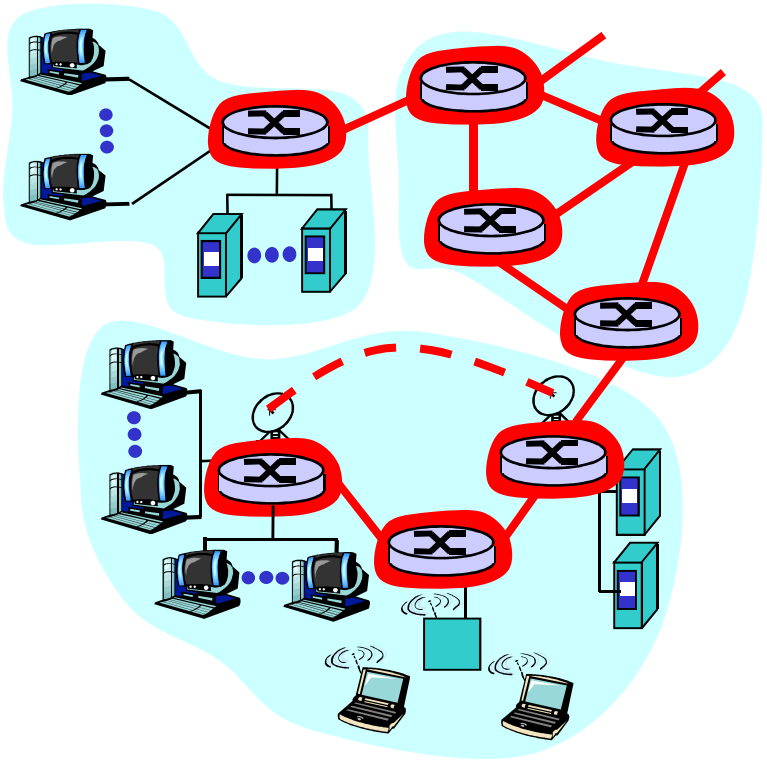
Networks – A bottom up view

- The Network Layer
  - Host-to-host multihop routing of messages
  - In the Internet: IP



A closer look at network structure: The network core:

- mesh of interconnected routers
- fundamental question: how is data transferred through net?
  - circuit switching: dedicated circuit per call: telephone net
  - packet-switching: data sent thru net in discrete "chunks" (The Internet)



Network Core: Packet Switching

each end-end data stream divided into *packets*

- user packets *share* network resources
- resources used *as needed*

store and forward:

- packets move one hop at a time
  - transmit over link
  - wait turn at next link

resource contention:

