Operating Systems
Spring 2020

National University of Computer and Emerging Sciences

Acknowledgment

- Acknowledgements to
  - Marina Papatriantafilou (Chalmers)
  - Maarten van Steen (Vrije Amsterdam)
  - Boris Koldehofe (TU Darmstadt)

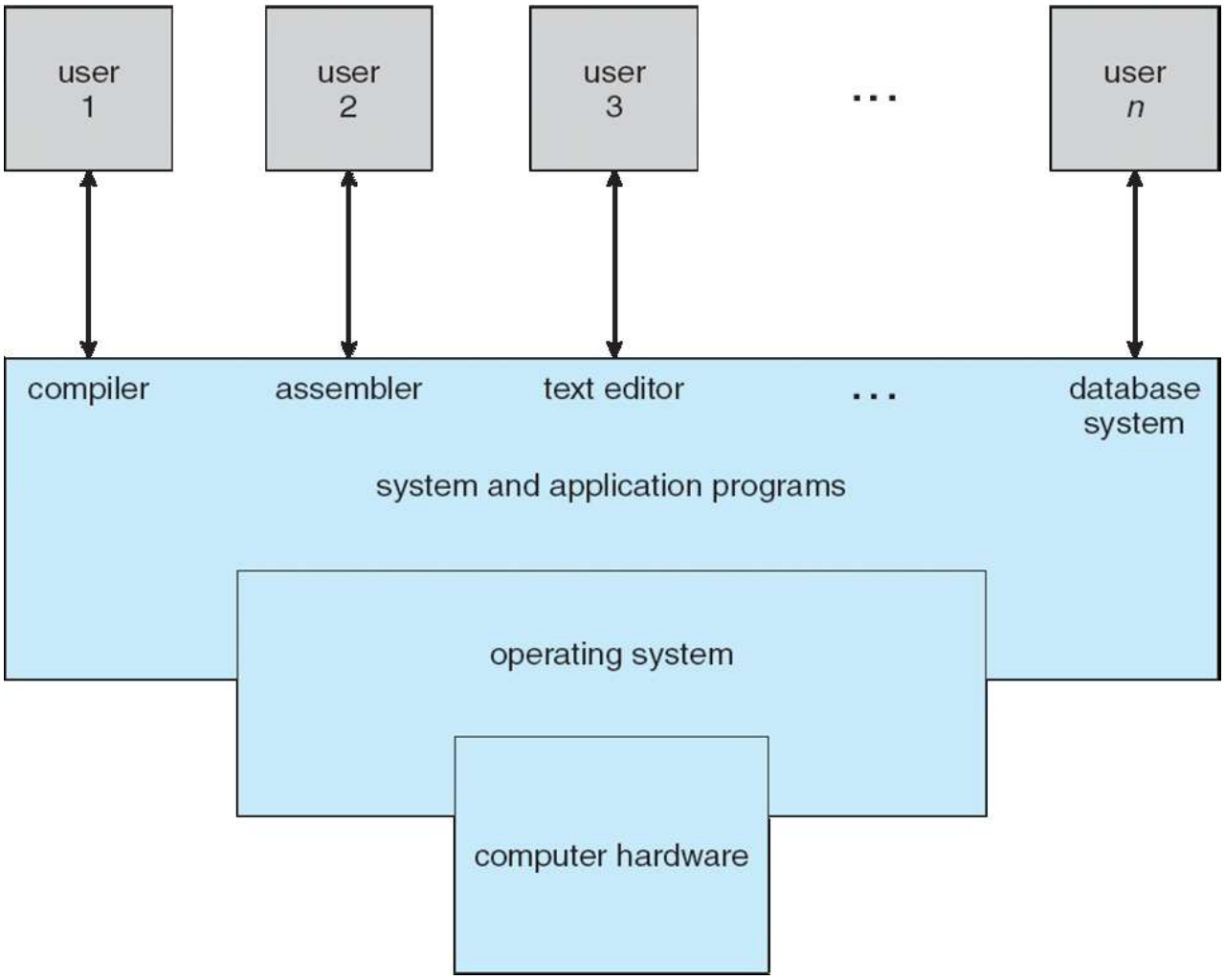
- A lot of the course material is based on their lecture notes

- Many Slides and Figures are also based on instructor material of various books
  - William Stallings (Operating systems)
  - Abraham Silberschatz (Operating System Concepts)
  - Andrew Tanenbaum (Modern Operating System Concepts)

