Major Achievements

- Four major advances in the development of operating systems
  - Processes
  - Memory management
  - Information protection and security
  - Scheduling and resource management
The Process

- Three major developments related to the concept of process
  - Multiprogramming: keep processors and I/O devices busy to improve resource utilization
  - Time sharing: be responsive while supporting many users simultaneously
  - Real-time transaction: exclusive access to resources with commit or abort operations
The Process

- Program coordination is difficult. Some sources of errors are:
  - Improper *synchronization*: signals from an I/O completion can be lost or duplicated
  - Failed *mutual exclusion*: shared resources are accessed by more than one programs
  - *Nondeterministic* program execution: depending on the memory footprint from other programs, the execution of a program differs
  - Deadlocks: two or more programs are waiting for the resources held by others indefinitely
The Process

- To tackle such errors
  - Needs a systematic way of monitor and control various programs running on a processor
  - The concept of process provides a foundation

- Components of a process
  - An executable program
  - Associated data needed by the program
    - variables, work space, buffers, ...
  - The execution context of the program
    - Registers, priority, is waiting for an I/O completion, ...
The Process

A typical Process Implementation
Memory Management

- Five principal storage management responsibilities
  - Process isolation:
    - Prevent processes from interfering with other’s memory
  - Automatic allocation and management:
    - Memory hierarchy should be used dynamically
    - Allocation should be transparent to the programmer
  - Support of modular programming:
    - Modules can be dynamically created, destroyed, ...
Memory Management

- Five principal storage management responsibilities (continued)
  - Protection and access control:
    - Sharing of memory enables one process to access the address space of another
    - OS must allow portions of memory to be accessible in various ways by various users
  - Long-term storage:
    - Storing information for extended period of time after the computer has been powered down
Memory Management

- Virtual memory
  - Allows programs to address memory from a logical point of view without considering the physical amount of main memory
- Paging
  - Reduces fragmentation in virtual memory
- Process isolation
  - Give each process a unique, non-overlapping memory
- Memory sharing
  - Overlapping portions of two virtual memory space
If valid bit is set, MMU uses the physical address in PTE (page table element) to construct the physical address of the word.
Memory Sharing by Virtual Memory
Memory Protection by Virtual Memory

- Control the access to the contents of a virtual page by adding some additional permission bits
  - SUP: can be accessed in kernel mode
  - READ, WRITE: read/write control
Linux Virtual Memory System
Information Protection and Security

- Controlling the access to computer systems and the information stored in them

- Four categories:
  - Availability: protecting the system against interruption
  - Confidentiality: prevent unauthorized reading
  - Data integrity: prevent unauthorized writing
  - Authenticity: verification of the identity of the user and the validity of message or data
Scheduling and Resource Management

- Three factors to consider in resource allocation
  - Fairness: give equal and fair access to resources to all processes in the same class
  - Differential responsiveness: discriminate among different classes of jobs
  - Efficiency: maximize throughput, minimize response time, and accommodate as many users as possible
Developments Leading to Modern Operating Systems

- Microkernel architecture
  - Monolithic kernel:
    - A large kernel provides most of the OS functionality
    - A single process with all elements sharing the same address space
  - Microkernel architecture: Kernel has only few essential functionalities
    - Address space management, interprocess communication, basic scheduling
    - Other OS services are provided by user mode processes.

- Multithreading
  - Thread: a logical flow that runs in the context of a process
  - Process: A collection of one or more thread and associated system resources
Developments Leading to Modern Operating Systems

- Symmetric multiprocessing
  - OS schedules processes and threads across all processors
  - Benefits: performance, availability, incremental growth, and scaling

- Distributed operating systems
  - Provides an illusion that a cluster of machines is running as a single computer

- Object-oriented design
  - Adding modular extension to a small kernel
Fault Tolerance

- **Fault tolerance**
  - The ability of a system to continue normal operation despite the presence of hardware or software error

- **Reliability**
  - $R(t)$: the probability that a system operates correctly up to time $t$

- **Mean time to failure (MTTF)**
  - $\text{MTTF} = \int_0^\infty R(t) \, dt$

- **Mean time to repair (MTTR)**
  - The average time it takes to repair a faulty element
Fault Tolerance

- **Availability**
  - Fraction of time the system is available to serve users’ requests

\[
A = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}}
\]

\[
\text{MTTF} = \frac{B_1 + B_2 + B_3}{3}
\]

\[
\text{MTTR} = \frac{A_1 + A_2 + A_3}{3}
\]
Fault Tolerance

- **Solutions: adding redundancy**
  - **Spatial (physical) redundancy:** use multiple components performing the same function or backup
    - Backup name server on the Internet
  - **Temporal redundancy:** repeating a function or operation when an error is detected
    - Data retransmission
  - **Information redundancy:** replicating or coding data such that an error can be detected and corrected
    - RAID disks
Fault Tolerance

- **Operating System Mechanisms**
  - Process isolation: main memory, file access, flow of execution
  - Concurrency controls: recover from fault conditions like deadlock
  - Virtual machines: application isolation and redundancy
  - Checkpoints and rollback:
    - Checkpoint: a copy of application’s state
    - Rollback: restart the execution from a checkpoint