Published online on the journal website:https://publikasi.mercubuana.ac.id/index.php/fasilkom/

Collabits Journal

Vol 1, No 1 (2024) ISSN :

CONSTRUCTION AND MANAGEMENT OF A COSMOS-BASED OPERATING SYSTEM USING VISUAL STUDIO DEVELOPMENT AND VMWARE VIRTUALIZATION TECHNOLOGY

Mohamad Yusuf[1], Zidane Fahrezi[2], Rafif Syari Hidayah[3], Yudha Andika Istanto[4], Gilas Adi Saputra[5]

[1,2,3,4,5] Informatics Engineering Study Program, Mercu Buana University[1] *mhd.yusuf@mercubuana.ac.id*, [2]*aezidan10@gmail.com,*[3]rafanime3@gmail.com, *4yudhaandikaistanto@gmail.com,*[5]adigilas88@gmail.com

Abstract

Operating systems play an important role in bridging hardware and software on various computing devices. This research focuses on building an operating system based on Cosmos, an open source project that allows the creation of operating system kernels quickly and efficiently. In the process, we leverage Visual Studio development tools to develop and maintain the kernel, while VMware virtualization technology is used to test and manage development. This research contributes to further understanding of the development of Cosmos-based operating systems with optimal use of Visual Studio development tools and VMware virtualization technology.

Keywords : *Operating System*, Cosmos*, Visual Studio, VMware*

1. Introduction

This research covers the detailed steps taken in building a Cosmos-based operating system, from initial kernel design to overall implementation development. We dig deeper into the use of Visual Studio development tools, highlighting the advantages and ease of integration it offers in the kernel coding and debugging process. Additionally, we document our experience in using VMware virtualization technology as a powerful and efficient testing tool to ensure the stability and reliability of operating systems under development. The research methodology involves a series of detailed experiments and tests, with a focus on optimizing kernel performance and module integration.

We discuss the challenges and solutions encountered during development, highlighting the use of Cosmos structures to simplify resource management and increase efficiency. Additionally, this research evaluates the impact of using Visual Studio development tools and VMware virtualization technology on the overall development cycle.

The results of this research not only provide deep insight into the development of Cosmos-based operating systems, but also provide practical guidance for developers interested in using Visual Studio development tools and VMware virtualization technologies in similar projects. It is hoped that this contribution can enrich literature related to operating system development.

The following is the software used in the research:

1.1 VMWare, VMWare is a software designed to manage virtual machines. Its functions include executing multiple operating systems on a single piece of hardware, as well as running applications designed for other operating systems. Apart from that, VMWare also plays a role in facilitating learning and development of operating systems by allowing users to run several operating systems simultaneously on one PC machine without the need to repartition or reboot. By using a virtual machine, users can easily run the desired operating system according to their needs (Ramdan, 2016). With this method, users can boot an operating system as a host operating system, then run another operating system, for example MS Windows.

1.2 Visual Studio 2022, Use of Visual Studio 2022 as an effective text editing tool in operating system development using the C++ programming language. With an intuitive interface and a variety of powerful development features, Visual Studio 2022 provides an ideal environment for exploring the complexities of operating system development (Habib, Syafii, Rudi, Moranain,

& Welly, 2022)

1.3 Cosmos, Cosmos (C# Open Source Managed Operating System) is a development platform for operating systems that primarily uses Visual Studio as a development environment (Giska, Imam, Ira, & Falaah, 2021). Even though it uses the name C#, various .NET-based languages, such as VB.NET, Fortran, Delphi Prism, IronPython, F#, and others, can be used in development. Cosmos' main resources and kernel routines are mostly written in the C# programming language, and that is why it is named Cosmos.

2. Conceptual Stage

This research is focused on developing a Cosmos-based operating system, with three main stages in its approach. The conceptual stage includes understanding the basic concept, initial design, and development approach. Basic concepts of operating systems discuss the basic principles and their role in bridging hardware and software, while understanding Cosmos as an operating system development platform is also emphasized. Initial kernel design involves explaining the foundations for further development and identifying the requirements and key features of the kernel.