Outline

- Basic Information
  - Motivation
  - Brief history of operating systems
  - Course overview
  - Course information

Components of a Computer System



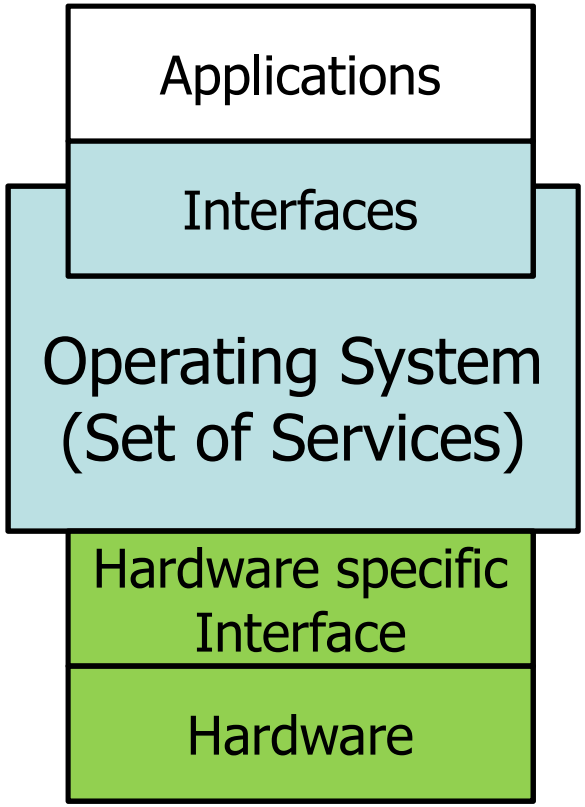
Components of a Computer System

Computer system can be divided into four components

- Hardware
  - Provides basic computing resources
  - CPU, memory, I/O devices
- Operating system
  - Controls & coordinates use of hardware among various applications and users
- Application programs
  - Define the ways in which the system resources are used to solve the computing problems of the users
  - For example, word processors, compilers, web browsers, database systems, video games
- Users
  - People, machines, other computers

Operating System

- Provides a set of services to system users
  - Intermediary between user and hardware
- Shield between the user and the hardware
- Resource manager
  - CPU(s), memory, I/O devices
- Objectives
  - Convenience (to use)
  - Efficiency (utilization of hardware)
  - Ability to evolve



Operating System Viewpoints

Use view

- Users want convenience, ease of use
- Don’t care about resource utilization

System view

- Resource allocator
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use
- Control program
  - Controls execution of programs to prevent errors and improper use of the computer

