Bring the power of Mentimeter to PowerPoint

- Seamlessly embed your favorite Menti slide without changing windows.
- Edit and more on Mentimeter.com and sync in real-time.

Log in to use Mentimeter

Work email

o.ascigil@lancaster.ac.uk

Your password

Menti123!

Log in

That account appears to be connected with SSO. Do you want to log in?

Forgot password **Log in with SSO**

or

Don't have an account? **Sign up**

Lancaster University

Introduction

SCC.231 – Computer Networks and Systems

Onur Ascigil Week 1, Lecture 1

Lecturing team

Charalampos Rotsos

- Senior Lecturer, and member of the *Computer Networking research* group
- c.rotsos@lancaster.ac.uk

Lecturing team

Onur Ascigil

- Lecturer and member of the *Systems and Security* research group
- o.ascigil@lancaster.ac.uk

Schedule

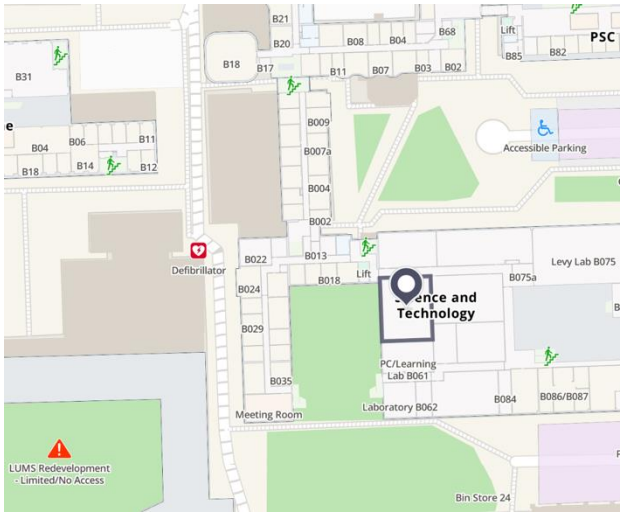
Lectures

Lecture	Time
Lecture 1 of Weeks 1-10	Group 1: Monday 10:00-11:00, Margaret Fell LT Group 2: Mondays 15:00-16:00, LT18 Mgmt school
Lecture 2 of Weeks 1-10	Group 1: Thursday 9:00-10:00, LT18 Mgmt school Group 2: Friday 9:00-10:00, Physics Faraday LT

Schedule

Practical Labs – Science & Technology Building

Monday	**12:00 – 13:00 (Session 1 in SAT B080)** **14:00 – 15:00 (Session 2 in SAT B080)** **16:00 – 17:00 (Session 3 in SAT B080)**
Wednesday	**13:00 – 14:00 (Session 8 in SAT B070)** **13:00 – 14:00 (Session 8 in SAT B070)**
Thursday	**10:00 – 11:00 (Session 5 in SAT B070)** **11:00 – 12:00 (Session 6 in SAT B070)** **12:00 – 13:00 (Session 7 in SAT B070)** **16:00 – 17:00 (Session 4 in SAT B070)**



No labs this week. We are starting next week (Oct. 13)!

Module Aims

From Module Handbook (paraphrased)

This module aims to:

- Instill a deep understanding of how computer operating systems and networks interact to enable the distributed applications that are ubiquitous in the world today.
- Convey the knowledge and practical experience of Internet architecture, network protocols, and operating system principles expected of all computer science graduates.

Topics

First Half of Term

Week 1	Lecture 1	Introduction to Course
	Lecture 2	Introduction to System Design
Week 2	Lecture 3	**Operating System I**: Operating System and Process
	Lecture 4	**Operating System II**: Interprocess Communication and Threads
Week 3	Lecture 5	**Operating System III**: Memory Virtualisation
	Lecture 6	**Internet Protocol Stack:** Layers & Encapsulation
Week 4	Lecture 7	**Application Layer I:** Introduction to Socket Interface
	Lecture 8	**Application Layer II**: Web & HTTP Basics
Week 5	Lecture 9	**Application Layer III:** The Domain Name System
	Lecture 10	**Transport Layer I**: UDP Protocol

Topics

Second Half of Term

Week 6	Lecture 11	**Transport Layer II**: Principles of Reliable Data Transfer
	Lecture 12	**Transport Layer III**: Principles of Congestion Control
Week 7	Lecture 13	**Network Layer I**: Inside a Router
	Lecture 14	**Network Layer II**: Link State Routing
Week 8	Lecture 15	**Network Layer III**: Distance Vector Routing
	Lecture 16	**Link Layer I**: Link Layer functionality
Week 9	Lecture 17	**Link Layer II**: Addressing and Switching
	Lecture 18	**Cloud I**: Introduction to Virtualisation
Week 10	Lecture 19	**Cloud II**: Introduction to Cloud Computing
	Lecture 20	Revision Lecture

Lab Plan

First Half of Term

Week 2	Processes APIs in the Linux Kernel
Week 3	Introduction to Systems Programming (IPC)
Week 4	Introduction to network monitoring and testing
Week 5-6	Building a network application/Introduction to Mininet
Week 7	Coursework Support
Week 8-9	Configuring an OSPF network
Week 10	Coursework submission

Lab Setup

Lancaster University

- DevContainers + Visual Studio Code
  - Containers are a popular mechanism to create reproducible development environments.
  - DevContainers integrate a container environment with VS.Code.
  - We will use a custom 231 container to develop lab exrcersises and coursework.
- Mininet
  - Create virtual network topologies in lab machines and your personal computers.
  - Describe topologies in Python.
- Python
  - Lab exercises and coursework will use Python 3.
  - Build on SCC.111 knowledge from summer term.