The use of Visual Studio as a development tool becomes the focus in the next stage. The utilization of this tool in the kernel coding and debugging process is highlighted, along with an understanding of the advantages and ease of integration it provides. Next, the testing and development management phase using VMware addresses the role of virtualization technology in ensuring the stability and reliability of the operating system.

In the final stage, research includes kernel performance optimization and module integration. This research methodology focuses detailed experiments and testing to improve kernel performance, while describing module integration to increase efficiency. Challenges that may be encountered during development are identified, and solutions and strategies implemented to overcome these obstacles are also considered. Evaluation of the impact of using Visual Studio development tools and VMware technologies on the overall development cycle highlights the positive contributions and problem solving brought about by both tools.

3. Experimental Setup

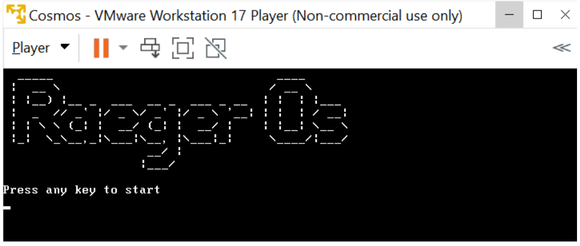
This Experimental Setup discusses detailed steps in building a Cosmos-based operating system using Visual Studio and VMware virtualization technology. First, the development environment setup is explained, including the installation of Visual Studio 2022 and the configuration of the Cosmos project for kernel development. Then, the focus shifted to using Cosmos as a development framework, with an emphasis on the C# programming language to simplify the development process.

Next, the Visual Studio 2022 integration section highlights the benefits and ease of use of this tool in coding and debugging operating system kernels. The use of VMware as a testing platform is also explained in detail, including virtual machine configuration and operating system integration for testing and evaluating functionality. The functionality testing process discusses the testing stages, including the operating system's ability to manage files and directories.

Kernel performance evaluation is the next focus, with an explanation of the parameters and metrics used to measure kernel reliability and performance. Visual documentation includes images of the appearance of the Raeger Os operating system on VMware, providing visualization of implementation results such as logos, menus, commands, and results of operations such as creating and deleting files. Finally, the use of Raeger Os operating system commands is explained to provide a complete picture of the experimental results of this setup. This entire experiment provides an in-depth understanding of the development process of a Cosmos-based operating system with using Visual Studio and VMware, including detailed performance evaluation

4. Results and Analysis

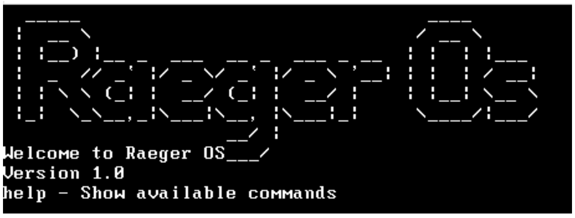
At this stage we use Vmware as a tool to visualize and test the Operating System that we have created. The results achieved are as follows:


Cosmos - VMware Workstation 17 Player (Non-commercial use only)
Player

```
Press any key to start
```

Figure 1: Operating System View*RaegerOS*

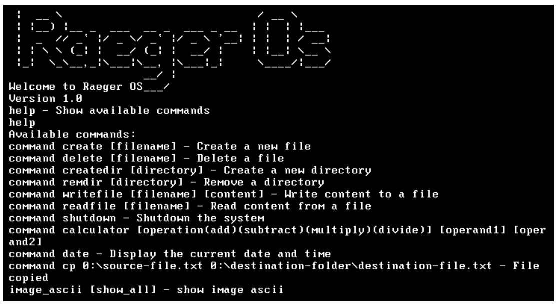
At this stage the Raeger Os logo is displayed which is made from ASCII ART (ASCII art is a form of art that uses ASCII characters to represent visual information. This art consists of 95 ASCII characters that can be printed (Katsunori, Henry, & Tomoyuki, 2011)), the user can press any button to start or go to the next stage of the Operating System (Raeger Os) that we created.



```
Welcome to Raeger OS___/
Version 1.0
help - Show available commands
```

Figure 2: Operating System View*RaegerOS*

At this stage, a menu displays that says "Help – Show available commands" which shows a guide for starting commands in the Raeger Os operating system.

