COMPUTER SYSTEMS & PROGRAMMING

Operating System Structures

Zuhaib A. Shaikh,
Asst. Prof., CSE Deptt.,QUEST

Web: zuhaib-shaikh.neocities.org

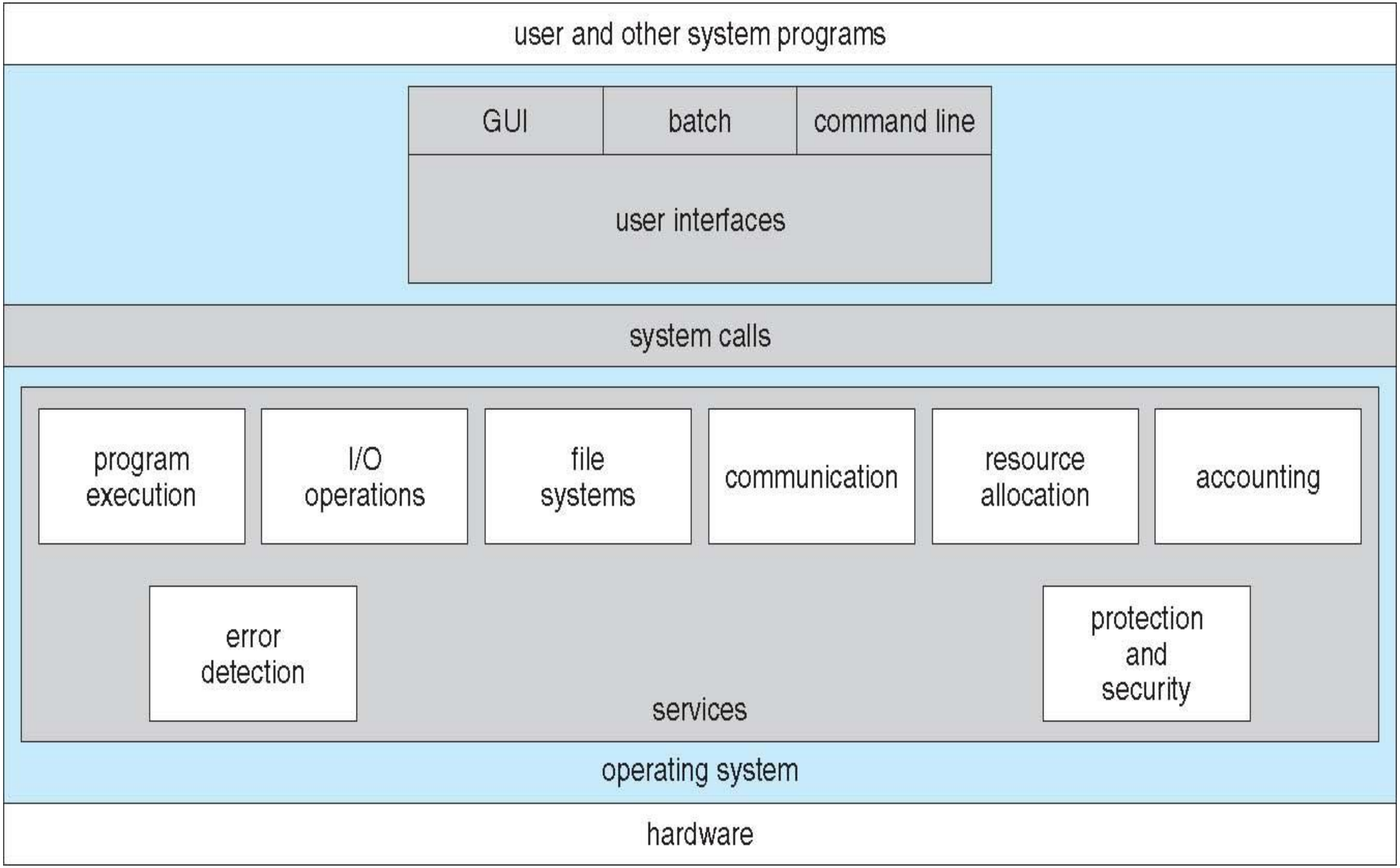
Operating System Structures

- Operating System Objectives
- Operating System Services
- Common System Components
- System Calls and APIs
- System Programs
- OS design & evolution

OS Objectives

- Convenience: To provide a way to use system easily
- Efficiency: OS should manage all resources effectively
- Ability to evolve: OS should be able to evolve without disturbing whole system

A View of Operating System Services



Common System Components

- Process Management
- Main Memory Management
- File System Management
- I/O System Management
- Mass-Storage Management
- Networking
- Command Interpreter
- Protection and Security

Process Management

- A process is a program in execution. It is a unit of work within the system.
- Program is a passive entity, process is an active entity.
- The operating system is responsible for the following activities in connection with process management:
  - Creating and deleting both user and system processes.
  - Suspending and resuming processes.
  - Providing mechanisms for process synchronization.
  - Providing mechanisms for process communication.
  - Providing mechanisms for deadlock handling.