Assessment

Exam – 70% of Overall Weight

- Exam takes place during the summer term
- Any topic in lectures or labs may be included in exam, unless stated otherwise
- Last year: in-person computer-based, 2hr.
- Note: Some topics have changed this year, and some past exam questions are not relevant.
    - We will discuss this in *revision* lecture in week 10.

Assessment

Coursework – 30% of the Overall Weight

- Network Application Development
  - *Multiple parts – approximately a part per week*
  - 30% of the overall weight
  - To be released at the end of Week 5

Due: Friday of Week 10

Assessment

Coursework – 30% of the Overall Weight

Coursework Support:

- Lecture 5 covers socket programming – Monday of week 3
- One week per task
- Labs 6-10 will include new activities to apply and test lecture knowledge and offer coursework support.

In-person Attendance Check-in

iLancaster

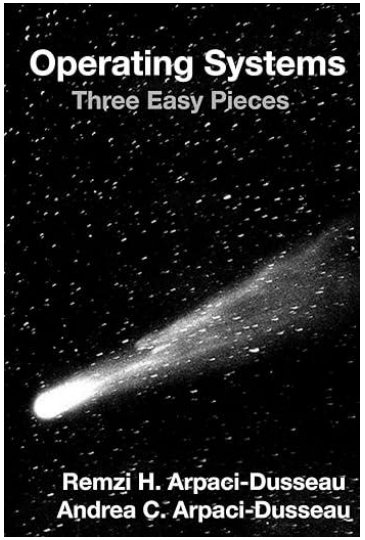
- Ensure you have the most up-to-date version of iLancaster
- Then, to check in automatically:
  - **Turn on Bluetooth and Location Services on your** device
  - If you want to know when you're checked in, **turn on Notifications** on your device

Required Reading (I)

Lancaster University

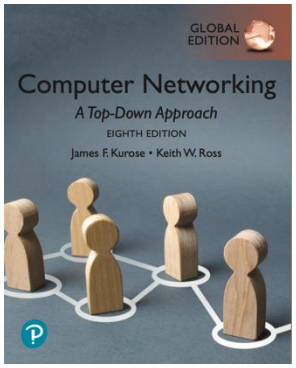
- "Operating Systems: Three Easy Pieces"
  - Remzi & Andrea Arpaci-Dusseau

  - Each lecture in our course is linked with a corresponding chapter
  - Digital version available for free:
  - https://pages.cs.wisc.edu/~remzi/OSTEP/

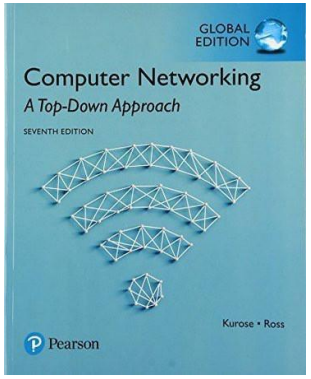


Required Reading (II)

- "Computer Networking: A Top-Down Approach"
  - Kurose & Ross
  - 8th, Global Edition (7th is close enough for the purposes of this course)
  - Each lecture in our course is linked with a corresponding chapter
  - Digital version available through Library:
    - https://bit.ly/2FunP0a (8th Edition)
  - Physical copies too:
    - https://bit.ly/2ALdtVu



or



Lancaster University

A word on plagiarism

- Plagiarism is passing off someone else's work as your own, including:
  - Submitting (e.g.) code that someone else wrote
  - Paying someone else to do it for you
  - Working on non-group work as a group, and then submitting it as individual work
  - Sharing code that you (or others) subsequently adapt/ obfuscate

What do we expect from you?

- **Integrity** (no plagiarism, no faking of results)
- **Effort** (active learning):
  - Come to lectures (it really helps!)
  - Come to labs (they are compulsory)
  - Get the textbook and use our resources effectively
  - Take notes (there's clear evidence that hand-written notes taken by students are more effective than handouts!)
- Take notes (again, because the slides are not enough when try to revise, really…!)
- Read around the subject and try things out for yourself
  - Ask us questions in lectures/labs
  - Plan your time and coursework carefully

Lancaster University

What can you expect from us?

- We'll do our best to:
  - make all our lecture material available on Moodle
  - give you appropriate references to follow up
  - check personally that the labs are running smoothly and the TAs are offering good support
  - arrange extra support if you've already tried the normal routes (books -> web -> TAs)
  - respond to email (ideally as a last resort!)
    - we get more email than we can handle, and have a lot of teaching/research/admin commitments, so we're not always free (even though we are in our offices)!

Bring the power of Mentimeter to PowerPoint

- Seamlessly embed your favorite Menti slide without changing windows.
- Edit and more on Mentimeter.com and sync in real-time.

Log in to use Mentimeter

Work email

Your password

Log in

Forgot password Log in with SSO

or

Don't have an account? **Sign up**

Lancaster University

What is this module about?

What is a System?

Lancaster University

- A **system** is a collection of **components** (such as hardware and software) that work together to achieve a **specific goal**.
  - A system is **more than the sum of its parts** — its behaviour comes from the **interactions** between components.
  - Systems engineering involves designing, integrating, and managing complex systems throughout their entire life-cycle.

- Explore three standard computing systems:
  - *Operating Systems*, *networks/the Internet*, and *cloud infrastructures*.
- Learn how to enhance performance by utilizing system design patterns.