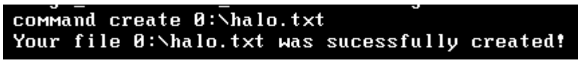


```
Welcome to Raeger OS___/
Version 1.0
help - Show available commands
help
Available commands:
command create [filename] - Create a new file
command delete [filename] - Delete a file
command createdir [directory] - Create a new directory
command remdir [directory] - Remove a directory
command writefile [filename] [content] - Write content to a file
command readfile [filename] - Read content from a file
command shutdown - Shutdown the system
command calculator [operation(add)(subtract)(multiply)(divide)] [operand1] [oper
and2]
command date - Display the current date and time
command cp 0:\source-file.txt 0:\destination-folder\destination-file.txt - File
copied
image_ascii [show_all] - show image ascii
```

Figure 3: Help display from the Operating System*RaegerOS*

At this stage several commands are displayed to run the operating system, which consist of:

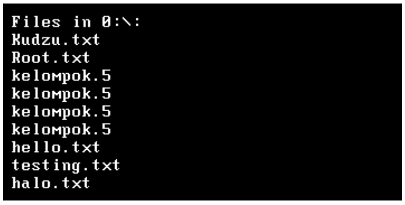
1. Command create [filename] – Create a new file. In this command the user can create a new file by typing the command “command create 0:\[Name of the file you want to create] as shown in the following image:



```
command create 0:\halo.txt
Your file 0:\halo.txt was sucessfully created!
```

Figure 4: Display of Create Operating System*RaegerOS*

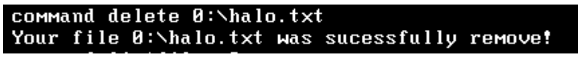
2. Command delete [filename] – Delete a file. In this command the user can delete a file from the specified directory by typing the command "command delete 0:\[name of the file you want to delete] as shown in the following image:



```
Files in 0:\:
Kudzu.txt
Root.txt
kelompok.5
kelompok.5
kelompok.5
kelompok.5
hello.txt
testing.txt
halo.txt
```

Figure 5: Display of the Operating System listfiles*RaegerOS*

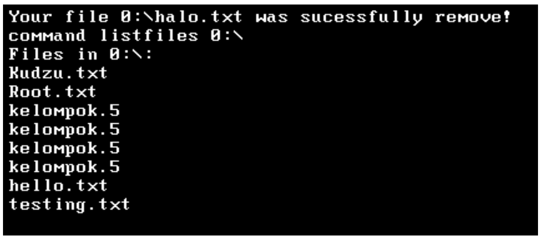
This image shows a list of files that have been created by typing the command "command listfiles 0:\" . You can see in Figure 5 that you can see a file called halo.txt, in this file we will delete the file from the directory.



```
command delete 0:\halo.txt
Your file 0:\halo.txt was sucessfully remove!
```

Figure 6: Display of the Operating System delete files command *RaegerOS*

In Figure 6, after writing the command "command delete 0:\halo.txt" the system will notify you that the file has been deleted from the specified directory.

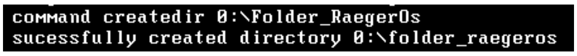


```
Your file 0:\halo.txt was sucessfully remove!
command listfiles 0:\
Files in 0:\:
Kudzu.txt
Root.txt
kelompok.5
kelompok.5
kelompok.5
kelompok.5
hello.txt
testing.txt
```

Figure 7: Display of Operating System listfiles*RaegerOS*

In Figure 7, you can see that the file called halo.txt is no longer available because we have deleted it from that directory.