Process Management

- A Process needs resources to accomplish its task:
  - CPU, memory, I/O devices, files
  - Initialization data
- Process termination requires reclaim of any reusable resources.
- Typically a system has many processes, some user, some operating system running concurrently on one or more CPUs
  - Concurrency by multiplexing the CPUs among the processes/threads.

Main Memory Management

- All data in memory before and after processing.
- All instructions in memory in order to execute.
- Memory management determines what is in memory and how to handle it
  - Optimizing CPU utilization and computer response to users.
- Memory management activities:
  - Keeping track of which parts of memory are currently being used and by whom.
  - Deciding which processes (or parts thereof) and data to move into and out of memory.
  - Allocating and de-allocating memory space as needed.

File System Management

- A file is a collection of related information defined by its creator.
- Commonly, files represent programs and data.
- Files are usually organized into directories.
- OS activities include:
    - Creating and deleting files and directories.
    - Primitives to manipulate files and directories.
    - Mapping files onto secondary storage.
    - Backup files onto stable (non-volatile) storage media.

I/O System Management

- One purpose of OS is to hide peculiarities of hardware devices from the user.
- I/O subsystem is responsible for:
  - Memory management of I/O including:
    - buffering (storing data temporarily while it is being transferred)
    - caching (storing parts of data in faster storage for performance)
    - spooling (CPU track I/O devices for communication)
    - interrupts (I/O communication during execution of other process)
    - DMA (Direct Memory Access, to transfer data from one memory to another).
  - General device-driver interface.
  - Drivers for specific hardware devices.

Mass-Storage Management

- Usually disks used to store data that does not fit in main memory or data that must be kept permanently.
- Entire speed of computer operation depends on disk subsystem and its algorithms.
- OS activities:
  - Free-space management
  - Storage allocation
  - Disk scheduling
- Some storage need not be fast:
  - Storage includes optical storage, magnetic storage etc.
  - Varies between WORM (write-once, read-many-times) and RW (read-write)

Networking

- The processors in the system are connected through a communication network.
- Communication takes place using a protocol.
- A networked/distributed system provides user access to various system resources.
- Access to a shared resource allows:
  - Computation speed-up
  - Increased data availability
  - Enhanced reliability

Command Interpreter

- Command-Interpreter:
  - The program that reads and executes commands given to the operating system.
  - Examples of command-line interpreters are command (DOS), shell (UNIX).
  - In windows and in some other operating systems have GUI interface– WIMP (Windows, Icons, Menu, and Pointing device).

Protection and Security

- Protection – any mechanism for controlling access of processes or users to both system and user resources.
- Security – defense of the system against internal and external attacks, with huge range, including:
    - Denial-of-service
    - Worms
    - Viruses
    - Identity theft
    - Theft of service

Error Detection/Response

- Error Detection
  - internal and external hardware errors
    - memory error
    - device failure
- Software errors
    - arithmetic overflow
    - access forbidden memory locations
- Error Response
  - simply report error to the application
  - Retry the operation
  - Abort the application

Accounting

- Accounting keeps track of and records which users use how much and what kinds of computer resources.
- Accounting:
  - collect statistics on resource usage
  - monitor performance (e.g., response time)
  - used for system parameter tuning to improve performance
  - useful for anticipating future enhancements
  - used for billing users

System Calls and APIs

- Systems calls provide programming interface to the services provided by the OS.
- Typically written in a high-level language (C/C++).
- Mostly accessed by programs via a high-level Application Program Interface (API) rather than through direct system calls.
- Three most common APIs:
  1. Win32 API for Windows
  2. POSIX API for POSIX-based systems (including virtually all versions of UNIX, Linux, and Mac OS X)
  3. Java API for the Java Virtual Machine (JVM)

System Programs

- System programs provide a convenient environment for program development and execution.
- Most users’ view of the operating system is defined by system programs, not the actual system calls.
- E.g.
  - File management & modification
  - Status information
  - Programming language support
  - Program loading and execution
  - Communications

Utility Programs

- Perform tasks related to the control and allocation of computer resources
- E.g.
  - Data-recovery utility
  - Virus Protection
  - Data Compression
  - Disk Scanner & Disk Cleanup
  - File Defragmentation
  - Disk Management

OS design & evolution

- Operating system should be modular in design
- Doing so, one component can change without interfering other components/system
- Ease of evolution of OS:
  - New fixes
  - Hardware change/upgrades
  - New services

Questions