IITK$~SUMMER_PROJECT_2024/NFS_GOES_ONLINE

Mentors :: Hardik Jindal, Ashish Ahuja, Ritvik Goyal, Shivansh
PClub, IIT Kanpur

/Objective

- Understand the Infrastructure of Computer Networks
- Learn Basics and In-Depth Concepts of Operating Systems and Filesystems
- Build a Rust Server and a Distributed Filesystem Using C

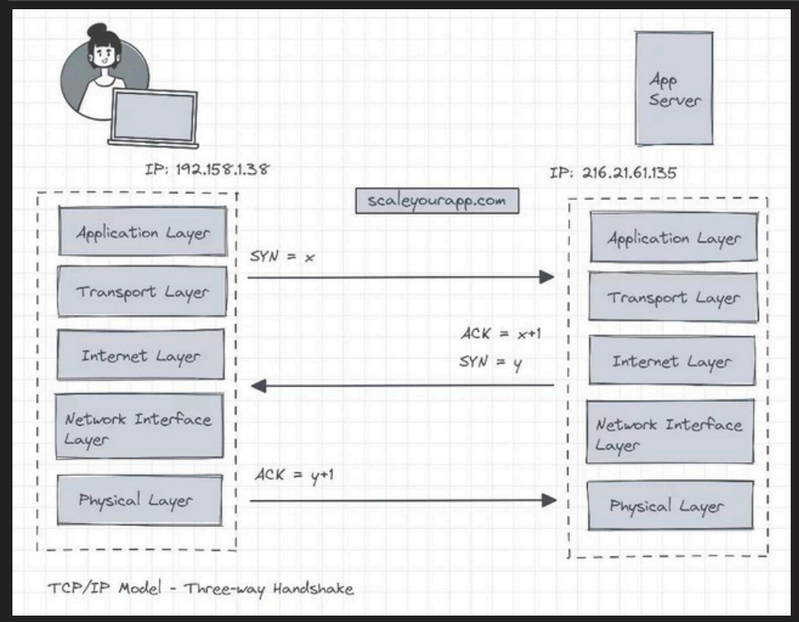
/Rust

- Rust is designed for system-level programming, similar to C and C++, but with modern features and improved memory safety.
- Key features: Basic Syntax, Ownership, Concurrency
- Ownership ensure memory safety in rust.
- We also participated in a CodeForces contest using Rust!

/Implementation/Rust_Server

- Listening for Connections: We used TcpListener to listen for incoming TCP connections on a specified address and port.
- Handling Requests: For each incoming connection, we read the HTTP request, processes it, and sends back an HTTP response.
- Parsing Requests: We read from the TCP stream, parse the request, and generates the required response
- Sending Responses: The server sends an HTTP response back to the client, usually a simple HTML file or a plain text response.

/Networking



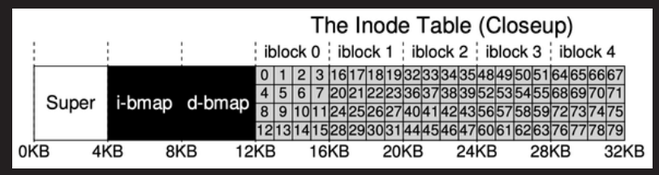
- ***Application, Transport, Network, Data-Link, Physical Layer,*** the message is passed over each layer which addes up a header which is communicated over network.
- TCP and UDP, Tcp provides a realible data trasfer system using SYN ACK approch, Udp is just simple trasfer without reliability, used in less important data transfer.

/Operating_System

- It manages and provides resources on requirement to applications, according to system policies for fair and efficient access to resources to all
- Other important feature is the Virtualization, which gives each process the impression that it alone is actively using the CPU.
- Then we came across how a piece of code in percistent memory comes to life by OS

/File_System

- Provides an abstraction for storage, allowing users to store and retrieve data in a structured way.
- System Calls: fundamental interface between an application and the operating system (OS)
- Inodes: Stores info about file data blocks, and file in directory
- Bitmaps: Takes note of which inodes and data blockes are empty
- Super Block: Holdes the inforamtion about file system, helpfull for virtual file system mouting
- Data Block: holds the contents of file.



/Implementation/NFS

- Distributed File System enables file access and storage across multiple machines.
- Location transparency allows file access without knowing physical locations.
- Caching improves performance by storing frequently accessed data closer to us.
- Replication increases reliability but challenges consistency across copies.
- Journaling ensures data integrity, aids crash recovery, and maintains consistency.
- Strategies like counters and cursor on request helps to handle failures of request, response
- We implemented the functions like read, write, stat , lookup, creat, unlink for the server side

/references

- Rust Book: https://doc.rust-lang.org/book/
- Networking: https://www.hackers-arise.com/networks-basics and https://www.geeksforgeeks.org/tcp-ip-in-computer-networking/
- Operating Systems: Three Easy Pieces : https://pages.cs.wisc.edu/~remzi/OSTEP/

/mentees

Manas, Rishi, Himanshu, Devansh, Aarush, Aryan, Harshit, Nikhil, Pranav, Abhay, Abdul, Arsh, Advaid, Divyansh, Jayant