The Operating System

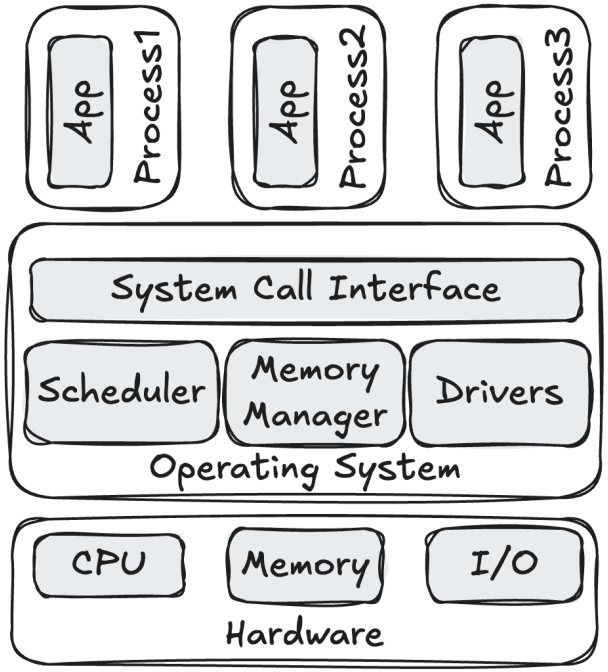
Lancaster University

Components

- Hardware resources → CPU, memory, devices
- Communication → IPC channels (pipes, message queues, shared memory, signals)

Goal

- Share resources (e.g., CPU, memory, disk, devices) safely across **multiple processes**
- **Simplify** the programming API
- Allow processes to **communicate** without managing low-level details



The Network / The Internet

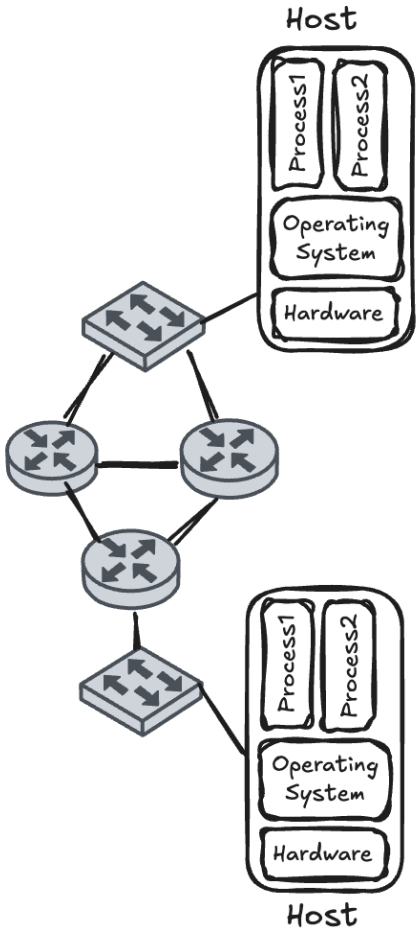
Lancaster University

components

- Physical connections → **links and interfaces**
- Communication across machines → **sockets & protocols (TCP/UDP, IP, DNS)**
- End-to-end data transfer → **logical channels between processes**

Goal

- Communication API **similar to** local IPC
- Improve **scalability,** allow remote processes to cooperate and form larger systems
- Simplify the use of heterogeneous hardware and networks (LAN, WAN, Internet)



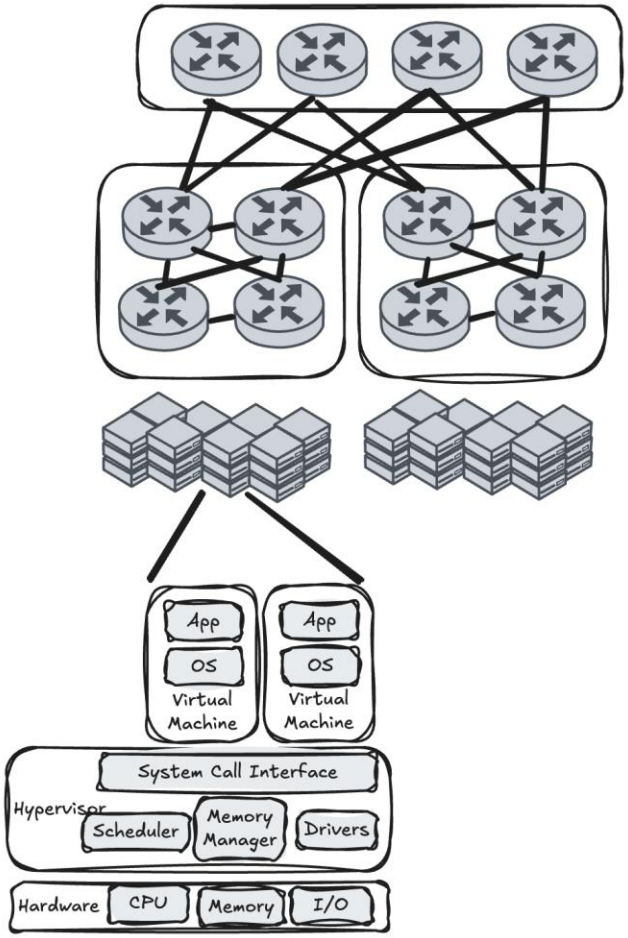
The Cloud

components

- Entire **server clusters** → Virtual Machines (VMs) and Containers

Goal

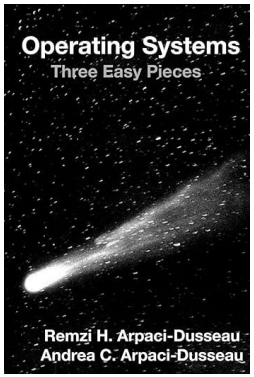
- **Elastic, on-demand** resources, improved **scalability**
- Hide management **complexity**
- **Isolate** hosts from different users running on the same hardware