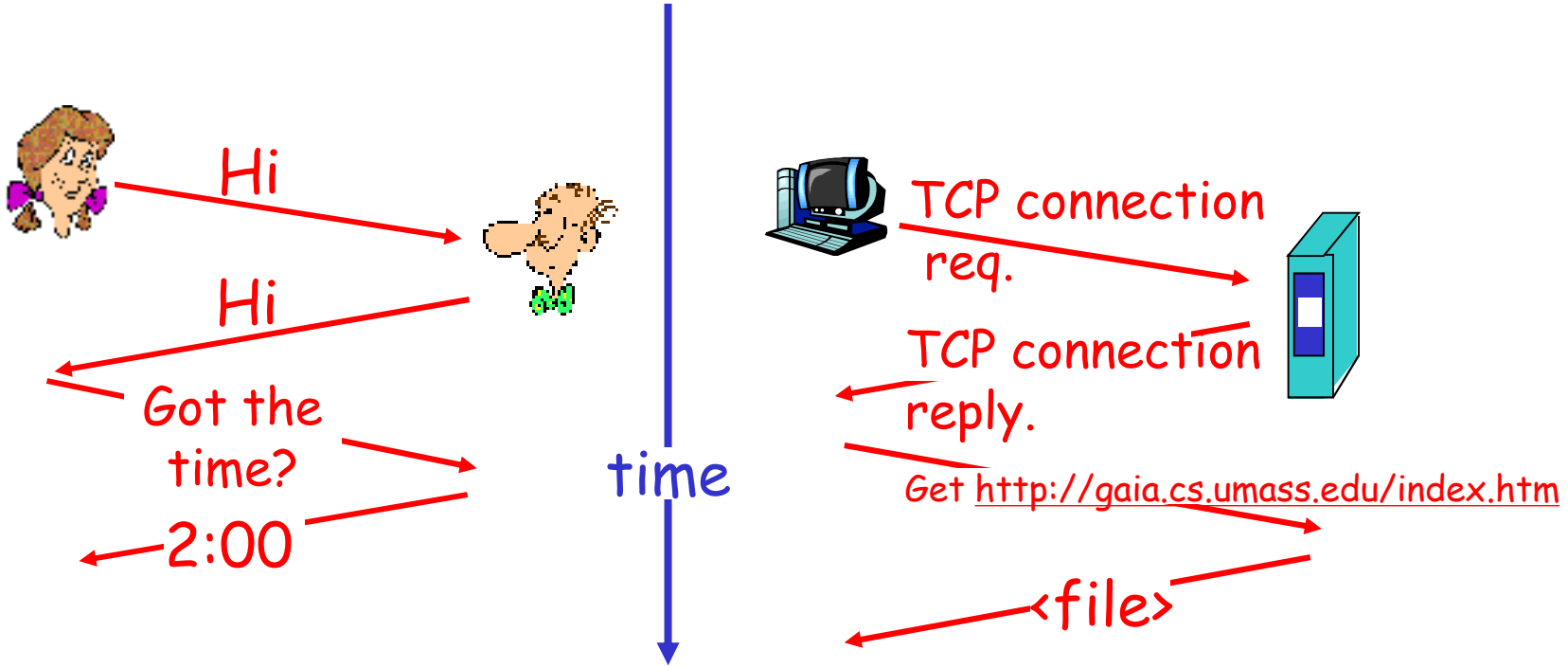
- aggregate resource demand can exceed amount available
- congestion: packets queue, wait for link use

Roadmap

- Operating Systems
  - What is an Operating System
  - OS evolution
  - OS details
- Networking
  - The Internet
  - Network protocols
  - Security

What's a protocol?

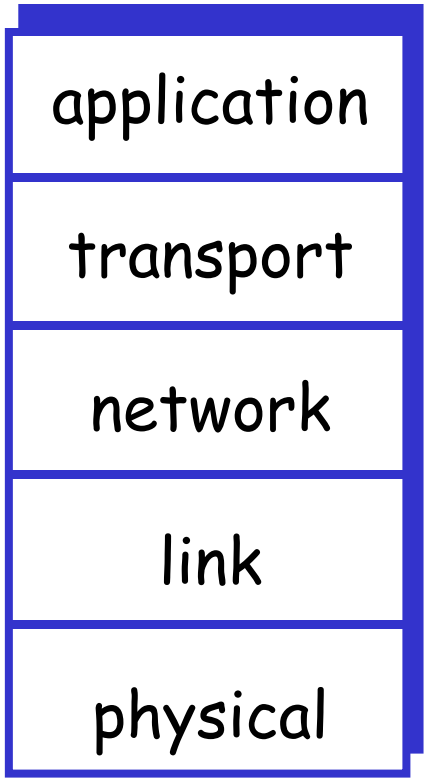
a human protocol and a computer network protocol:



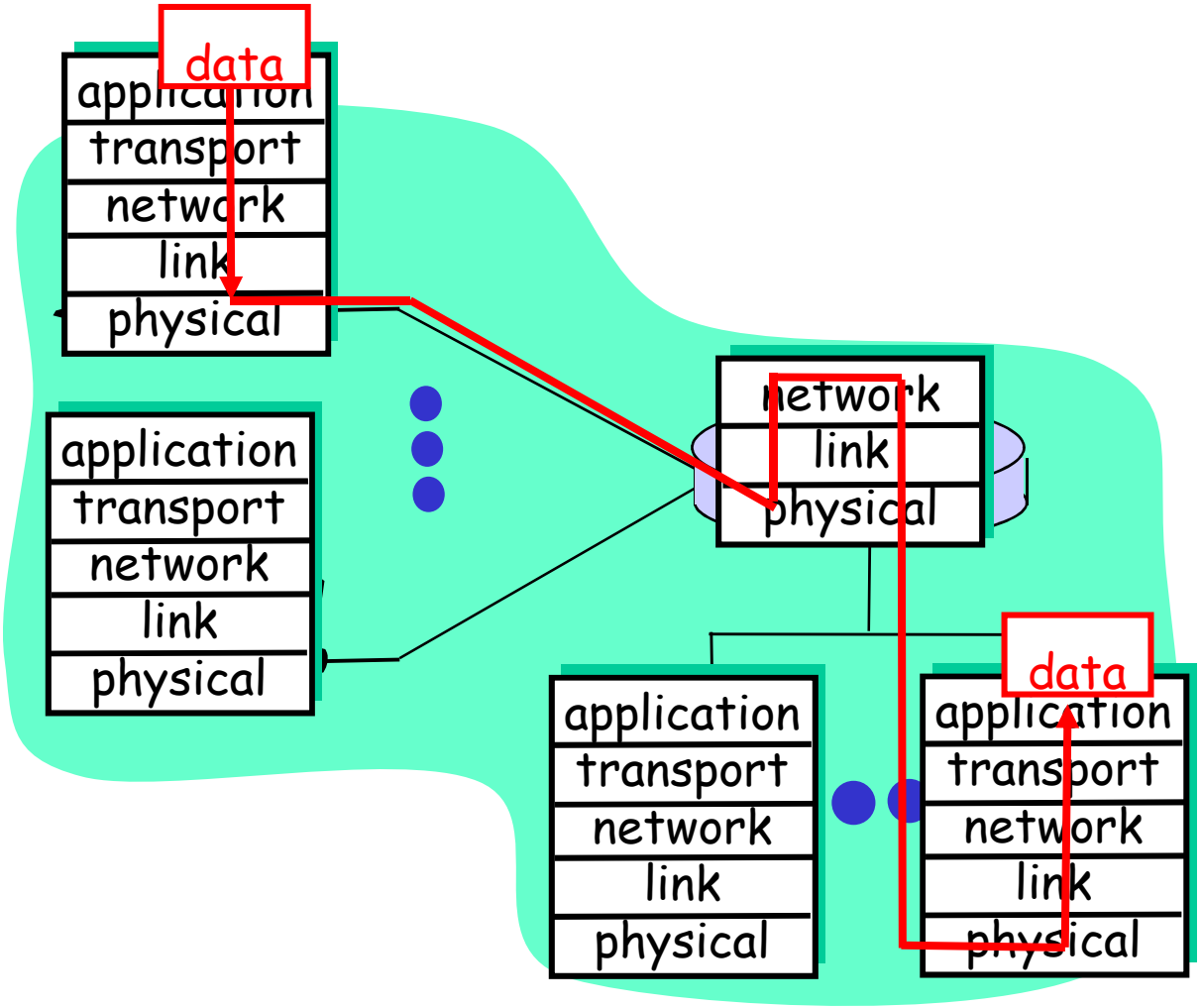
protocols define format, order of msgs sent and received among network entities and actions taken on msg transmission, receipt

Internet protocol stack

- application: ftp, smtp, http, etc
- transport: tcp, udp, …
- network: routing of datagrams from source to destination
  - IP, routing protocols
- link: data transfer between neighboring network elements
  - ppp, ethernet
- physical: bits "on the wire"