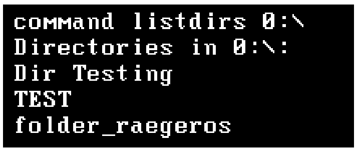
3. Command createdir [directory] – create new directory. In this command the user can create a folder by typing the command "Command createdir 0:\[Directory Name] as shown in the following image:



```
command createdir 0:\Folder_RaegerOs
sucessfully created directory 0:\folder_raegeros
```

Figure 8: Display of the Operating System folder create command *RaegerOS*

If you have typed the command "command createdir 0:\Folder_RaegerOs" the system will tell you that your folder has been successfully created as in Figure 8.

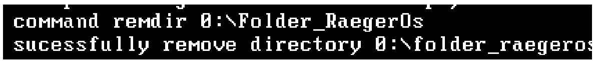


```
command listdirs 0:\
Directories in 0:\:
Dir Testing
TEST
folder_raegeros
```

Figure 9: Display of the Operating System folder list*RaegerOS*

In Figure 9, if it has been successfully created it will be saved in the directory list. By typing the command "command listdirs 0:\" we can see what directories we have created.

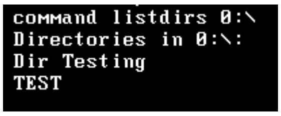
4. Command rmdir [directory] – Remove a directory. In this command the user can create a folder by typing the command "Command rmdir 0:\[name of the directory you want to delete] as shown in the following image:



```
command remdir 0:\Folder_RaegerOs
sucessfully remove directory 0:\folder_raegero
```

Figure 10: Display of the Operating System Remove Command *RaegerOS*

If you have typed the command "command rmdir 0:\Folder_RaegerOs" the system will tell you that your folder has been successfully deleted as in Figure 10.

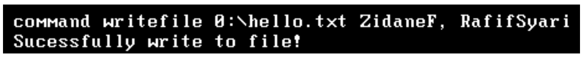


```
command listdirs 0:\
Directories in 0:\:
Dir Testing
TEST
```

Figure 11: Display of the Operating System folder list*RaegerOS*

In Figure 11, there is no name for Folder_RaegerOs because the folder has been deleted.

5. Command writrefile [filename] [content] – write content to a file. In this command, users can write the file they want to create by typing the command "command writefile [the file they want to write] as shown in the following image:

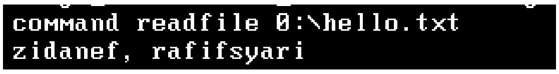


```
command writefile 0:\hello.txt ZidaneF, RafifSyari
Sucessfully write to file!
```

Figure 12: Display of the Operating System Writefile Command
RaegerOS

If you have typed the command "command writefile 0:\hello.txt ZidaneF, RafifSyari" the system will notify you that the command was created successfully.

6. Command readfile [filename] – read content from file. In this command the user can read the file that the user wants to create by typing the command "command readfile [the file you want to read] as shown in the following image:

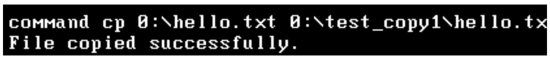


```
command readfile 0:\hello.txt
zidanef, rafifsyari
```

Figure 13: Display of the Operating System Readfile Command
RaegerOS

If you have typed the command "command redfile 0:\hello.txt" the system will read the file you want to read as in Figure 13.