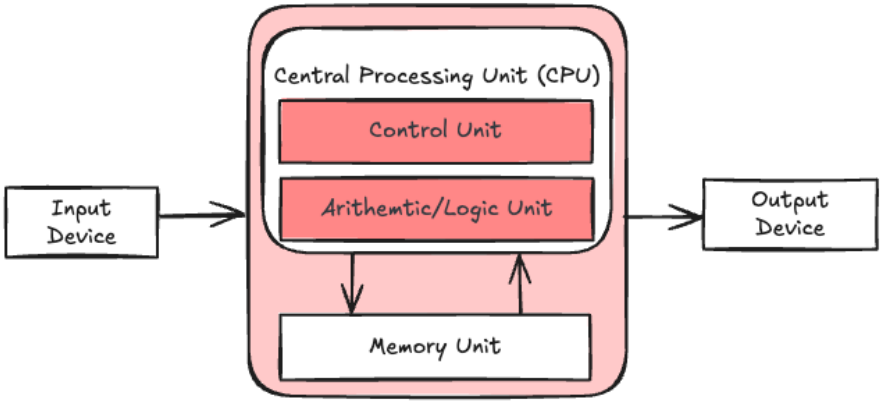
Lancaster University

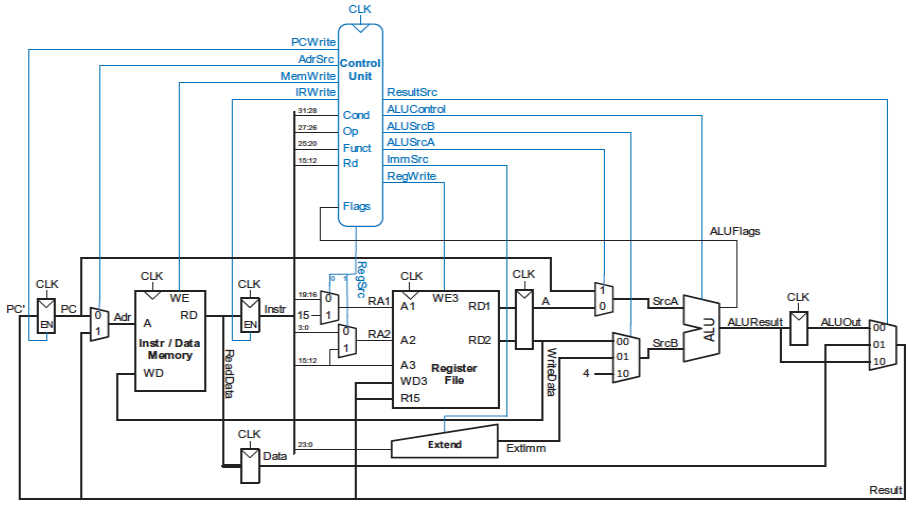
What is an Operating System?

Chapter 1



From Computer Architecture to …

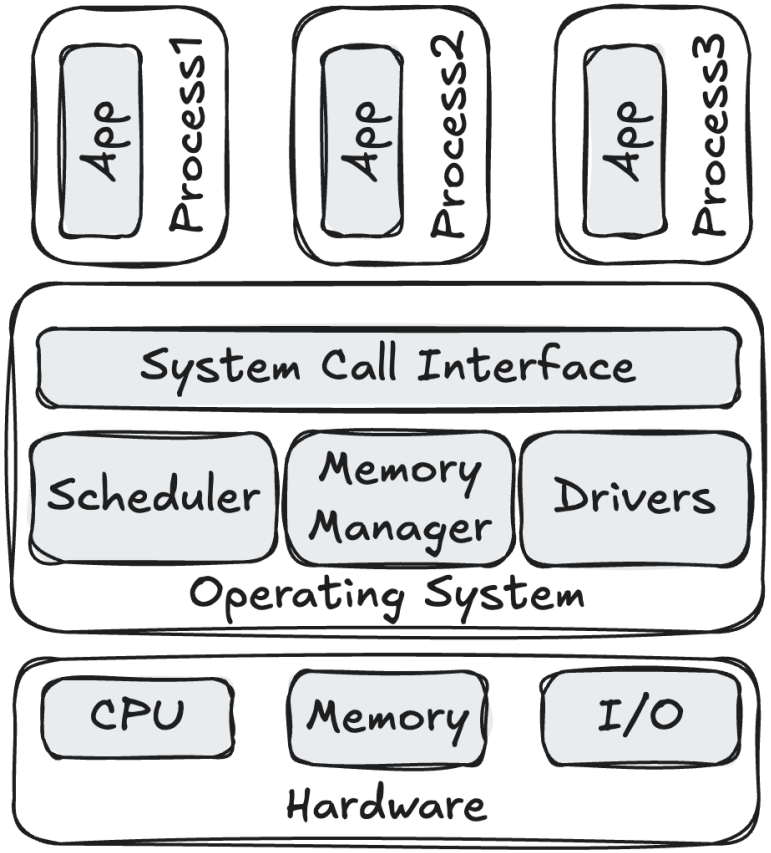





```
.syntax unified
.global func
.text
.thumb_func

func:
@ ----------------
@ Two parameters are in registers r0 and r1
adds r0, r0, r1 @ Add r0 and r1, result in r0

@ Result is now in register r0
@ ----------------
bx lr @ Return to the caller
```