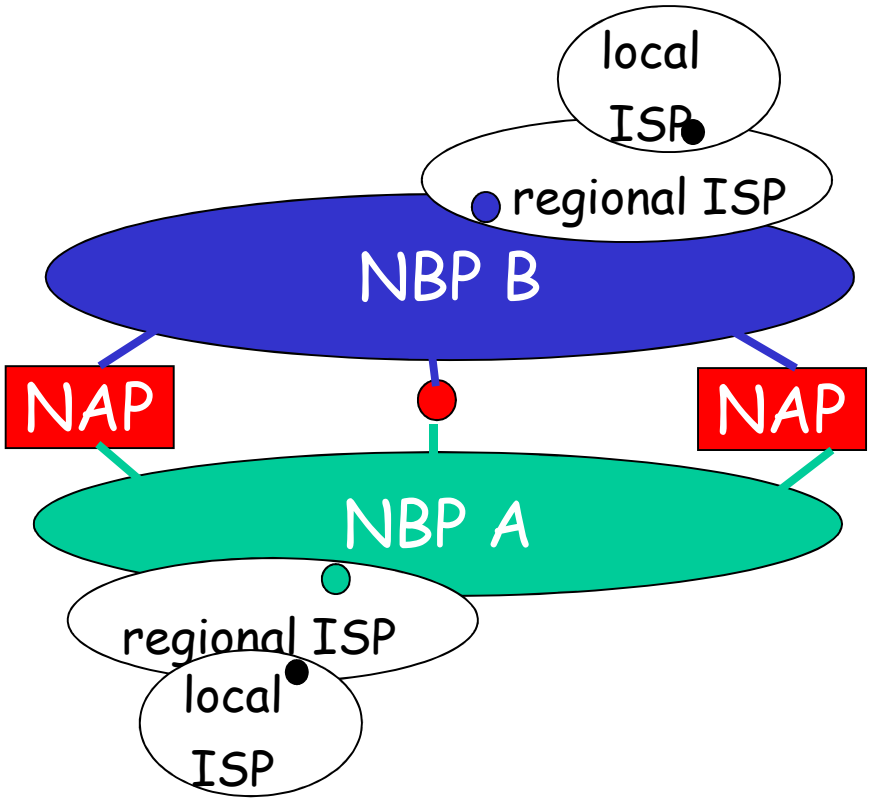


Layering: physical communication



Internet structure: network of networks

- roughly hierarchical
- national/international backbone providers (NBPs)
  - e.g. BBN/GTE, Sprint, AT&T, IBM, UUNet
  - interconnect (peer) with each other privately, or at public Network Access Point (NAPs: routers or (ATM) NWs of routers)
- regional ISPs
  - connect into NBPs
- local ISP, company
  - connect into regional ISPs



Internet Addressing

- An Internet host is identified by an IP-address
  - A unique 32 bit id number (IPv4)
  - Hierarchical w.r.t. routing
- Domain Name Service (DNS) name
  - Translates Human readable name to IP address
  - Eg: www.chalmers.se → 129.16.71.10

IPv6

- “Normal” IP have 4 billion addresses
  - A lot of them are wasted
  - Chalmers alone have 65535 addresses
- We are running out of addresses
- Solution: IPv6
  - 128 bit addresses
    - 50 billion billion billion addresses / person

Networks – A bottom up view

- The Transport Layer
  - Application-to-application communication
  - The Internet
    - TCP: connection oriented + reliable + point-to-point
    - UDP: connectionless + unreliable + point-to-point

Roadmap

- Operating Systems
  - What is an Operating System
  - OS evolution
  - OS details
- Networking
  - The Internet
  - Network protocols
  - Security

What is network security? CIA!

Confidentiality:

only sender, intended receiver should "understand" message contents

- Encryption!