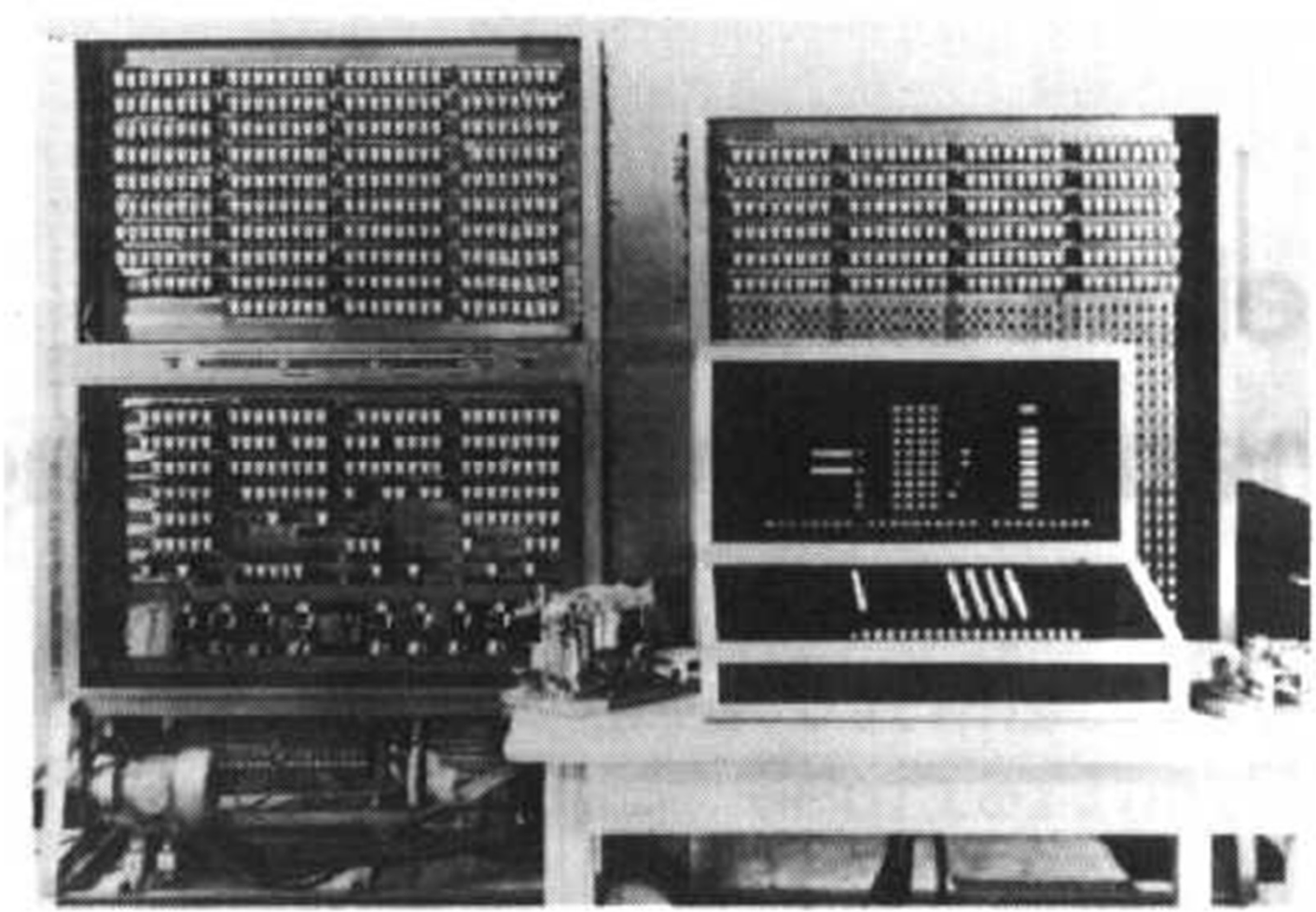
Services Provided by the Operating System

- Program execution
  - CPU scheduling, resource (memory) allocation and management, synchronization
- Access to I/O devices
  - Uniform interfaces, hide details, optimize resources (disk scheduling)
- Controlled access to files
  - Structure of data
- System/resource access
  - Allocation, authorization, protection,
- Utilities, e.g., for program development
  - Compilers, debuggers
- Error detection and response, when, e.g.,
  - Hardware, software errors
  - Operating system cannot grant request of application
- Monitoring, accounting

Some History: Evolution of Operating Systems

- Hardware upgrades, new types of hardware, enabled features
- New services, new needs
- Development
  - Serial processing (1948 – 1955)
  - Batch systems (1955 for IBM)
  - Multiprogramming
  - Time shared systems (1966: Compatible Time Sharing System (CTSS) by MIT)
- Before 1948?

Konrad Zuse (1910-1995) / Z3 (1941)