Caution: You can relax, no micro:bit will be involved in this course

Operating Systems

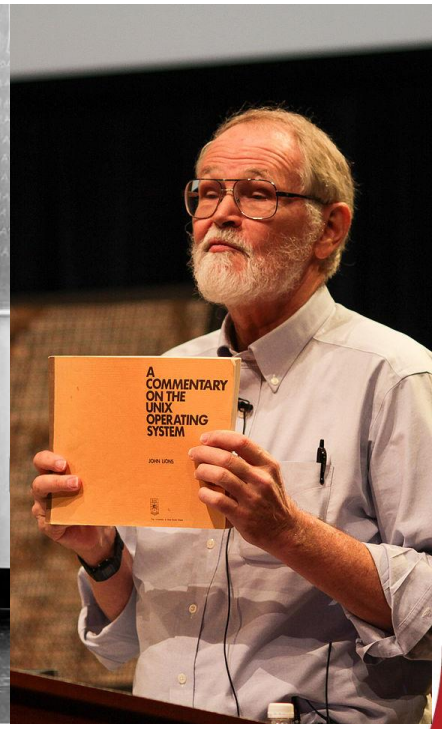


- An **Operating System** is a software layer that sits between hardware and user applications.
  - Manage resources
  - Safe and convenient programming abstractions
  - Services: scheduling, device drivers
- A collection of software services that manage hardware on behalf of your code.

(Short) History of Operating Systems

Lancaster University

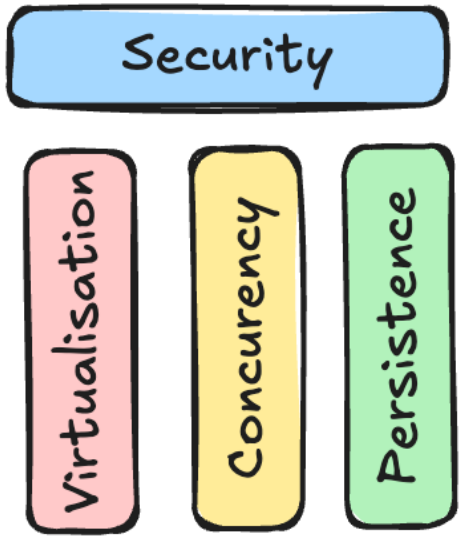
- Started as a convenience library of standard functions
- Evolved from procedure calls to system calls
- OS code executes at a higher privilege level
- Moved from a single process to concurrently executing processes



OS building blocks

Lancaster University

- **Virtualisation**: Each process thinks it has its own resources (e.g., CPU, memory)
- **Concurrency**: hide concurrency from independent processes; provide synchronisation to dependent processes.
- **Persistence**: Allow processes to store non-volatile information
- **Security**: Isolation, Authentication, Protection against exploits.



Example: OS Virtualisation

Lancaster University

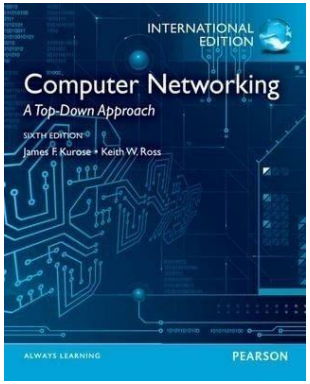
```
import sys
import time


def test(str: str) -> None:
  while True:
    # sleep for a second and
    # print the string
    time.sleep(1)
    print(str)

if __name__ == "__main__":

  # check if there is an argument
  if len(sys.argv) < 2:
    print("Usage: python cpu.py <string>")
    sys.exit(1)
  test(sys.argv[1])
```

```
> python3 cpu.py A & ;
python3 cpu.py B & ;
python3 cpu.py C & ;
python3 cpu.py D &
C
A
D
B
C
A
D
C
B
A
…
```

Lancaster University

Chapter 1.1

What is the Internet?



Overview

Lancaster University

- Internet fundamentals
- Protocols
- Importance of the Internet
- Why should I care?
- Internet Structure

Internet fundamentals

Lancaster University

The Internet comprises three main types of basic **components**:

Hosts:

- Billions of connected hosts, a.k.a., "end-systems"
- Each runs one or more *distributed (networked) applications*

Communication links:

- Fibre, copper, radio, satellite
- Key characteristics of transmission rate and delay: cf. *bandwidth, latency*

Packet switches:

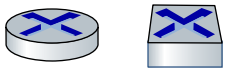
- Used to forward *packets* (chunks of data) arriving on an incoming link to an outgoing link
- *Routers* and link-layer *switches*

Internet infrastructure

An Overview

Billions of connected computing *devices*:

- *hosts = end systems*
- running *network apps* at Internet's "edge"



Packet switches: forward packets (chunks of data)

