Introduction

- What OS do
- Computer-System Organization/Architecture
- OS Structure/Operations
- Process Management
- Memory Management
- Storage Management
- Protection and Security
- Distributed and Special-Purpose Systems
- Computing Environments
- Open-Source Operating Systems
What is an OS?

- A program
- Intermediary between users and hardware

Operating system goals:
- Execute user
- Make the computer system convenient to use
- Use resources efficiently

Computer System Structure

- Computer system components:
  - Hardware – basic computing resources
    - CPU, memory, I/O devices
  - Operating system
    - Controls and coordinates use of hardware
  - Application programs – uses system resources to solve computing problems
    - Word processors, compilers, web browsers, database systems, video games
  - Users
    - People, machines, other computers
Four Components of a Computer System

What Operating Systems Do

- Users want convenience, ease of use
- Don’t care about resource utilization
- Shared computers must keep all users happy
- Users of workstations have dedicated resources … frequently use shared resources from servers
- Handheld computers are resource poor (relative), optimized for usability and battery life
- Embedded computers have little or no user interface
### Operating System Definition

- **OS is a resource allocator**
  - Manages all resources
  - Decides between conflicting requests - efficient and fair resource use

- **OS is a control program**
  - Controls execution of programs to prevent errors and improper use of the computer

### Operating System Definition

- No universally accepted definition

- “Everything a vendor ships when you order an operating system” is good approximation
  - But varies wildly

- “The one program running at all times on the computer” is the **kernel**. Everything else is either a system program (ships with the operating system) or an application program.
Computer Startup

- **bootstrap program** is loaded at power-up or reboot
  - Typically stored in ROM or EPROM, generally known as **firmware**
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution

![PDP 8](image-url)
Comp. Syst. Organization

- Computer-system operation
  - One or more CPUs & device controllers connect through common bus providing access to shared memory
  - Concurrent execution of CPUs and devices competing for memory cycles

Computer-System Operation

- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
- Each device controller has a local buffer
- CPU moves data from/to main memory to/from local buffers
- I/O is from the device to local buffer of controller
- Device controller informs CPU that it has finished its operation by causing an interrupt
Interrupts

- Interrupt transfers control to the interrupt service routine, through the interrupt vector (addresses of all service routines)
- Interrupt architecture must save the address of the interrupted instruction
- Incoming interrupts are disabled while another interrupt is being processed to prevent a lost interrupt
- A trap is a software-generated interrupt caused either by an error or a user request
- An operating system is interrupt driven

Interrupt Handling

- OS preserves the state of the CPU by storing registers and the program counter
- Determines which type of interrupt has occurred:
  - polling
  - vectored interrupt system
- Separate segments of code determine what action should be taken for each type of interrupt
Interrupt Timeline

- After I/O starts, control returns to user program only upon I/O completion
  - Wait instruction idles the CPU until the next interrupt
  - Wait loop (contention for memory access)
  - At most one I/O request is outstanding at a time, no simultaneous I/O processing

- After I/O starts, control returns to user program without waiting for I/O completion
  - System call – request to the operating system to allow user to wait for I/O completion
DMA Structure

- Device controller transfers blocks of data from buffer storage directly to main memory without CPU intervention
- One interrupt per block, not one per byte
- High-speed I/O devices transmit data at high speeds

Storage Structure

- Main memory – only large storage media that the CPU can access directly
  - Random access
  - Volatile
- Secondary storage – large nonvolatile
- Magnetic disks – rigid platters covered with magnetic recording material
  - Disk surface is logically divided into tracks, which are subdivided into sectors
  - Disk controller – logical interaction between device and computer
Storage-Device Hierarchy

Caching

- Information in use copied from slower to faster storage temporarily
- Faster storage (cache) checked first to determine if information is there
  - ✓, used directly from cache (fast)
  - ✗, data copied to cache and used there
- Cache smaller than storage being cached
  - Cache size and replacement policy
- Performed at many levels
Modern Computers

- Most systems use a single general-purpose processor (PDAs through mainframes)
  - special-purpose processors as well

- Multiprocessors systems growing in use and importance
  - Also known as parallel systems, tightly-coupled systems
  - Advantages include:
    1. Increased throughput
    2. Economy of scale
    3. Increased reliability – graceful degradation or fault tolerance