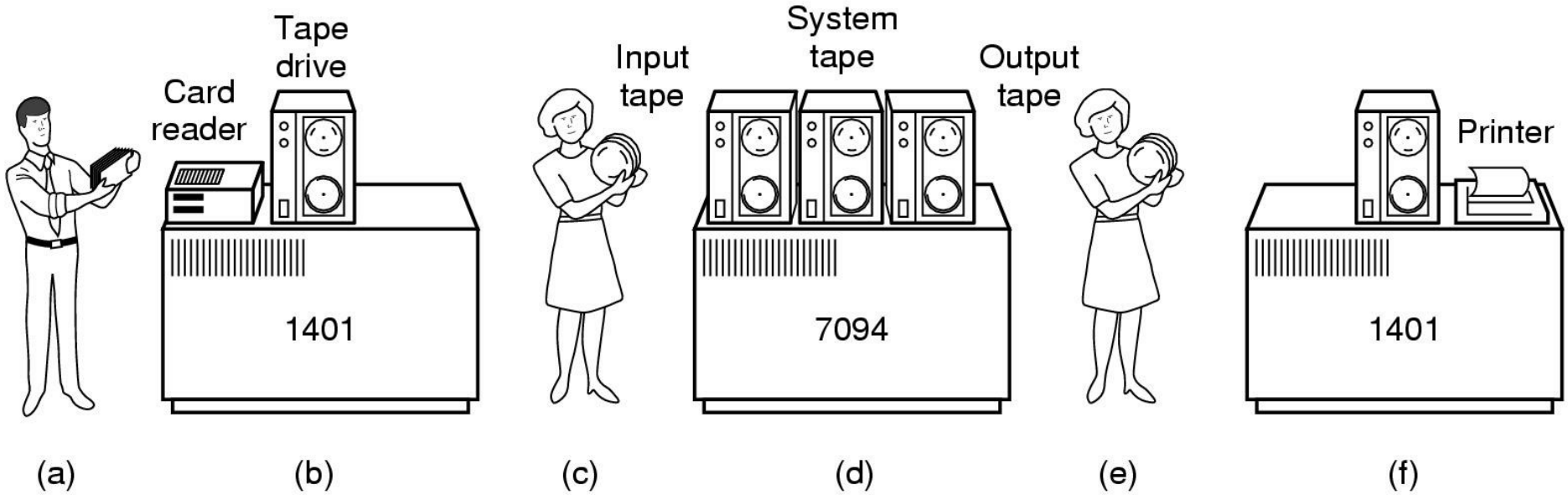
OS History: Serial Processing (Before the Stone Age)

- No operating system

- System model
  - Simple console: display lights and toggle switches
  - Input device (card reader)
  - Execution monitoring
    - Output on a printer (job went well)
    - Control lights for error detection (check status of registers)

- Job scheduling
  - Manual (reservations on a piece of paper)
    - Overestimation of execution time
    - Low utilization of CPU

OS History: Early Batch System

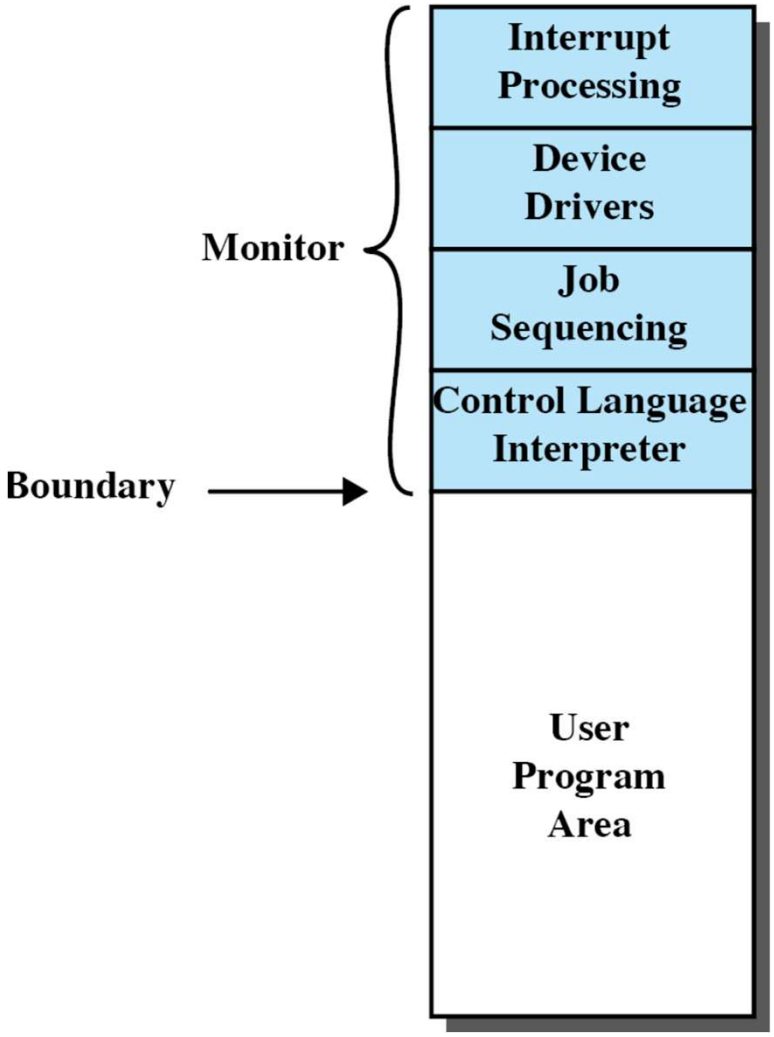


a) Programmer bring cards to 1401
b) 1401 reads batch of jobs onto tape
c) Operator caries input tape to 7094
d) 7094 does computing
e) Operator carries output tape to 1401
f) 1401 prints output

