Tanish Mohanta

Mobile: +91 790 310 9365
Email: tanishmohanta1901@gmail.com
GitHub: https://github.com/tanishtt
LinkedIn: https://www.linkedin.com/in/tanish-mohanta-09b07b1b2/

Aspiring software engineer driven by a passion for solving complex problems through creative and efficient coding. Committed to continuous learning, I eagerly embrace new technologies, ensuring I look forward to advancements and contribute innovative solutions to the dynamic world of software development.

EDUCATION

MIT-World Peace University — **Pune, Maharashtra, India**
Bachelor of Technology in Computer Science and Engineering [**Aggregate: 9.42 CGPA**] — June, 2024

DAV Public School, Jamshedpur — **Jamshedpur, Jharkhand, India**
Class 12th [**Aggregate: 94%]** — April, 2020

DAV Public School, Chaibasa — **Chaibasa, Jharkhand, India**
Class 10th [**Aggregate: 96%]** — April, 2020

TECHNICAL SKILLS

- C | CPP | ASSEMBLY | HTML | CSS | JAVASCRIPT | SQL
- Data Structures and Algorithms | Compiler Design | Operating System | Object-Oriented Programming | Database Management System

PROJECTS

- **32 BITS - OPERATING SYSTEM**

Utilized : C | ASSEMBLY | MakeFile

- Implemented a custom boot loader in assembly language, which brings kernel from second sector of Hard Disk Drive.
- Implemented I/O to take input and output from I/O Ports.
- Implemented Interrupt Descriptor Table and Interrupt Service Routine(80h) with various system command(print, putchar, getkey, malloc, free etc).
- Implemented classic Keyboard Driver.
- Memory management with paging mechanism and heap and kernel heap implementation(hcreate, hmalloc, hfree).
- Implemented a custom FAT16 filesystem driver, understanding the filesystem's cluster-chaining structure and enabling file creation.
- Implemented kernel main by initializing terminal, kheap, IDT, Task segment, ISR 80h register commands, Paging, Keyboard.
- Tested .elf file by executing and printing the result in terminal.

- **HEAP MEMORY MANAGER**

Utilized : C

- Implemented Virtual Memory Page allocation and deallocation API and Page family registration.
- Implemented Meta blocks and Data blocks and block splitting and merging of freed memory.
- Implemented Free Block Management to manage free blocks.
- Implemented custom Malloc and Free along with internal fragmentation handling.
- Tested by writing a printing function which dumpa the state of HMM.

- **MEMORY LEAK DETECTOR**

Utilized : C | MakeFile

- Implemented Structure Database to store application structure information maintained by MLD library.
- Implemented Memory Leak Detection Algorithm in MLD library.
- Reported the leaked objects by processing the object database.
- Analyzed MLD library limitations.

- **ELF LOADER**

Utilized : C | MakeFile

- Implemented elf_load to read an ELF file, allocate memory, and copy segments.
- Managed virtual and physical addresses in struct elf_file and provided functions for retrieving address boundaries.
- Implemented functions (elf_header, elf_sheader, elf_pheader, elf_program_header, elf_section) for easy access to different headers.
- Implemented elf_phdr_phys_addrss to determine the physical address of a program header.
- Tested by loading .asm file.

- **MUSIC PLAYER**

Utilized : HTML | CSS | JAVASCRIPT

- Dynamically filled song details using JavaScript, updated song list.
- Implemented play, pause, and progress features for easy music playback control.
- Implemented navigation between songs with forward and backward controls.

CERTIFICATIONS

- Data structures and algorithms by Abdul Bari
- Operating System by Vignesh Sekar
- Competitive Programming by Prateek Narang

LINKS

- Leetcode : https://leetcode.com/tanishmohanta1901/
- Codechef : https://www.codechef.com/users/tan_moh9999
- Codeforces : https://codeforces.com/profile/tan_moh99