- Two types:
  1. Asymmetric Multiprocessing
  2. Symmetric Multiprocessing

Comp. Syst. Architecture

- Most systems use a single general-purpose processor (PDAs through mainframes)
  - special-purpose processors as well

- Multiprocessors systems growing in use and importance
  - Also known as parallel systems, tightly-coupled systems
  - Advantages include:
    1. Increased throughput
    2. Economy of scale
    3. Increased reliability – graceful degradation or fault tolerance

- Two types:
  1. Asymmetric Multiprocessing
  2. Symmetric Multiprocessing
Symmetric Multiprocessing Architecture

A Dual-Core Design
Clustered Systems

- Like multiprocessor systems, but multiple systems working together
  - Share storage via a storage-area network (SAN)
  - Provides a high-availability service which survives failures
    - Asymmetric clustering has one machine in hot-standby mode
    - Symmetric clustering has multiple nodes running applications, monitoring each other
- High-performance computing (HPC)
  - Applications must be written to use parallelization
Operating System Structure

- **Multiprogramming** for efficiency
  - One user cannot keep CPU and I/O devices busy at all times
  - Multiprogramming - CPU always has one job to execute
  - Subset of jobs in memory
  - Jobs selected and run - **job scheduling**
  - Job waits (for I/O for example), OS switches to another job

- **Timesharing (multitasking)**
  - CPU switches jobs so fast - users don’t notice ⇒ **Interactive** computing
  - Each user has at least one program executing in memory ⇒ **process**
  - > 1 Jobs ready to run ⇒ **CPU scheduling**
  - If processes don’t fit in memory ⇒ **Swapping**
  - **Virtual memory** ⇒ execution of processes not completely in memory

---

Memory Layout for Multiprogrammed System

```
0

operating system

job 1

job 2

job 3

job 4

512M
```
OS Operations

- Interrupt driven by hardware
- Software error or request creates **exception** or **trap**
  - Division by zero, request for operating system service
- Other process problems include infinite loop, processes modifying each other or the operating system

OS Operations

- **Dual-mode** ⇒ OS protects itself and others
  - **User mode** and **kernel mode**
  - **Mode bit** provided by hardware
    - Distinguish user code or kernel code
    - **Privileged** instructions ⇒ executable only in kernel mode
    - System call changes mode to kernel, and back before returning
Resource Hogs

- Timer to prevent infinite loop / process hogging resources
  - Set interrupt after specific period
  - Operating system decrements counter
  - When counter zero generate an interrupt
  - Set up before scheduling process to regain control or terminate program that exceeds allotted time

Process Management

- A process = program in execution.
- Process needs resources: CPU, memory, I/O, files
- Process termination: reclaim reusable resources
- Single-threaded process - one program counter
- Multi-threaded process - one program counter per thread
- Many processes running concurrently on one or more CPUs
  - Concurrency by multiplexing the CPUs among the processes / threads
Process Management

OS responsibilities:
- Creating and deleting processes
- Suspending and resuming processes
- Providing mechanisms for
  - process synchronization
  - process communication
  - deadlock handling

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 when
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes and data to move into and out of memory
  - Allocating and deallocating memory space as needed
**Storage Management**

- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - file
  - Each medium is controlled by device (i.e., disk drive, tape drive)

- File-System management
  - Files usually organized into directories
  - Access control - who can access what
  - OS activities include
    - Creating and deleting files and directories
    - Manipulate files and dirs
    - Mapping files onto secondary storage
    - Backup files onto stable (non-volatile) storage media

---

**Mass-Storage Management**

- Disks store data that does not fit in main memory or 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

- Tertiary storage
  - Optical storage, magnetic tape
  - Managed by OS or applications
  - Varies between WORM (write-once, read-many-times) and RW (read-write)
Performance of Various Levels of Storage

