**PROJECT AND TEAM INFORMATION :**

**Project Title – stakOS**

**Student / Team Information –**

|  |  |
| --- | --- |
| Team Info. | Image |
| **Team Member 1 (Team Lead)**  First Name : Komal  Last Name : Karki  Student ID : 22012399  University Roll No : 2218987  Email : [komalkarki167@gmail.com](mailto:komalkarki167@gmail.com) |  |
| **Team Member 2**  First Name : Swati  Last Name : Bhatt  Student ID : 22011621  University Roll No : 2219796  Email : [swattidccd@gmail.com](mailto:swattidccd@gmail.com) |  |
| **Team Member 3**  First Name : Animesh  Last Name : Rawat  Student ID : 220112585  University Roll No : 2218355  Email : [animeshrwt77@gmail.com](mailto:animeshrwt77@gmail.com) |  |
| **Team Member 4**  First Name : Tushar  Last Name : Goel  Student ID : 220112706  University Roll No : 2219833  Email : [tushargoelrke@gmail.com](mailto:tushargoelrke@gmail.com) | A person in a suit  AI-generated content may be incorrect. |

**PROPOSAL DESCRIPTION :**

**Motivation –**

In the modern world where technical dependency is there the need for a basic yet customize OS is need of hour. Building an OS which simply caters the needs of users and provide an interface between the user and the computer hardware, enabling efficient execution of applications is our *Primary Motivation.*

*stakOS* serves in that direction, where it handles all the basic OS functions like - process, memory, file management and gives a platform to newbies to learn and practice freely. The Idea of creating an OS as a part of PBL (Project Based Learning) in Software Engineering(OS) is itself inspiring and gave us chance to learn the basic core concepts of Machines and add Stars in our Resume.

**State of the Art / Current Solution –**

A lot of Open-Source Platforms/Contributions are there providing such facilities and on other hand we have Big Tech companies like Microsoft, Apple dominating this market for decades and help to solve a lot of problems.

However, these solutions are often complex, resource-intensive, and not easily customizable for learning purposes. A lot of students face difficulties in basic core concepts of OS and in basic architecture although lightweight OS's like Minix, TinyOS, Ubuntu exist but they also require some amount of technical knowledge. This is where stakOS aims to fill the gap by offering a minimalistic yet customizable OS that covers fundamental OS functionalities while allowing users, especially students, to experiment and learn OS concepts interactively.

**Project Goals and Milestones –**

StakOS is a 32-bit custom operating system designed from scratch, using a custom bootloader, a basic linux like kernel, and a terminal for executing custom commands. The goal of this project to enhance our understand using low level programming,OS development, and x86 architecture while creating a basic and functional operating system.

**MILESTONES :-**

**1. Tools Setup :-**Tools like gcc, nasm and qemu.

**2. Bootloader :-**making our own custom bootloader

**3. Kernel :-**using a linux like kernel

**4. Terminal :-**terminal to run our custom commands.

**5. File system :-** For managing files and directories.

**Project Approach –**

We are developing this project in linux or unix-like operating system, for ease in development. After setting up our IDE’s we made our custom Bootloader which will tell the bios about our operating system . We are writing this Bootloader in Assembly (ASM) we are using **NASM (Netwide Assembler)** as it works well with C code.

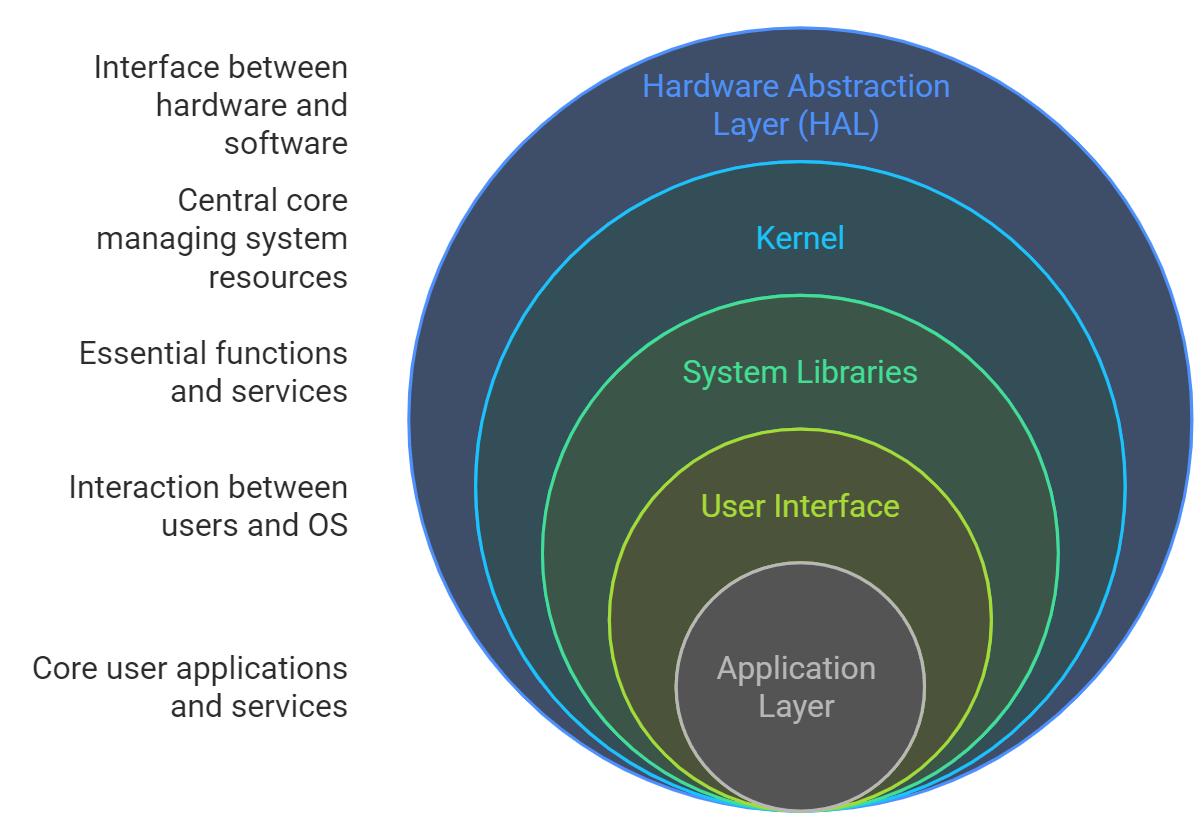
We are using Qemu (Quick Emulator) to test and debug our custom OS, this makes the development of the OS more robust.

