Hi, my name is Mustafa Sibai, and today I'll be presenting my research proposal. Presentation entitled Developing an Operating System from Scratch for Educational Purposes.

Creating an operating system from scratch is a complex and challenging undertaking, but it offers invaluable insights into the core principles of computer systems and software development. This research proposal outlines a self-directed learning project to develop a basic operating system and create a tutorial series to be uploaded on YouTube. Software today is slow, bloated, inefficient and full of bugs. This can be attributed to many reasons, one of which is the programmer's lack of deep understanding on how computers really work.

Many programmers today graduate from college without a deep understanding of how the computers really work, what the CPU is doing, and what the RAM is touring. This is a really big problem because in order for programmers to create a fast, efficient software, they need to understand what the computer is doing. They need to understand what the hardware is doing. They need to understand what the CPU is calculating, what the RAM is storing and how the CPU and RAM are communicating between each other.What are the registers touring? How many registers are there? Why is this piece of code slower than this piece of code? And how can you optimize your software to run on a variety of different hardware types? This lack of understanding on what the hardware is really doing on a low level can be attributed to many reasons. One programmer are just not really interested in learning how the hardware works, what it's doing underneath. And two, they don't have the time or energy to learn complex hardware things. Three, they can't really find any resources that can aid them in learning what the hardware is doing. Those resources could be scattered all over the Internet. They could be outdated or they just could be plain wrong. Four, they may be too busy in their day-to-day job that they do not have the time to learn a new skill or to learn what the computer is doing on a lower level. And finally, five, they could be simply lazy. Whether the answer is 1234 or five, the issue really still remains. There are software engineers out there that are creating complex software that is used day today by hundreds, thousands or even millions of people without really understanding how the computer works.And as I mentioned previously, the software could be really, really slow, buggy or just plain doesn't work. And the biggest problem is we as a society kind of just got used to this. We expect this now. We expect software to be buggy. We expect software to be slow. We expect software developers to release software that unfit to be used day to day. And if it is, it is generally really slow. Users have forgotten how software should act, how fast the computer can execute our programs. Software shouldn't take two, three or four minutes to open. In fact, it shouldn't take 10 seconds. It shouldn't take more than 3 seconds. Our computers today are billions and billions of times faster than what they used to be in 1970 and computers back then took us to the Moon. With all that in mind, the question remains why create an operating system? Well, creating an operating system will give a deep insight on how the hardware works and how a typical operating system works. Behind the scenes. Creating an operating system will give a deep insight into how an operating system manages memory, how an operating system schedules applications and processes, and how drivers are written. When creating an operating system, programmers are programming on bare-bone metal. They have nothing between them and the hardware. So in order to create an operating system, they must understand the hardware.

Now that you understand my reasoning for creating an operating system, my research questions would be what are the fundamental building blocks for developing an operating system from scratch? And how can it be explained effectively? To beginners, what are the programming languages and tools are most suitable for constructing an operating system? How can the video series be organized to facilitate a gradual and interactive learning experience suitable for individuals with different levels of experience? What are the main obstacles a student could face while developing an operating system? How can the video educational content be optimized for clarity and active participation to enhance the learning experience for a diverse audience? What are the root causes of inefficiency within an operating system? And finally, how can the gained knowledge and skills through the operating system development be utilized beyond the project? Now that you understand my research questions, we move on to the aims and objectives of my proposal.This is the list of aims and objectives that I currently have to start off.Develop a functional operating System The primary aim of this project is to create and develop a functional operating system from scratch and understand what goes into building an operating system.Where the building blocks of an operating system?How does the hardware work on a deep level and how can we create a fast and efficient operating system?In order for me to accomplish that, I have to learn the X 86 assembly programming language and the C programming language.Furthermore, I have to demonstrate an understanding of fundamental operating system concepts and techniques.Second, facilitates progressive learning.This project aims to provide a structure and a progressively challenging learning path.The aim is to allow students to understand how an operating system works, how a computer works, and how to build an operating system.A simple operating system but a functional operating system.While the primary goal of this project is to create an operating system, the secondary goal for this project is to be able to teach this content to an audience in both a video form and a written form.These tutorials are catering for both a beginner audience who have just started their programming career and more advanced learners.The way I can accomplish that is by first teaching the basics of the C programming language and the basics of X86 programming language and moving from there.Third, promote self-directed learning the project encourages self-directed learning by providing video content and written content to an audience to learn at their own pace.The project encourages self-directed learning by providing resources in both written form and video form for individuals to explore operating system development, hardware architecture and a deep understanding of what the computer is doing.And four, foster a supportive learning community.Another aim of this project is to build an online learning community around the YouTube channel and around a messaging app such as Discord or Slack.This way, students and participants can engage, ask questions and support each other in their learning.Five and finally, share knowledge and experience.Another aim to the project is to share insights, experience and resources to be able to develop an operating system.The objectives for this project is one develop a bootloader and a kernel.Creating a bootloader and a kernel for an operating system focusing on the core functionalities such as memory management, file system and basic process scheduling is one of the fundamental core systems.An operating system would have.Two learn the C programming language and the X 86 assembly programming language.Those are some of the most important languages to know and deeply understand when building an operating system.Without those languages, you would not be able to create a reasonably fast operating system.Three, structure educational content it is really important to plan, script, record and edit the YouTube tutorial videos that explain the development of an operating system.It is extremely important to have this planning beforehand and make sure it is well-paced and understandable by all level of programmers, such as beginner programmers and experienced programmers alike, and finally collect viewer feedback.It is really important to get feedback from viewers before releasing the videos and from viewers after releasing the videos.This can be done by simply talking to those people if possible, or reading the comment section on YouTube, or asking them to fill a Google form survey, or simply asking them in a chatting software such as Discord or Slack if they have joined that software. Regarding the key literature review for this proposal and this project. Many resources exist on how to create an operating system from scratch. However, a lot of them skip essential knowledge that programmers need before making an operating system. I have linked many of these literature in the references section, which can be found at the end of the presentation and. Regarding the methodology for this project.This is going to bea self directive learning methodology.I will be conducting a systematic search of academic databases such as Google Scholar, Research Gate and Lean Pub. I will be utilizing specific keywords to pinpoint relevant literature on operating system development. Will be incorporating scholarly articles and books that have been published in the last decade.I will be screening publications based on their relevancy and excluding any advanced or specialized operating system topics such as multithreading, encryption and networking.Regarding the review process, I will be critically evaluating select works on fundamental operating system components and their user-friendliness in terms of educational approach.Regarding synthetization and analysis, I will be identifying key elements and recurring patterns in the literature to establish a comprehensive grasp of operating system basics.