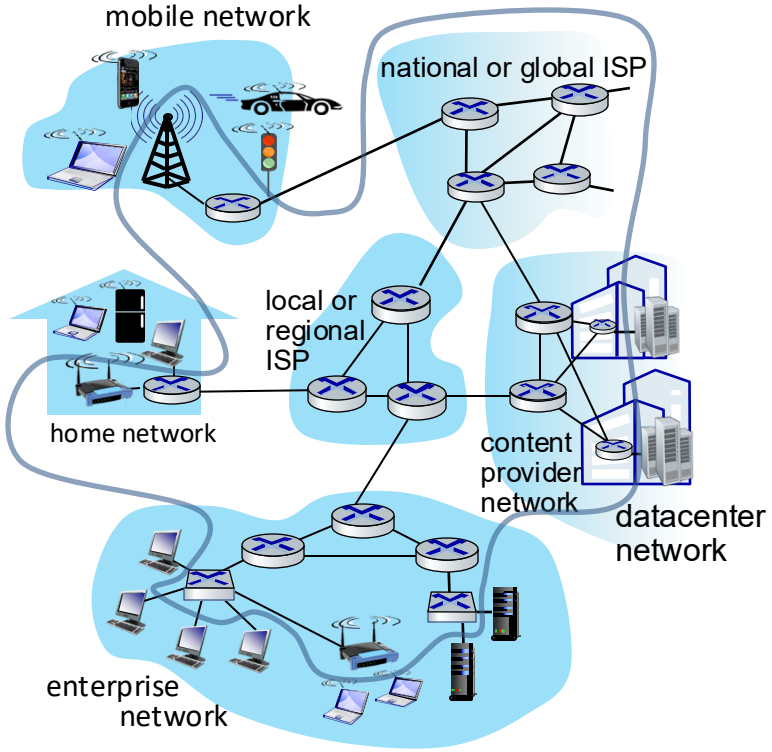
- *routers*, *switches*

Communication links

- fiber, copper, radio, satellite
- transmission rate: *bandwidth*

Networks

- collection of devices, routers, links: managed by an organization



Internet fundamentals

Black box view

- The Internet is an *infrastructure* that provides **services** to *distributed applications,* such as:

  Web, VoIP, e-mail, games, e-commerce, social networks, …
- Distributed Applications:
  - Programs (written in C, Python, Java, etc.) running on different hosts have to communicate (i.e., send data to each other)
- **How does one program running on one end system instruct the Internet to deliver data to another program running on another end system?**
  - There's a *programming interface (Socket API) interfacing with the Operating System* that offers applications access to the Internet infrastructure (more on this later on Monday, Week 4)
  - "Hooks" that allow applications to "connect" to the Internet for sending/receiving data
  - Provides various service options

Internet infrastructure

A more detailed view

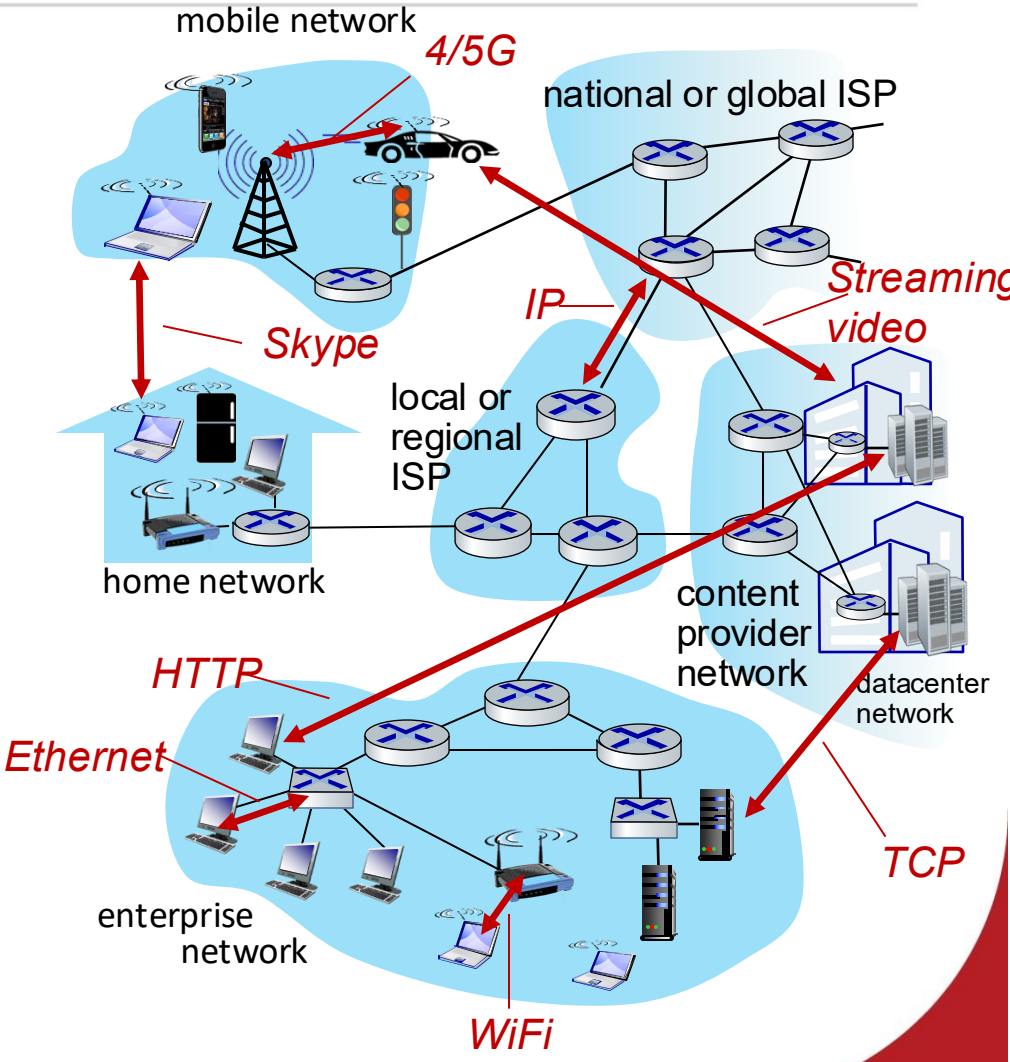
Internet: "network of networks"

- Interconnected ISPs (and other neworks)

Protocols control sending, receiving of messages, e.g., HTTP (Web), streaming video, Skype, TCP, IP, WiFi, 4/5G, Ethernet

Internet standards

- Deliver standardised protocols that the world can use
- IETF: Internet Engineering Task Force
  - RFC: "Request for comments"documents



What's a Protocol?