Integrity of Messages:

sender, receiver want to ensure message not altered (in transit, or afterwards) without detection

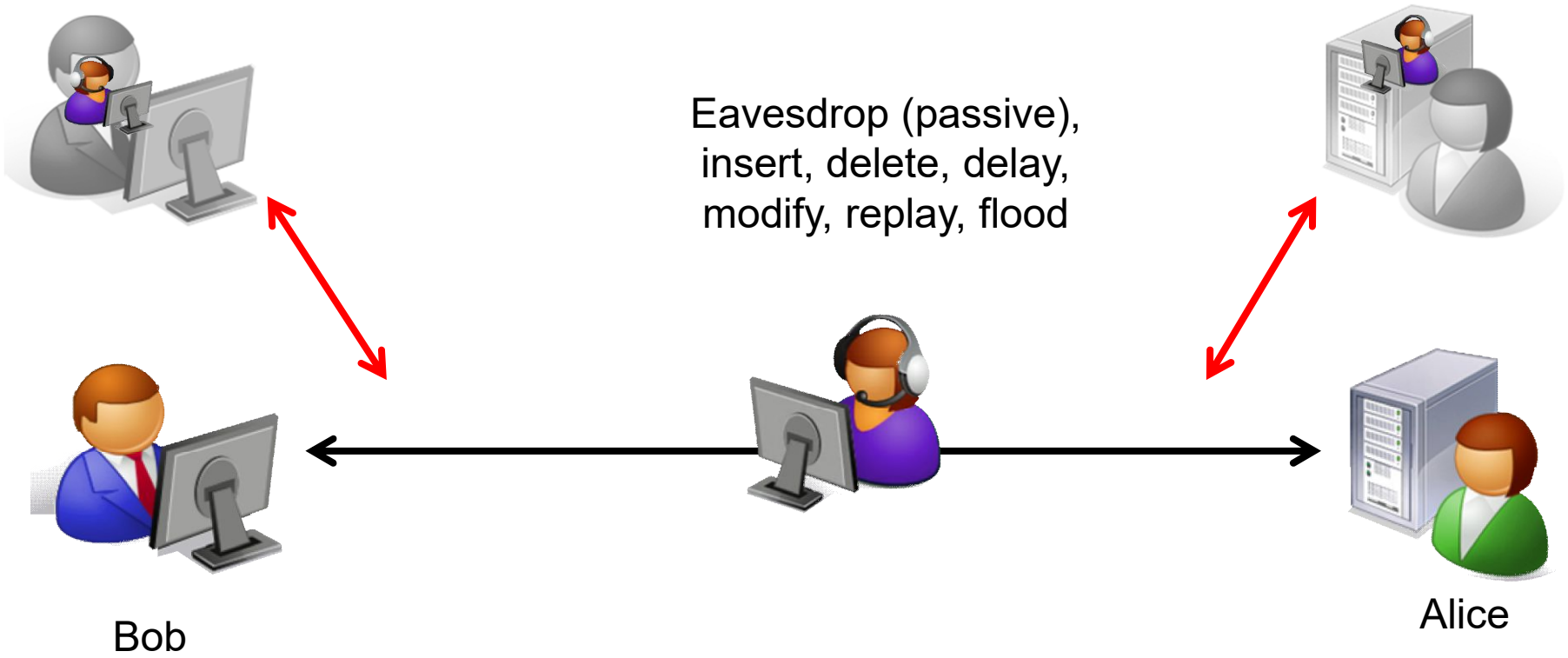
- Digital signatures!