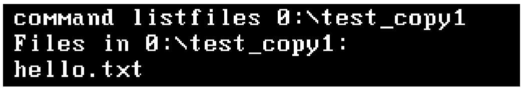
7. Command cp 0:\[filename] 0:\[filename]. In this command the user can copy the files the user wants to another folder. Like the picture below



```
command cp 0:\hello.txt 0:\test_copy1\hello.tx
File copied successfully.
```

Figure 14: Operating System Copy Command Display
RaegerOS

If you have typed the command "command cp 0:\hello.txt 0:\test_copy1\hello.txt" the system will notify you that the file has been successfully copied. Figure



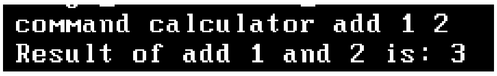
```
command listfiles 0:\test_copy1
Files in 0:\test_copy1:
hello.txt
```

15: Operating System File Folder List Display
RaegerOS

In Figure 15, if you have successfully copied the file to another folder, the user can type the command "command listfiles 0:\test_copy1" to display the list from the test_copy1 folder. You can see in figure 15 that the file is already in the test_copy1 folder.

8. Command shutdown – Shutdown the system. In this command the system will shutdown (turn off the operating system running on Vmware).

9. Command calculator [operation (add) (subtract) (multiply) (divide)]. In this command the user can use an available calculator, the calculator can calculate addition, multiplication, division, subtraction. Like the example shown below:



```
command calculator add 1 2
Result of add 1 and 2 is: 3
```

Figure 16: Calculator Command Display on the Operating System*RaegerOS*

In Figure 16, the user types the command "command calculator add 1 2" which is the command for the calculator to calculate (1 + 2) with the result being 3

5. System File Management Code Analysis

The class in the C# programming language used to create virtual file management is the VFSManager Class which is used to manage files and directories in a virtual file system. A virtual file system is a way to represent data stored across multiple storage media as a consistent and unified file structure. The VFSManager class provides methods for registering, initializing, and accessing different virtual file systems. Cosmos has a path in the format 0:\ to manage the file system.

In creating file system management, there are several methods from the VFSManager class as below:

1. Create File()

   public static DirectoryEntry CreateFile (string aPath)

   It is a method used to create files where the CreateFile method accepts the aPath parameter, which is a string that indicates the name of the file you want to create or access, such as 0:\file_name.txt.

2. DeleteFile(String)

   public static void DeleteFile(string aPath)

   Is a method used to delete files specified by the aPath parameter where the aPath argument will be filled with 0:\(file_name).

3. CreateDirectory(String)

   Public static DirectoryEntry CreateDirectory

   (string aPath)

   Is a method used to create a DirectoryEntry object that represents an existing or new directory in Active Directory Domain Services. DirectoryEntry is a class that provides access to nodes in Active Directory Domain Services, which is a directory service used to store information about objects in the network1. With use DirectoryEntry object, you can create, modify, or delete directories and subdirectories, and read or write properties of those directories.

4. DeleteDirectory(String, Boolean)

   public static void DeleteDirectory (string aPath, bool recursive)

   Is a method used to delete the directory specified by the aPath parameter, along with all its contents if the recursive parameter is true. This method is an overload of the Directory.Delete method, which is a built-in method of the Directory class. The Directory class is a class that provides static methods for creating, moving, and deleting directories.

5. GetFile(String)

   public static DirectoryEntry GetFile (string aPath)

   The GetFile method accepts aPath parameter, which is a string indicating the name of the file you want to access.

6. GetFileStream(String)

   public static Stream GetFileStream(string aPathname)

   This method functions to open a file for reading and returns a Stream object that represents the file. Stream, as an abstract class, provides basic operations for reading and writing data from various sources such as files, memory, and network. By using Stream, users can access data in files without needing to know details about the file system used. The GetFileStream method accepts aPathname parameter to indicate the location of the file you want to open. In the process, the method uses File.OpenRead to create a FileStream object linked to the file, allowing data reading and writing operations. If the file does not exist or is inaccessible, this method will throw an IOException. The Stream object returned can be used for various operations on a file, such as reading the contents, copying to another location, encrypting, and using other methods such as Seek and Close to change the reading or writing position, as well as closing the file and releasing used resources

6. Conclusion

In this research, we succeeded in implementing a simple operating system using Cosmos as the development kernel, the C# programming language, Visual Studio 2022 as a text editor, and VMware as a platform for running and visualizing the operating system. The operating system development process includes various stages from architectural design to implementation and testing. Through this research, we were able to highlight several key points:

- Cosmos as a development framework provides high flexibility in operating system kernel development. The choice of C# language also helps simplify the development process with clean syntax and modern features.
- Visual Studio 2022 is a text editor that simplifies the process of setting up a Cosmos project. Visual Studio 2022's newest features help increase developer productivity.
- Using VMware as a platform to run and visualize operating systems helps in testing and evaluation. The virtual machine configuration steps and operating system integration with VMware make it easy to understand and monitor the performance of the simple operating system under development.
- Testing the functionality and performance of the operating system shows that the implementation successfully achieves the desired goals. Despite the simple nature of this operating system, the test results provide valuable insight into the reliability and performance of the developed kernel.
- This research contributes to understanding the basic concepts of operating system development using modern technology. In addition, this journal can be used as a reference for developers who are interested in starting a simple operating system development project.

  Through this research, we concluded that developing a simple operating system with Cosmos, C#, Visual Studio 2022, and VMware is possible and can provide valuable insights for developers. We emphasize the importance of understanding the basic concepts underlying the development of system operations as a foundation for further research and development in this area.

7. Acknowledgement

The author would like to thank various parties who have provided support and assistance during this research process. We would like to thank the Cosmos development team for creating a very flexible framework and making operating system development easy. Using Visual Studio 2022 as a text editor is a smart move, and we appreciate the new features that have increased efficiency in project setup.

In addition, we would like to thank VMware as a virtualization platform that helps in testing and evaluating our operating systems. Their contribution in providing a stable virtual environment is of great significance to the progress of this research.

Thank you also to all parties who have provided valuable input and suggestions during the development process. Each contribution and support has a positive impact on the successful implementation of the operating system we develop. Finally, we express our highest appreciation to all

parties who have read and provided constructive input on this journal (Clark et al., 2021). All these donations and support mean a lot to the development of this simple operating system. Thank you for your dedication and cooperation.

BIBLIOGRAPHY

[1] G. A., I. M., I. M., & F. A. (2021). DESIGN OF A PURCHASE ACCOUNTING INFORMATION SYSTEM USING MICROSOFT VISUALS. *MEA Scientific Journal (Management, Economics and Accounting*), 1 - 17.

[2] H. S., Syafii, R. S., M. M., & W. Y. (2022). Design visual studio based GUI applications on-grid connected rooftop photovoltaic measurement. *TELKOMNIKA Telecommunication Computing Electronics and Control, 1 - 8*.

[3] Ramdan, S. (2016). THE INFLUENCE OF VMWARE WORKSTATION-BASED LEARNING MEDIA AND LEARNING MOTIVATION ON LEARNING ACHIEVEMENT IN THE SUBJECT OF SERVER ADMINISTRATION CLASS XI COMPUTER NETWORK ENGINEERING IN VOCATIONAL SCHOOLS IN KULON PROGO DISTRICT. FINAL PROJECT THESIS ON ELECTRONICS ENGINEERING EDUCATIONAL STUDY PROGRAM, 1 - 110.

[4] K. M., H. J., & T. N. (2011). An interactive system for structure-based ASCII art creation. An interactive system for structure-based ASCII art creation, 1 - 6.

[5] A Knowledge-based Operating System G. S. Blair, J. A. Mariani, J. R. Nicol, D. Shepherd The Computer Journal, Volume 30, Issue 3, 1987, Pages 193–200,

[6] S. Elvin., A. Dandi., R. Yasir. (2023). OPERATING SYSTEM INSTALLATION TRAINING

COMPUTER WITH VMWARE AT SMK MUHAMMADIYAH 04 MEDAN. Journal Community Service, 24 – 32.

[7] E. C. Rahayu. (2021). UNDERSTANDING THE STRUCTURE OF THE OPERATING SYSTEM AND THE VARIOUS OPERATING SYSTEM STRUCTURES THAT HAVE BEEN TRIED. Jurnal Fakultas Komputer, 1-8