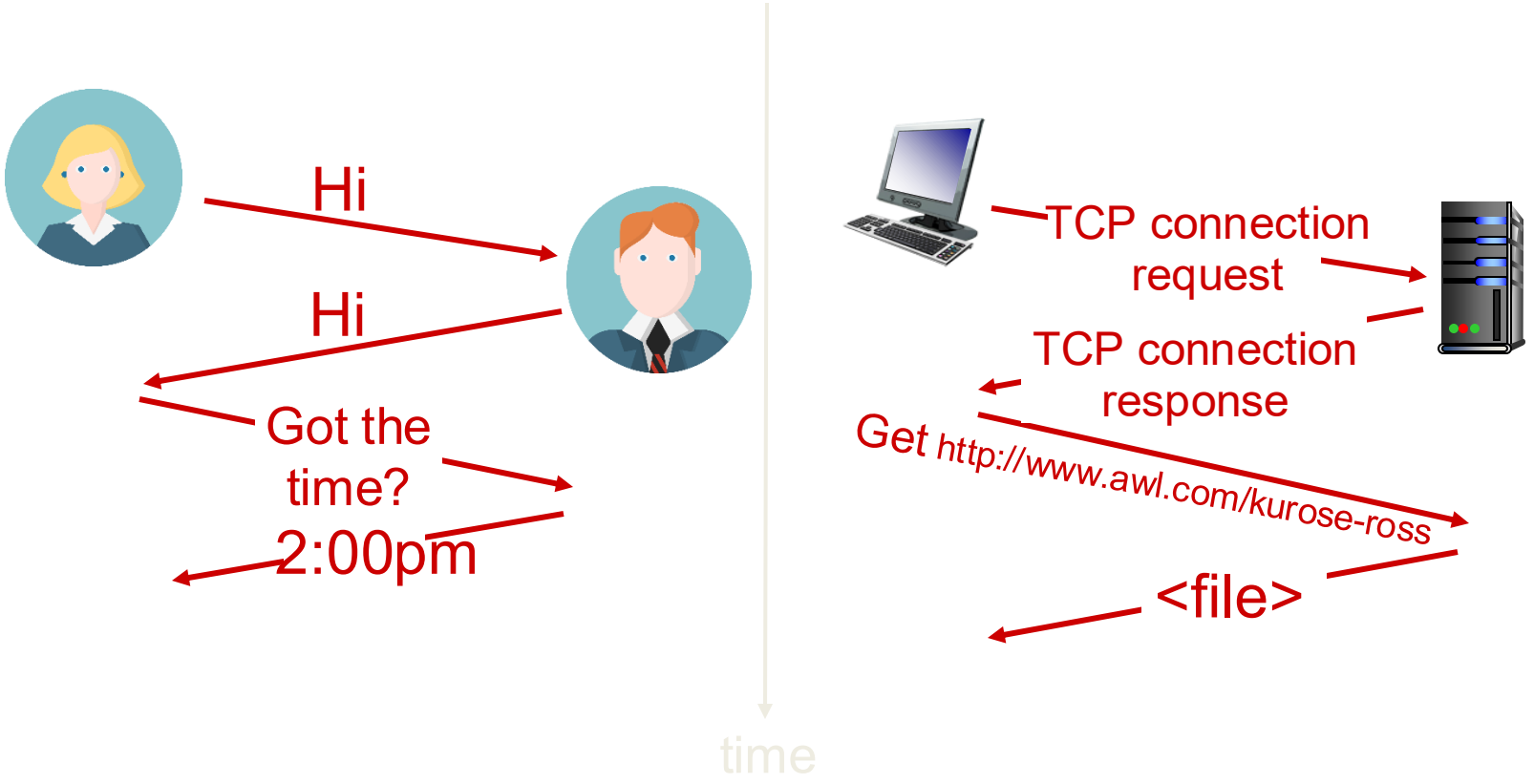
Definition

A Protocol defines the format, order of messages exchanged among network entities (e.g., hosts, routers), and actions to be taken on message transmission/receipt/non-receipt

What's a Protocol?

Comparison

A human protocol and a computer network protocol:



What's a Protocol?

Human vs. Network protocols

Human protocols:

- "what's the time?"
- "I have a question"
- introductions

Rules for:

… specific messages sent

… specific actions taken when message received, or other events

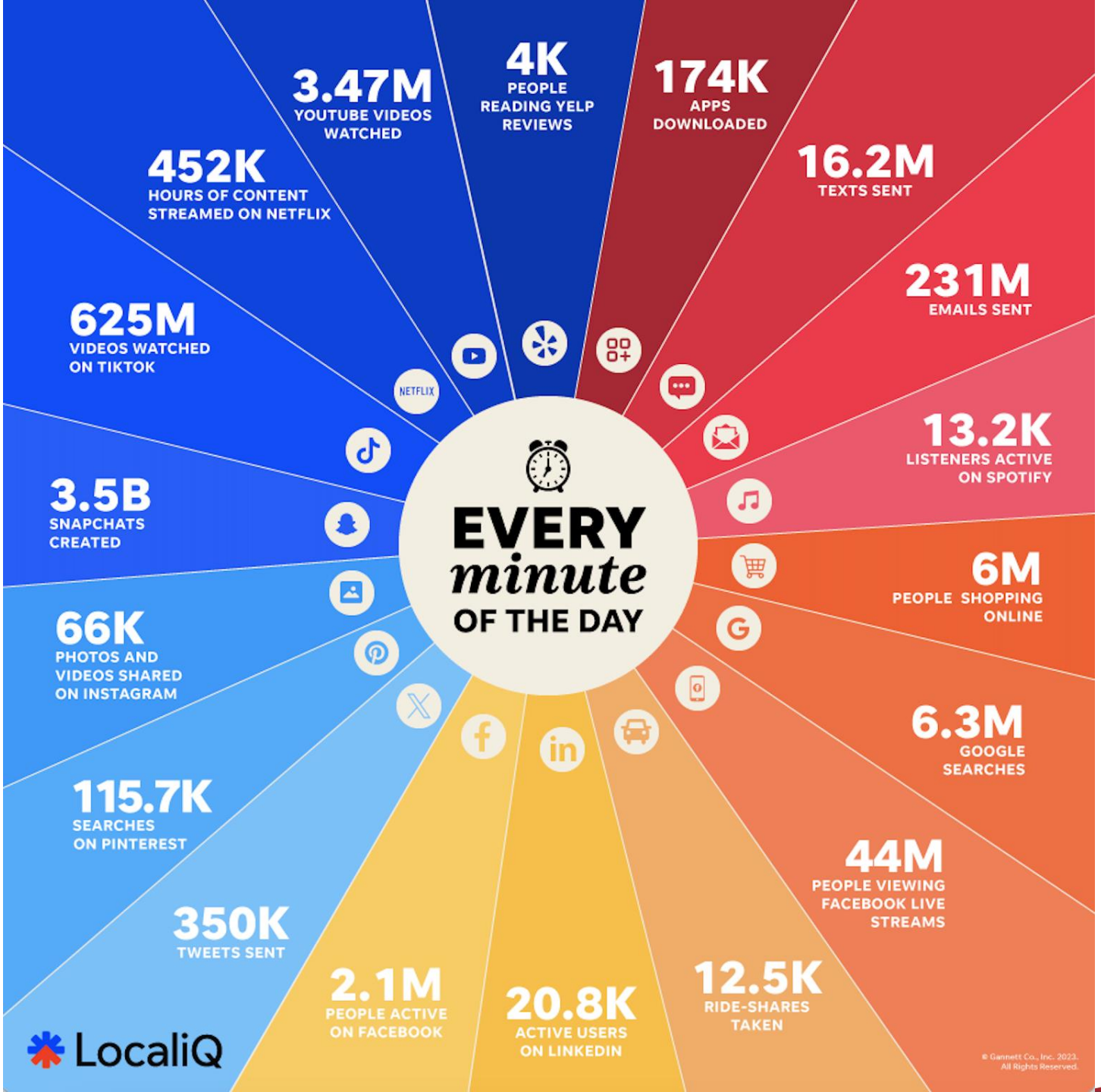
Network protocols:

- computers (devices) rather than humans
- all communication activity in Internet governed by protocols

Protocols define the format, order of messages sent and received among network entities, and actions taken on message transmission, receipt

Lancaster University

Why is the Internet so important?


EVERY minute OF THE DAY
3.47M YOUTUBE VIDEOS WATCHED
4K PEOPLE READING YELP REVIEWS
174K APPS DOWNLOADED
16.2M TEXTS SENT
231M EMAILS SENT
13.2K LISTENERS ACTIVE ON SPOTIFY
6M PEOPLE SHOPPING ONLINE
6.3M GOOGLE SEARCHES
44M PEOPLE VIEWING FACEBOOK LIVE STREAMS
12.5K RIDE-SHARES TAKEN
20.8K ACTIVE USERS ON LINKEDIN
2.1M PEOPLE ACTIVE ON FACEBOOK
350K TWEETS SENT
115.7K SEARCHES ON PINTEREST
66K PHOTOS AND VIDEOS SHARED ON INSTAGRAM
3.5B SNAPCHATS CREATED
625M VIDEOS WATCHED ON TIKTOK
452K HOURS OF CONTENT STREAMED ON NETFLIX
NETFLIX
LocaliQ
© Gannett Co., Inc. 2023. All Rights Reserved.

What if we didn't have the Internet?

Lancaster University

It is the unseen infrastructure behind much of our daily lives

Imagine if it went away tomorrow - what would you miss?

- Education
- Banking
- Commerce
- Inter-personal and social communication
- Entertainment
- Transport control
- Home control
- Political discourse
- ....??

Lancaster University

Why should we care about networking and the Internet?

What does it mean for your future career?

Lancaster University

Thinking about your job prospects in a few years…

- You *may or may not* become a network or systems engineer
- You *may well*, however, become a software developer

It is easy to ignore the network, and treat it as a given commodity that is always there and always perfect

But if you don't understand the underlying reality, it can easily:

- Break the best-engineered application
- Ruin the smoothest user interaction

What this course will and won't do…

Lancaster University

With this course, we are **not** suggesting that you should be able to go and fix the network when it malfunctions

Although that would be nice 

However, if you can at least understand *why* and *how* it might be having an effect on your software, then we have succeeded!

Thanks for listening!
Any questions?

Lancaster University

Reading Material:

- "Computer Networking: A Top-Down Approach" Sections 1.1 and 1.2
- "Operating Systems: Three Easy Pieces", Sections 2.1, 2.2,2.3, and 2.4