TheHackersBrain OS (THBOS)

A minimal x86 operating system written in C, ASM from Scratch

Gaurav Raj

Email: me@thehackersbrain.xyz

Website: https://thehackersbrain.xyz

Red Team OP & Software Engineer

Patna, India, November 2025

TheHackersBrain OS (THBOS)

A minimal x86 operating system written in C,
ASM from Scratch

Gaurav Raj

Email: me@thehackersbrain.xyz

Website: https://thehackersbrain.xyz

Red Team OP & Software Engineer

Patna, India, November 2025

Preparation of this work was facilitated by the use of the *Thesis-template* template.

Acknowledgements

Writing Guidance

In the *Acknowledgment* section, express your gratitude to those who helped and supported your work. Start by thanking your advisors, mentors, or supervisors who provided guidance and expertise. Mention any colleagues, classmates, or team members who contributed to discussions or offered assistance. You can also acknowledge specific organisations, institutions, or funding sources that supported your research or work. Lastly, include any personal acknowledgments for family or friends who offered encouragement and moral support during the project. Keep this section sincere, concise, and professional.

Abstract

TheHackersBrain OS (THBOS) is a minimal x86 *operating system* kernel built from scratch in C and x86 Assembly, designed to provide direct, unmediated access to CPU-level instructions and hardware state inspection. Unlike traditional minimal OS projects that prioritize shell functionality or device driver frameworks, THBOS focuses on CPU introspection through the CPUID instruction, VGA text buffer manipulation at `0xB8000`, and transparent boot-to-kernel transitions via GRUB's Multiboot specification.

But at the core, it's just some dude sharing the knowledge back to the community, from where he learned all that. The kernel operates in protected mode with a 16KB stack allocation, directly mapping `C` functions to hardware memory operations without abstraction layers. This paper deconstructs THBOS's boot sequence, memory layout via custom linker scripts, and CPUID-based vendor/model extraction, while documenting critical design decisions including GCC optimization-induced Multiboot header displacement and stack alignment constraints.

Why you ask?, it's almost a base for low-level development in software engineering, binary exploitation in hacking, and just a curiosity to learn and design and understand the thing we can't live without (though as a dev or hacker or any tech person, although now even a normal person). I present a structured roadmap for extending minimal kernels with Interrupt Descriptor Tables (IDT), Programmable Interrupt Controllers (PIC), and keyboard input handlers. THBOS serves as both an educational platform for understanding x86 architecture fundamentals and a foundation for low-level system experimentation, security research, and exploit development environments.

Keywords: Operating System, Kernel, BootLoader, CPU, C, Assembly, OS Hacking, etc.

Contents

List of Figures

List of Tables

1

Introduction to the Template Motivation and Steps

Author: Gaurav Raj

Official Repository: GitHub Repository

Welcome to the *Thesis Template* template! Thank you for choosing it for your dissertation, report, or project. This template reflects many hours of development, and I hope you enjoy using it as much as I did creating it. This chapter introduces its purpose and helps you get started. See **??** for a detailed guide, and **??** for a brief LaTeX tutorial to maximise its use.

1.1 Motivation

Modern operating systems abstract hardware complexity through layers of device drivers, system calls, and memory management frameworks. While this abstraction enables application portability and developer productivity, it obsecures the fundamental hardware-software interface that defines computing. Understanding this interface – how instructions execute, how memory is addressed, how peripherals communicate with the CPU–remains essential for systems programmers, security researchers, and anyone seeking to master computer architecture at its lowest level.

Building an operating system from scratch forces direct engagement with these fundamentals. There are no libraries to mask segmentation faults, no kernel panic handlers to catch undefined behavior, and no scheduler to hide the cost of context switches. Every function call maps to assembly instructions, every pointer dereference translates to physical memory access, and every hardware interaction requires manual register manipulation.

THBOS (*TheHackersBrain Operating System*) was conceived as a minimal x86 kernel that prioritizes transparency over functionality. Rather than implementing a full-featured environment, THBOS focuses on direct CPU interaction through the CPUID instruction, provid-

ing real-time hardware introspection while maintaining a codebase small enough to audit in a single sitting. The entire kernel—boot loader, VGA driver, and CPUID wrapper—comprises fewer than 500 lines of combined C and Assembly.

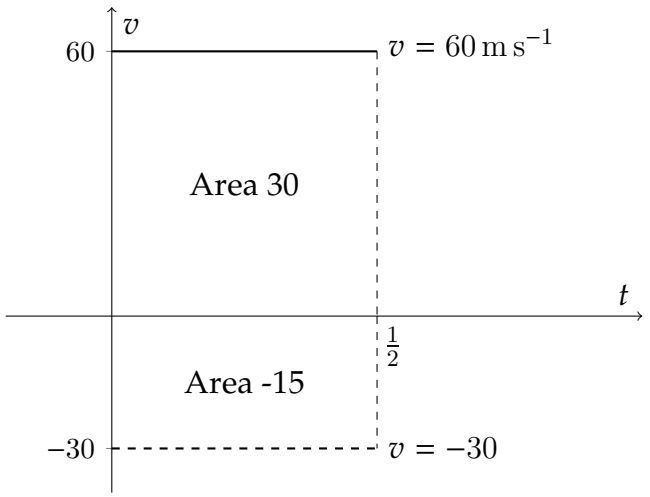
This is a complete expermentation project while learning and keeping things practical for everyone.

1.2 Another Important Note

Okay, My emacs setup is finally working and is awesome.

