Republic of the Philippines
Isabela State University
Echague, Isabela

College of Computing Studies, Information and Communication Technology

Chapter II

Operating Systems Basics

Introduction to Operating Systems

1. Functions and Services of an Operating System

Objectives

1. Understand the definition and role of an Operating System (OS) in computing.
2. Learn about the primary functions of an OS and their importance.
3. Identify and distinguish between different types of Operating Systems.
4. Recognize various examples of Operating Systems used in different contexts.

Activities

1. **Interactive Discussion:** Ask students what they think an OS does. Write their responses on the board.
2. **Case Studies:** Present real-world scenarios where different OS functions are critical.
3. **Group Activity:** Have students work in groups to compare and contrast different types of OS.
4. **Q&A Session:** Encourage students to ask questions and discuss.

Definition of an Operating System (OS)

- **Definition:** An Operating System (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs.

  Role and Importance in Computing:

  - **Interface Between User and Hardware:** OS acts as an intermediary, allowing users to interact with the computer hardware without needing to understand the complex details.
  - **Resource Management:** Efficiently manages hardware resources like CPU, memory, and storage.
  - **Execution of Applications:** Provides a platform for running application software.

- **Security and Access Control:** Ensures data security and access management for users.

Primary Functions of an OS

- **Process Management:**
  - **Definition:** Manages the execution of processes, including multitasking and process synchronization.
  - **Importance:** Ensures efficient execution of multiple processes and optimal CPU utilization.
- **Memory Management:**
  - **Definition:** Manages computer memory, including allocation and deallocation of memory spaces as needed by programs.
  - **Importance:** Prevents memory leaks, optimizes memory usage, and ensures processes have necessary memory resources.
- **File System Management:**
  - **Definition:** Manages files and directories, providing a way to store, retrieve, and update data on storage devices.
  - **Importance:** Organizes data for easy access and management, ensuring data integrity and security.
- **Device Management:**
  - **Definition:** Manages hardware devices, providing drivers and interfaces for interaction.
  - **Importance:** Ensures smooth communication between hardware and software components.
- **Security and Access Control:**
  - **Definition:** Protects data and resources from unauthorized access and ensures user authentication.
  - **Importance:** Maintains system integrity and confidentiality of data.

Types of Operating Systems

- **Single-user and Multi-user:**
  - **Single-user:** Designed for one user at a time (e.g., Windows 10).
  - **Multi-user:** Allows multiple users to use the computer simultaneously (e.g., UNIX).
- **Multi-tasking and Single-tasking:**
  - **Multi-tasking:** Can run multiple tasks (processes) at the same time (e.g., Windows, Linux).
  - **Single-tasking:** Can run only one task at a time (e.g., DOS).
- **Real-time OS:**
  - **Definition:** Provides real-time processing and response to inputs (e.g., RTOS used in embedded systems).
  - **Importance:** Critical for applications requiring timely and predictable response (e.g., medical systems, industrial control).

- **Distributed OS:**
    - **Definition:** Manages a group of independent computers and makes them appear as a single computer (e.g., distributed computing systems).
    - **Importance:** Enhances computational power and resource sharing.

Examples of Operating Systems

- **Windows:** Popular desktop OS with a user-friendly interface, widely used in personal and business environments.
- **Linux:** Open-source OS known for its security and stability, used in servers, desktops, and embedded systems.
- **macOS:** Apple's proprietary OS for Mac computers, known for its sleek design and robust performance.
- **Android:** Mobile OS developed by Google, dominant in the smartphone market.
- **iOS:** Apple's proprietary mobile OS for iPhones and iPads, known for its smooth user experience and integration with Apple ecosystem.

2. Overview of Windows and Linux (for theoretical knowledge)

Objectives

1. Understand the history and evolution of Windows and Linux operating systems.
2. Learn about the key features and components of both Windows and Linux.
3. Identify common versions and distributions of Windows and Linux.
4. Recognize use cases and applications of both operating systems.
5. Conduct a comparative analysis of Windows and Linux, focusing on architecture, file systems, and command line interfaces.
6. Discuss the pros and cons of each operating system and scenarios where each excels.

Activities

1. **Timeline Creation:** Have students create a timeline of key events in the history of Windows and Linux.
2. **Feature Comparison Chart:** Students work in groups to create a comparison chart of features and components of Windows and Linux.
3. **Case Study Analysis:** Present real-world scenarios and have students analyze which OS would be more suitable.
4. **Class Debate:** Organize a debate where students argue the pros and cons of Windows vs. Linux.