Our Bootloader loads the kernel which is a integral part of the OS,we will use a prexisting kernel written in C ,we will customize the kernel according to our needs.

We will make a terminal interface for our custom OS which will allow users to enter basic commands like **echo ,mkdir ,ls** and interact with the system. For that first we will need to enable keyboard inputs so that we can run our commands this will be done using keyboard driver and then we need to parse and execute the commands.After this the final step will be to add a basic file system support **.**

**System Architecture –**

This Portion Outlines the OS architecture and further highlighting its key features and functionalities. The below Architecture gives us an idea about the primary layers and tells the inner working of a bootloader and rough Architecture of an Assembler Diagrammatically.

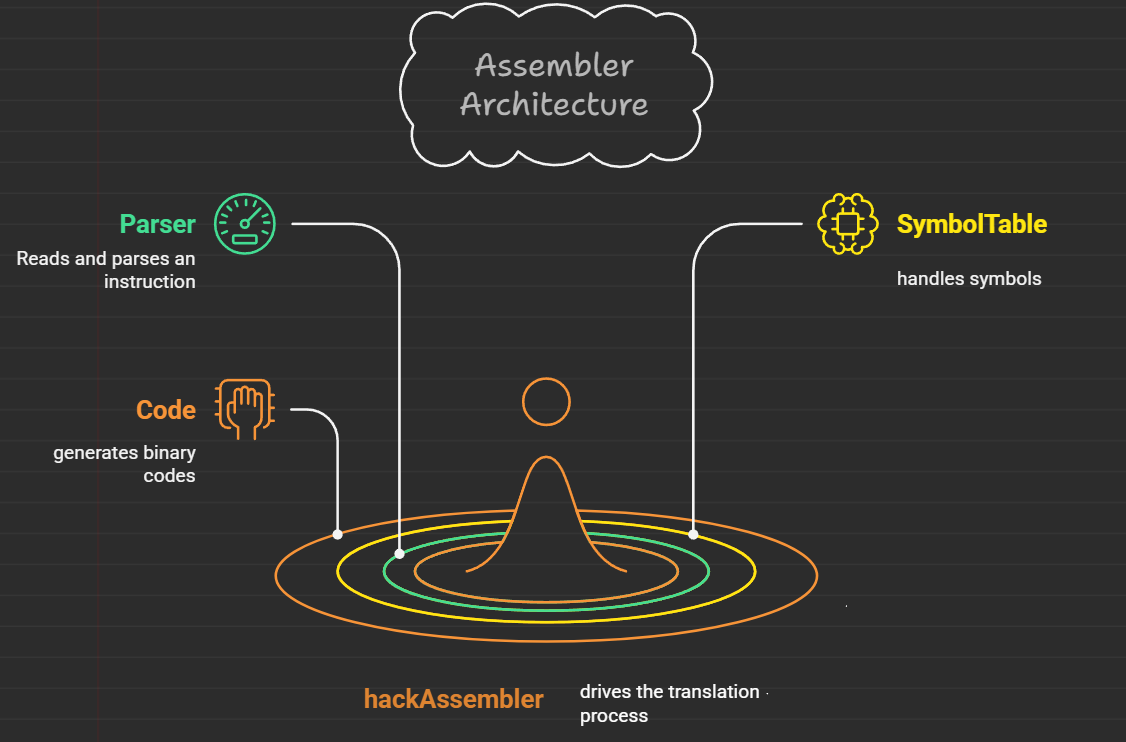


A white sign with red text

AI-generated content may be incorrect.

A diagram of a computer

AI-generated content may be incorrect.



**Project Outcome / Deliverables –**

The Outcome of this Project is that we get a Simple, Lightweight, Easily Bootable Operating System that can work on most of the modern computers. The OS (stakOS) is very flexible and compatible with different Architecture machines.

In the process of making this project we will properly document each and every step required to make a Robust System which will hopefully help others to understand low level programming and system Architecture.

**Assumptions –**

Some prerequisites we expect from the user are:

1. 32-bit x86 architecture

2. BIOS based setup

3. limited built-in commands

**References –**

1. operating systems from 0 to 1 by tu do hoang.

2. <https://interrupt.memfault.com/blog/how-to-write-a-bootloader-from-scratch>

3. <https://developer.arm.com/documentation/ka002218/latest/>

4. <https://www.linuxfromscratch.org/>

5. <https://www.minix3.org/>

-----------------------------------------------THE END----------------------------------------------------------