1.3 Maths Testing

This is the section to plot maths graphs.



1.3.1 Function Plots

You think trigonometry is only for navigaters. Think again. Some of the best use cases of trigonometry is in ***rotation***, ***vibration***, and ***Oscillation***. Basically in repeating patterns or motions.

Here's a simple ***Sinusoidal Oscillation***.

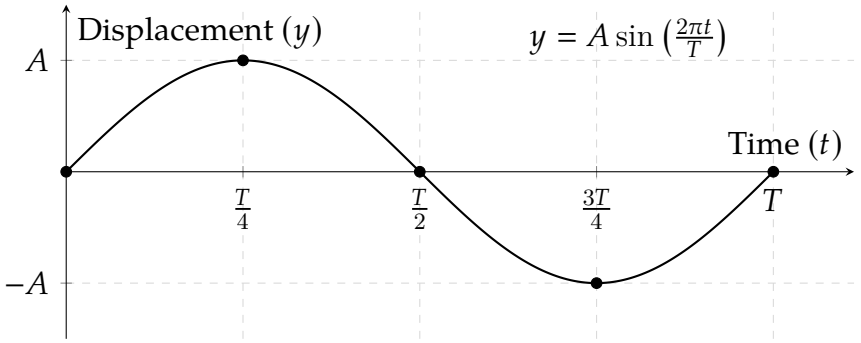


Figure 1.1: *Sinusoidal oscillation showing one complete period T with amplitude A.*

1.3.2 Formulas

Now Formulas, a basic **Quadratic Equation** cheatcode formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

1.3.3 Theorems

Theorem 1. *For all* $a, b \in \mathbb{R}$, $(a + b)^2 = a^2 + 2ab + b^2$.

2
Logrithms

2.1 What is Logrithms?

A logarithm is the **inverse operation of exponentiation**.

If you know that:

$$a^2 = c$$

then the logarithm answers the reverse question:

$$\log_a(c) = b$$

It basically asks: *"To what power must I raise a to get c?"*

2.1.1 Basic Examples

- $2^2 = 8 \implies \log_2(8) = 3$
- $10^4 = 10000 \implies \log_{10}(10000) = 4$
- $e^2 \approx 7.389 \implies \ln(7.389) = 2$
- $5^0 = 1 \implies \log_5(1) = 0$
- $7^1 = 7 \implies \log_7(7) = 1$
- $9^{1/2} = 3 \implies \log_9(3) = \frac{1}{2}$
- $27^{1/3} = 3 \implies \log_{27}(3) = \frac{1}{3}$

2.1.2 Common Logarithm Bases

- **Base 10 (common logarithm)**: $\log_{10}(x)$. Used often in real-world scales like decibels, pH, Richter scale.
- **Base e (natural logarithm, ln)**: $log_e(x)$. Super important in calculus, growth/decay, probability, finance, etc.
- **Base 2 (binary logarithm)**: $log_2(x)$. Used in computer science (bits, complexity, etc.).

2.1.3 Logarithms Rules (Laws of Logs)

These come from exponent rules:

1. **Product Rule**: $\log_a(MN) = \log_a(M) + \log_a(N)$
2. **Quotient Rule**: $\log_a(M/N) = \log_a(M) - \log_a(N)$
3. **Power Rule**: $\log_a(M^k) = k \cdot \log_a(M)$
4. **Change of Base**: $\log_a(b) = \dfrac{\log_c(b)}{\log_c(a)}$ (lets you convert between bases).

2.1.4 Why are Logs Useful?

- **Compressing big numbers**: Turns multiplication into addition (handy in computation).

- **Scales**: Earthquakes, sounds, acidity (pH), star brightness, all useful logarithmic scales.
- **Mathematics**: Solving equations like $3^x = 81$. Instead of guessing, you use logs.
- **Computer Science**: Algorithms often run in $O(\log n)$ time, meaning they're super efficient.

Example: A novel signature scheme is introduced, along with an implementation of the Diffie-Hellman key distribution scheme that accomplishes a public key cryptosystem (Elgamal, 1985). According to Elgamal (1985), a new signature scheme that accomplishes a public key cryptosystem is introduced (...) This template was created by Gaurav Raj, with the title *TheHackersBrain Operating System from Scratch*.

Bibliography

Elgamal, T. (July 1985). "A public key cryptosystem and a signature scheme based on discrete logarithms". In: *IEEE Transactions on Information Theory* 31.4, pp. 469–472. ISSN: 1557-9654. DOI: 10.1109/TIT.1985.1057074. URL: https://ieeexplore.ieee.org/document/1057074.

Gaurav Raj (Dec. 2023). *TheHackersBrain Operating System from Scratch*. URL: https://github.com/thehackersbrain/thbos.

Appendices

A

Showcasing the First Appendix

Writing Guidance

Appendices contain supplementary material **created by the author** that enhances the reader's understanding of the dissertation while not being essential for following the primary narrative. These sections often include detailed tables, figures, complex calculations, or materials like survey questions and interview transcripts produced in the course of the research. The appendices allow readers to explore the research in greater detail, offering a deeper insight into methods and findings without interrupting the main body of work.

Annexes

L
Showcasing the First Annex

Writing Guidance

Annexes are supplementary sections in a dissertation that provide additional information or external documents not essential to the main arguments but that support or complement the research. Unlike appendices, **annexes generally contain material that was not developed by the author**, such as reports, legal documents, or published datasets from external sources. This information is placed separately to keep the main content concise, allowing readers access to relevant external references without disrupting the dissertation's flow.