- Movement between levels of storage hierarchy can be explicit or implicit

<table>
<thead>
<tr>
<th>Level</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>registers</td>
<td>cache</td>
<td>main memory</td>
<td>disk storage</td>
</tr>
<tr>
<td>Typical size</td>
<td>&lt; 1 KB</td>
<td>&gt; 16 MB</td>
<td>&gt; 16 GB</td>
<td>&gt; 100 GB</td>
</tr>
<tr>
<td>Implementation technology</td>
<td>custom memory with multiple ports, CMOS</td>
<td>on-chip or off-chip CMOS DRAM</td>
<td>magnetic disk</td>
<td></td>
</tr>
<tr>
<td>Access time (ns)</td>
<td>0.25 – 0.5</td>
<td>0.5 – 25</td>
<td>80 – 250</td>
<td>5,000,000</td>
</tr>
<tr>
<td>Bandwidth (MB/sec)</td>
<td>20,000 – 100,000</td>
<td>5000 – 10,000</td>
<td>1000 – 5000</td>
<td>20 – 150</td>
</tr>
<tr>
<td>Managed by</td>
<td>compiler</td>
<td>hardware</td>
<td>operating system</td>
<td>operating system</td>
</tr>
<tr>
<td>Backed by</td>
<td>cache</td>
<td>main memory</td>
<td>disk</td>
<td>CD or tape</td>
</tr>
</tbody>
</table>

Migration of Integer A from Disk to Register

- Multitasking environments must be careful to use most recent value, no matter where it is stored in the storage hierarchy

- Multiprocessor environment must provide cache coherency in hardware - all CPUs have the most recent value in their cache

- Distributed environment situation even more complex
  - Several copies of a datum can exist
I/O Subsystem

- One purpose of OS is to hide peculiarities of hardware devices from the user
- I/O subsystem responsible for
  - Memory management of I/O (buffering, caching, spooling)
  - General device-driver interface
  - Drivers for specific hardware devices

Protection and Security

- **Protection** – any mechanism for controlling access of processes or users to resources defined by the OS
- **Security** – defense of the system against internal and external attacks
  - Denial-of-service, worms, viruses, identity theft, theft of service
- Who can do what
  - **User IDs** – name and associated number
  - User ID associated with all files to determine access control
  - **Group ID** – set of users
  - **Privilege escalation** – user change to effective ID with more rights
Distributed Computing

- Collection of separate, possibly heterogeneous, systems networked together
  - Local Area Network (LAN)
  - Wide Area Network (WAN)
  - Metropolitan Area Network (MAN)

- Network OS provides features between systems across network
  - Communication scheme – exchange messages
  - Illusion of a single system

Special-Purpose Systems

- Real-time embedded systems – most prevalent form of computers
  - Special/limited purpose OS, real-time OS

- Multimedia systems
  - Streams of data must be delivered according to time constraints

- Handheld systems
  - PDAs, smart phones, limited CPU, memory, power
  - Reduced feature set OS, limited I/O
Computing Environments

- Traditional computer
  - Blurring over time
  - Office environment
  - PCs connected to a network, terminals attached to mainframe or minicomputers providing batch and timesharing
  - Now portals allowing networked and remote systems access to same resources
- Home networks
  - Used to be single system, then modems
  - Now networked, firewalled

Computing Environments

Client-Server Computing
- Many systems now servers, responding to requests generated by clients
  - **Compute-server** – interface to client requests (i.e., database)
  - **File-server** – interface for clients to store and retrieve files
Peer-to-Peer Computing

- Another model of distributed system

- P2P does not distinguish clients and servers
  - Instead all nodes are considered peers
  - May each act as client, server or both
  - Node must join P2P network
    - Registers its service with central lookup service on network
    - Broadcast request for service and respond to requests for service via discovery protocol
  - Examples include Napster and Gnutella

Web-Based Computing

- Web has become ubiquitous
- PCs most prevalent devices
- More devices becoming networked to allow web access
- New category of devices to manage web traffic among similar servers: load balancers
- OS allow computers to be clients and servers
Open-Source OS

- Operating systems made available in source-code rather than binary (closed-source)
- Counter to the copy protection and Digital Rights Management (DRM) movement
- Started by Free Software Foundation (FSF), which has “copyleft” GNU Public License (GPL)
- Examples include GNU/Linux and BSD UNIX (including core of Mac OS X), and many more

Homework Assignment #1

Download the homework assignment off the schedule web page.

- Download Virtual Machine Player to your personal computer. If you do not have one, you can install it on the Dept.’s computers. Make sure you erase the virtual machine files after the exercise is finished (they are heavy files).
  - https://my.vmware.com/download/player

- Or Virtual Box
  - https://www.virtualbox.org/wiki/Downloads
Homework Assignment #1

- Download Ubuntu and install it in your Virtual machine
  - http://www.ubuntu.com/download/desktop

- Alternatively, download a Virtual Machine containing Ubuntu
  - https://solutionexchange.vmware.com/store

- Install a second OS: Windows, Debian, Suse, etc.

- Setup Grub to select the booting OS.