OS History: First "Tools" Appear

Simple Batch Systems

- Goal
  - Improve machine utilization
- Monitors
  - Software that controls the running programs
  - Batch jobs together
  - Program returns control to monitor when finished
  - Resident monitor is in main memory and available for execution
- Job Control Language (JCL)
  - Provides instruction to the monitor
  - What compiler to use, (Fortran, …)
  - What data to use



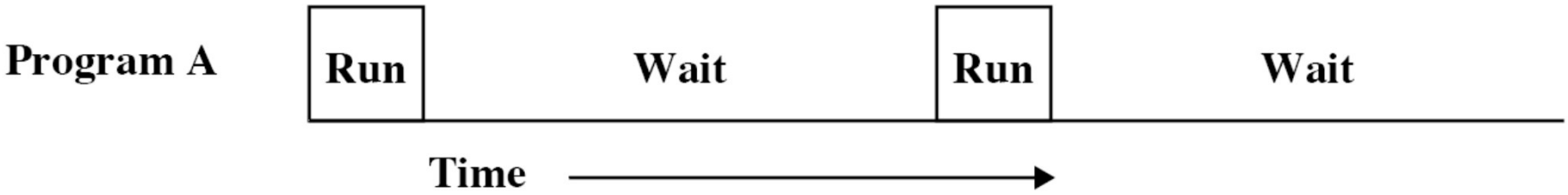
OS History: Hardware Features…

… which made the first tools possible

- Memory protection
  - Do not allow the memory area containing the monitor to be altered
  - On attempt processor hardware should detect an error
  - Concepts of mode operation
    - User mode: certain areas of memory are protected / some instructions not allowed
    - Kernel mode: allows access to protected memory / execution of special instructions
- Privileged instructions
  - Only for monitor, e.g., for interface with I/O devices
- Interrupts
  - Mechanisms for the OS to relinquish control and regain it
- Timer
  - After timeout job return to the monitor
  - Prevents a job from monopolizing the system

