

**Bangabandhu Sheikh Mujibur Rahman Digital University, Bangladesh**

**Faculty of Cyber Physical Systems**

**Department of Internet of Things and Robotics Engineering**

**Report Information:**

**Course Title:** Operating Systems Lab

**Course Code:** ICT 4260

**Submitted To:**

Suman Saha

Lecturer,

Department of IRE, BDU

**Submitted By:**

Avik Halder (2001009)

Akibul Hasan Anik(2001010)

Hrithik Das (2001016)

**Session:** 2020-21

**Date of Submission:** 25/05/2024

**Project Title:**

Calculation and file management Operating System

**Introduction:**

The VOID() Operating System project aims to develop a basic yet functional operating system using the Cosmos framework, Visual Studio, and the C# programming language. This project includes fundamental OS functionalities such as file and directory management, a calculator, basic scheduling algorithms, and memory management techniques. The report details the project's objectives, development process, functionalities, and future improvements.

**Objectives:**

* Develop a functional operating system kernel using C# and the Cosmos framework.
* Implement basic OS functionalities such as file and directory operations.
* Incorporate basic scheduling algorithms (FCFS, SJF) and memory management techniques (First-Fit, Best-Fit, Worst-Fit).
* Provide a simple command-line interface for user interaction.

**Tools and Technologies:**

* **Cosmos:** An open-source operating system toolkit that allows developers to create OS kernels using C#.
* **Visual Studio:** An integrated development environment (IDE) used for coding, debugging, and testing the OS.
* **C#:** The programming language used to develop the OS kernel and its functionalities.

**Methodology:**

* Install and Setup Visual Studio and COSMOS.
* Writing code in C#
* Run on VMware Workstation.

**Functionalities:**

**Basic Commands**

* **help:** Displays a list of available commands.
* **about:** Displays the OS version information.
* **date:** Shows the current date.
* **time:** Shows the current time.
* **day:** Displays the current day.
* **clear:** Clears the console.
* **calculator:** Provides basic arithmetic operations (addition, subtraction, multiplication, division).
* **shutdown:** Shuts down the OS.

**File Operations**

* **create\_file:** Creates a new file in the current directory.
* **write\_file:** Writes content to an existing file.
* **append\_file:** Appends content to an existing file.
* **show\_file:** Displays the content of a file.
* **delete\_file:** Deletes a specified file.
* **list\_file:** Lists all files in the current directory.

**Directory Operations**

* **create\_directory:** Creates a new directory.
* **list\_directory:** Lists all directories in the current directory.
* **change\_directory:** Changes the current directory.
* **delete\_directory:** Deletes a specified directory.
* **back:** Navigates to the parent directory.
* **current\_directory:** Displays the current directory path.

**Scheduling Algorithms**

* **fcfs:** Implements the First-Come, First-Served scheduling algorithm.
* **sjf:** Implements the Shortest Job First scheduling algorithm.

**Memory Management**

* **first\_fit:** Implements the First-Fit memory allocation algorithm.
* **best\_fit:** Implements the Best-Fit memory allocation algorithm.
* **worst\_fit:** Implements the Worst-Fit memory allocation algorithm.

**Challenges and Solutions:**

During the development process, several challenges were encountered, including:

* **File System Operations:** Handling file and directory operations within the constraints of the Cosmos VFS required careful implementation and error handling.
* **Command Parsing:** Efficiently parsing and executing user commands required a structured approach to ensure accurate and responsive command execution.

Solutions to these challenges involved thorough testing, debugging, and the implementation of robust error-handling mechanisms.

**Future Improvements:**

Future enhancements to the VOID() Operating System could include:

* **Graphical User Interface (GUI):** Developing a basic GUI to improve user interaction.
* **Networking Support:** Adding network functionalities to enable communication between devices.
* **Advanced Scheduling:** Implementing more advanced scheduling algorithms like Round Robin and Priority Scheduling.
* **Enhanced Memory Management:** Incorporating more sophisticated memory management techniques.

**Conclusion:**

The VOID() Operating System project successfully achieved its primary objectives by developing a basic yet functional OS with essential features. Utilizing Cosmos, Visual Studio, and C#, the project demonstrates a foundational understanding of OS concepts and provides a platform for further exploration and enhancement in operating system development.

**References:**

* **Cosmos:** <https://github.com/CosmosOS/Cosmos>
* **Visual Studio:** <https://visualstudio.microsoft.com/>
* **C# Programming Language**: <https://docs.microsoft.com/eus/dotnet/csharp/>