Windows Operating System

- **Brief History and Evolution:**
  - **Early Days:** Introduced by Microsoft in 1985 as a graphical extension for MS-DOS.
  - **Key Milestones:** Windows 3.0 (1990), Windows 95 (1995), Windows XP (2001), Windows 7 (2009), Windows 10 (2015), Windows 11 (2021).
  - **Current State:** Dominant in the personal computing market with a significant presence in enterprise environments.
- **Key Features and Components:**
  - **Graphical User Interface (GUI):** User-friendly interface with start menu, taskbar, and windowed applications.
  - **File System:** Uses NTFS (New Technology File System) primarily.
  - **System Components:** Windows Explorer, Control Panel, Device Manager, Task Manager.
  - **Security Features:** Windows Defender, User Account Control (UAC), BitLocker encryption.
- **Common Versions:**
  - **Windows 10:** Introduced in 2015, known for its hybrid start menu and Cortana integration.
  - **Windows 11:** Released in 2021, features a redesigned start menu, taskbar, and improved performance.
- **Use Cases and Applications:**
  - **Personal Computing:** Widely used in homes for general computing tasks.
  - **Business and Enterprise:** Common in office environments for productivity applications.
  - **Gaming:** Preferred by many gamers due to broad software compatibility.

Linux Operating System

- **Brief History and Evolution:**
  - **Origins:** Created by Linus Torvalds in 1991 as a free and open-source alternative to UNIX.
  - **Key Milestones:** Development of various distributions, adoption in servers and supercomputing, growth of desktop environments.
  - **Current State:** Popular in servers, cloud environments, embedded systems, and increasingly on desktops.
- **Key Features and Components:**
  - **Open Source:** Source code is freely available, allowing for customization and community-driven development.
  - **File System:** Commonly uses ext4 (Fourth Extended Filesystem) but supports others like Btrfs, XFS, and more.
  - **System Components:** Kernel, shell (Bash), package managers (APT, YUM), desktop environments (GNOME, KDE).
  - **Security Features:** SELinux (Security-Enhanced Linux), AppArmor, iptables for firewall configurations.

- **Common Distributions:**
    - **Ubuntu:** User-friendly, popular for desktops and servers, based on Debian.
    - **Fedora:** Cutting-edge features, sponsored by Red Hat.
    - **Debian:** Known for stability and extensive software repositories.
- **Use Cases and Applications:**
    - **Servers:** Dominant in web servers, database servers, and other server applications.
    - **Development:** Preferred by developers for its robust tools and open-source nature.
    - **Embedded Systems:** Used in devices like routers, smart TVs, and IoT devices.

Comparative Analysis: Windows vs. Linux

- **Differences in Architecture:**
    - **Kernel Design:** Windows uses a hybrid kernel, while Linux uses a monolithic kernel.
    - **File Systems:** Windows primarily uses NTFS, while Linux uses ext4, Btrfs, etc.
    - **Command Line Interfaces:** Windows uses Command Prompt and PowerShell; Linux uses various shells like Bash, Zsh.
- **Pros and Cons of Each OS:**
    - **Windows Pros:** User-friendly GUI, extensive software compatibility, strong support for gaming.
    - **Windows Cons:** Prone to malware, licensing costs, less control over system internals.
    - **Linux Pros:** Free and open-source, highly customizable, strong security features.
    - **Linux Cons:** Steeper learning curve, software compatibility issues, less user-friendly for non-technical users.
- **Scenarios Where Each OS Excels:**
    - **Windows:** Best for general users, gamers, businesses relying on Microsoft software.
    - **Linux:** Ideal for servers, development environments, users needing high customization and security.

Laboratory (6 hours)

Basic OS Navigation on Windows

Objectives

1. Master the use of Windows File Explorer and Command Prompt for file system navigation and management.
2. Learn to install and use Windows Subsystem for Linux (WSL) to run Linux commands on Windows.
3. Understand the integration between Windows and WSL for seamless operation of both environments.

Activities

1. **Hands-on Exercises:** Practice navigating and managing files using both File Explorer and Command Prompt.
2. **WSL Installation and Configuration:** Install WSL and set up a Linux distribution.
3. **Linux Command Practice:** Execute basic Linux commands and integrate Windows and Linux environments.

---

1. Using Windows File Explorer and Command Prompt

File Explorer

- **Objective:** Navigate the Windows file system and manage files and folders.
- **Activity:**
  1. **Navigating the File System:** Open File Explorer, navigate to different drives and directories, and locate specific files.
  2. **Creating, Renaming, and Deleting Files and Folders:** Create new folders, rename files, and delete unnecessary items.
  3. **Copying and Moving Files and Folders:** Copy files from one directory to another and move folders between locations.
  4. **Using Search Functionality:** Use the search bar in File Explorer to find files and folders quickly.

Command Prompt

- **Objective:** Navigate directories and manage files using Command Prompt.
- **Activity:**
  1. **Opening and Using Command Prompt:** Open Command Prompt and navigate the file system using basic commands.
  2. **Basic Commands:**
     - `dir`: List directory contents
     - `cd`: Change directory
     - `copy`: Copy files
     - `move`: Move files
     - `del`: Delete files
     - `mkdir`: Create directories
     - `rmdir`: Remove directories
  3. **Navigating Directories and Managing Files:** Practice changing directories, listing contents, and managing files and folders using the above commands.
  4. **Understanding and Setting Environment Variables:** Learn to view and set environment variables using `set` and `echo %VARIABLE%`.