Regarding any ethical considerations. It is really important to have transparency and honesty between me and my audience. Avoid presenting myself as an expert regarding any intellectual property.It is important to mention the use of any external copyrighted materials such as code or resources regarding safety and security, emphasize responsible and ethical practices in operating system development, and discourage any kind of hacking or creating malicious code. Regarding any attribution or citation, it is really important to do proper citation for any external resources and regarding any ethical responsibility. Recognize my responsibility as an educator and a content creator.I should be respectful in my videos and conscious about my viewers time. Regarding the artifacts.I'll be creating a video series posted on YouTube alongside a written format for readers to read.And I'll be creating code that can be downloaded and executed by my viewers. Regarding the timeline.I will be spending a week to gather literature and resources.I will be spending three months for research, development and testing one week to create the structure for the YouTube series and one month to create the YouTube video series tutorial and the final month and ten days will be creating the written research report. Finally, here are all the resources and references gathered for this project. Thank you very much.

References:

Alqarni, Shahad & El-Sharawy, Enas & Saeed, Eman & Khamis, Fatimah & Alqahtani, Lama. (2023). Operating System Architecture.

Renê S. Pinto, Pedro Nobile, Edwin Mamani, Lourenço P. Júnior, Helder J.F. Luz, Francisco J. Monaco, Operating System from the Scratch: A Problem-based Learning Approach for the Emerging Demands on OS Development, Procedia Computer Science, Volume 18, 2013, Pages 2472-2481,ISSN 1877-0509,

Michael D. Black. 2009. Build an operating system from scratch: a project for an introductory operating systems course. In Proceedings of the 40th ACM technical symposium on Computer science education (SIGCSE '09). Association for Computing Machinery, New York, NY, USA, 448–452.

Agal, Sanjay. (2023). FUNDAMENTALS OF OPERATING SYSTEMS. 10.5281/zenodo.8420194.

Hilal, Noor & Ali, Zainab & Dhafer, Rawan & El-Sharawy, Enas & Aldoosary, Badriah. (2023). Operating System.

CodePulse. (2020). Write Your Own 64-bit Operating System Kernel #1 - Boot code and multiboot header. [Online Video]. 20 September 2020. Available from: https://www.youtube.com/watch?v=FkrpUaGThTQ. [Accessed: 10 October 2023].

nanobyte. (2019). Building an OS - 1 - Hello world. [Online Video]. 27 January 2019. Available from: https://www.youtube.com/watch?v=9t-SPC7Tczc. [Accessed: 10 October 2023].

Chris Titus Tech. (2021). Build Your Own Operating System. [Online Video]. 29 January 2021. Available from: https://www.youtube.com/watch?v=YbXHU7W7Its. [Accessed: 10 October 2023].

Daedalus Community. (2020). Making an OS (x86) Chapter 1 - CPU, Assembly, Booting. [Online Video]. 23 August 2020. Available from: https://www.youtube.com/watch?v=MwPjvJ9ulSc. [Accessed: 10 October 2023].

Coding With Ashutosh. (2022). OS in Scratch 3.0 || Operating System Part - 1. [Online Video]. 20 March 2022. Available from: https://www.youtube.com/watch?v=ucMV68Gv9WQ. [Accessed: 10 October 2023].

Blundell, N., 2023. OS Dev. Writing a Simple Operating System — from Scratch, [Online]. December 2, 2010, 77. Available at: https://www.cs.bham.ac.uk/~exr/lectures/opsys/10\_11/lectures/os-dev.pdf [Accessed 10 October 2023].

Pesse, S. (2014). How to Make a Computer Operating System from Scratch using C/C++. Retrieved from https://github.com/SamyPesse/how-to-create-an-operating-system