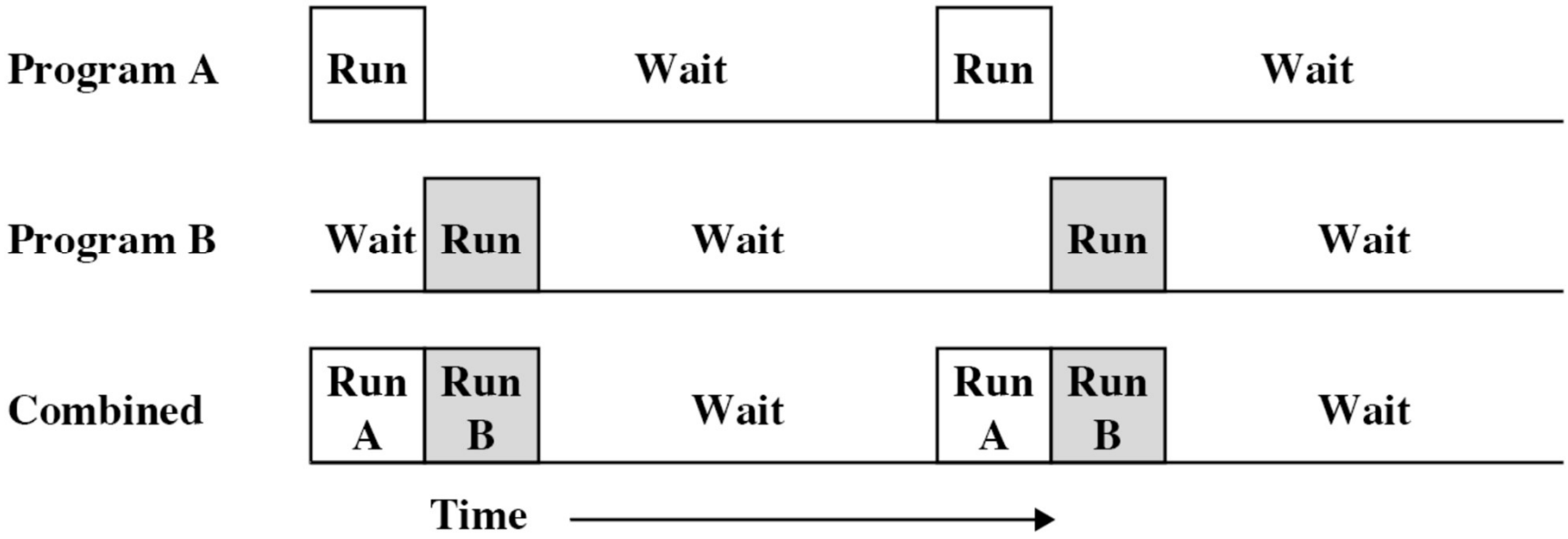
From Uniprogramming …

- Processor must wait for I/O instruction to complete before proceeding
- One job prevents others from making progress



… To Multiprogramming

- Increases processor utilization
- When one job needs to wait for I/O, the processor can switch to the other job



Obvious Disadvantages

- Certainly not suitable for modern applications
  - Word processing
- Even if computation only matters
  - Long response for a developer until feedback is received
  - Time until job is dispatched typically long
- Need for interactivity with a computer system

OS History: Multiprogramming, Time Sharing

- Time sharing systems use multiprogramming to handle multiple interactive jobs
- Processor's time is shared among multiple users
- Multiple users simultaneously access the system through terminals

	Batch Multiprogramming	**Time Sharing**
Principal objective	Maximize processor use	Minimize response time
Source of directives to operating system	Job control language commands provided with the job	Commands entered at the terminal

What to Learn in This Course?

Understand concepts of modern operating systems

- **Design principles**
  - Data structures, kernel organization, ...
- **Fundamental concepts**
  - Processes, Threads, Virtual Memory, File and I/O, ...
- **Basic algorithms**
  - Scheduling, Paging, Resource Allocation, ...
- **Characteristics/performance of a system**
  - Thrashing, Throughput, Real-time, ...
- **Concurrent programming concepts**
  - Synchronization, Mutual exclusion, Deadlock avoidance, ...

- **BUT:** No focus on a specific operating system

Course Outline

- Introduction
- Processes management
  - Multithreaded programming
  - Process scheduling
- Process Coordination
  - Synchronization
  - Deadlock
- Memory Management
  - Virtual memory
- Storage Management
  - File system
  - I/O system

Pre-Requisites

- Must have passed data structures (CS-201)
- Familiar with programming C/C++

Tentative Evaluation Breakdown

Evaluation Name	Weightage
Assignment*	10
Quiz	10 (Best 5)
Lab	10 (5 lab task + 5 exam)
Project	10
Mid Term	20
Final	40

- Same grading for both theory and lab
- The evaluation breakdown and course outline for all sections will be same
- Grading will be combined so don’t rely on your class position look for the batch position

Some Rules

- Class participation is important and encouraged
  - Wait for others to complete
- Never ever miss a class
  - You might miss quiz
- Never use mobile phone in the class
- Above all, whatever you do, please do not disturb others

Retake Policy

- No retake for assignments and quizzes
- No retake of mid term exam
  - Marks will be awarded based on the performance in the final exam
- Retake of project demos will be evaluated on individual basis
- Retake of final exam will be decided by the university policy

Dishonesty and Plagiarism

- Any kind of cheating will be considered serious offense
- All parties involve in cheating will get zero marks
  - Quizzes, assignments and project
- Habitual cases will be referred to DC

Attendance Policy

- A student is either "present" or "absent"
  - Late arrivals will be treated as absent
  - Student arriving 10 minutes late will be marked absent

- Students are **not** allowed to switch sections for any reason

Course Material (1)

Course book

- Avi Silberschatz, Peter Baer, and Galvin Greg Gagne: Operating System Concepts, 8/E, John Wiley & Sons.

Research books

- William Stallings**:** Operating Systems: Internals & Design Principles, 6/E, Prentice Hall.
- Andrew S. Tanenbaum: Modern Operating Systems, 3/E, Prentice Hall.
- Andrew S Tanenbaum and Albert S Woodhull: OS Design and Implementation, 3/E, Prentice Hall.

Course Material (2)

- Lecture slides and announcements for all sections
  - Google Classroom: Operating Systems Spring 2020
  - Class Code: 3wtc47r

Any Question So Far?