Availability:

System and resources should be available to (authorized) users

Communication threats

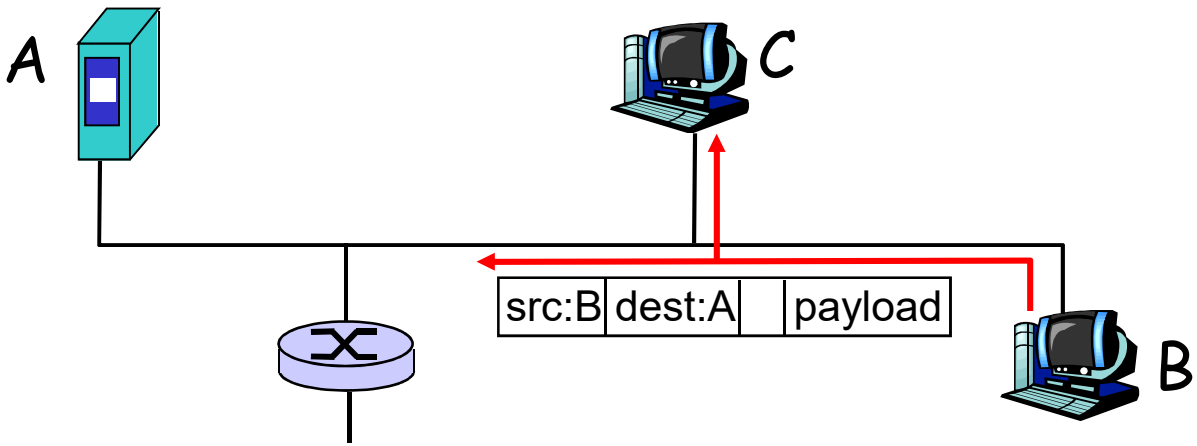
Impersonation (identity spoofing)
Data origin spoofing (insert, probe, scan)



Internet security threats

Packet sniffing:

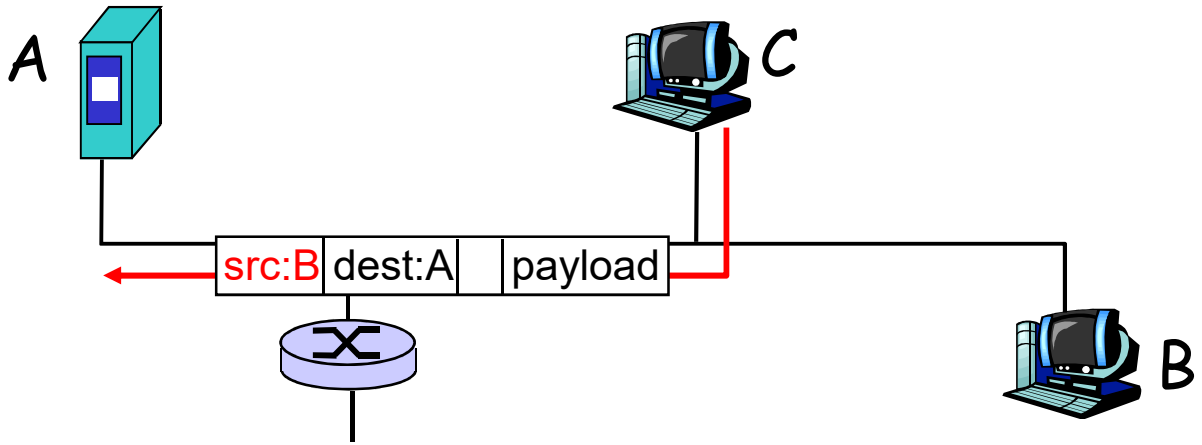
- broadcast media
- promiscuous NIC reads all packets passing by
- can read all unencrypted data (e.g. passwords)
- e.g.: C sniffs B's packets



Internet security threats

IP Spoofing:

- can generate "raw" IP packets directly from application, putting any value into IP source address field
- receiver can't tell if source is spoofed
- e.g.: C pretends to be B



Encryption

- Symmetric
  - Encryption/Decryption with the same key
  - Key distribution a problem
- Public key encryption
  - Encryption with public key
  - Decryption only with private key
  - Key distribution still a problem

Firewalls

firewall

isolates organization's internal net from larger Internet, allowing some packets to pass, blocking others.

Questions?