Detailed Steps:

1. **Navigating the File System:**

- Open File Explorer.
- Navigate to `C:\` and explore various folders.
- Locate specific files in the Documents, Downloads, and Pictures directories.

2. **Creating, Renaming, and Deleting Files and Folders:**
   - Create a new folder named "LabTest" on the desktop.
   - Create a new text file inside "LabTest" and rename it to "testfile.txt".
   - Delete "testfile.txt".
3. **Copying and Moving Files and Folders:**
   - Copy a file from the Downloads folder to the "LabTest" folder.
   - Move a file from the "LabTest" folder to the Documents folder.
4. **Using Search Functionality:**
   - Use the search bar in File Explorer to find all files with the ".txt" extension.
5. **Opening and Using Command Prompt:**
   - Open Command Prompt by typing `cmd` in the search bar and hitting Enter.
   - Navigate to the "LabTest" folder using `cd Desktop\LabTest`.
6. **Basic Commands:**
   - Use `dir` to list contents.
   - Use `cd ..` to go back to the Desktop directory.
   - Use `mkdir` to create a new directory.
   - Use `copy` and `move` to manage files.
   - Use `del` to delete files and `rmdir` to remove directories.
7. **Navigating Directories and Managing Files:**
   - Practice navigating different directories and managing files using Command Prompt commands.
8. **Understanding and Setting Environment Variables:**
   - View environment variables with `set`.
   - Set a new environment variable with `set MYVAR=HelloWorld`.
   - Display the value of the variable using `echo %MYVAR%`.

2. Installing Windows Subsystem for Linux (WSL) to Run Linux Commands

Introduction to WSL

- **Objective:** Understand what WSL is and why it is useful.
- **Activity:** Discuss the benefits and use cases of WSL.

Installing WSL

- **Objective:** Install and configure WSL on Windows.
- **Activity:**
  1. **Enabling WSL Feature in Windows:**
     - Open PowerShell as Administrator.
     - Run `wsl --install` to enable WSL and install the default Linux distribution (e.g., Ubuntu).
  2. **Installing a Linux Distribution:**

        - From the Microsoft Store, choose and install a Linux distribution like Ubuntu.

Using WSL

- **Objective:** Run Linux commands on Windows.
- **Activity:**
    1. **Running Linux Commands:**
        - Open the installed Linux distribution (e.g., Ubuntu).
        - Practice basic Linux commands:
            - `ls`: List directory contents
            - `cd`: Change directory
            - `cp`: Copy files
            - `mv`: Move files
            - `rm`: Delete files
            - `mkdir`: Create directories
            - `rmdir`: Remove directories
    2. **Accessing Windows File System from WSL:**
        - Navigate to Windows file system from WSL using `/mnt/c/Users/YourUsername`.

Practical Exercises:

1. **Navigating Directories:**
    - Navigate to different directories using `cd` and `ls`.
2. **Creating, Editing, and Deleting Files:**
    - Use `nano` or `vim` to create and edit files.
    - Use `rm` to delete files.
3. **Using Linux Text Editors:**
    - Practice editing text files using `nano` and `vim`.

Integration Between Windows and WSL

- **Objective:** Understand the integration between Windows and WSL.
- **Activity:**
    1. **Sharing Files Between Windows and WSL:**
        - Copy files between Windows and WSL file systems.
    2. **Running Windows Applications from WSL:**
        - Run a Windows application (e.g., Notepad) from within WSL using `notepad.exe`.

Detailed Steps:

1. **Enabling WSL Feature in Windows:**
    - Open PowerShell as Administrator.
    - Run `wsl --install`.

   - Restart the computer if prompted.
2. **Installing a Linux Distribution:**
   - Open Microsoft Store.
   - Search for "Ubuntu" and click Install.
3. **Running Linux Commands:**
   - Open Ubuntu from the Start menu.
   - Use `ls`, `cd`, `cp`, `mv`, `rm`, `mkdir`, and `rmdir` to navigate and manage files.
4. **Accessing Windows File System from WSL:**
   - In the WSL terminal, navigate to the Windows file system using `cd /mnt/c/Users/YourUsername`.
5. **Practical Exercises:**
   - Navigate directories with `cd` and `ls`.
   - Create and edit files with `nano` and `vim`.
   - Delete files with `rm`.
6. **Integration Between Windows and WSL:**
   - Copy files between Windows and WSL using `cp` and `mv`.
   - Run `notepad.exe` from within WSL to open Notepad.

Supplementary Materials

Reading Material:

- Textbook chapters on Operating Systems basics (specific chapters based on your course textbook)
- Online articles and tutorials on Windows and Linux OS

Tools and Resources:

- Windows 10/11 Operating System
- Internet access for downloading WSL and Linux distributions
- Text editor (e.g., Notepad++, Visual Studio Code)

Assignments:

- **Lecture Assignment:** Write a comparative essay on Windows and Linux, highlighting their key differences and similarities.
- **Lab Assignment:** Create a step-by-step guide on navigating the Windows file system using both File Explorer and Command Prompt, and another guide on using basic Linux commands in WSL.