Operating Systems & Computer Networks Track

Comprehensive Skill Development for System & Network Applications

Agnes, Sapiens AI

Source: 3d Rendering Of Programming Code Flowing Across Computer Screen ...

Program Goal & Structure

Program Goals

Comprehensive Understanding

Gain a deep understanding of OS principles and network fundamentals, forming the bedrock for advanced application development.

OS Principles | Network Fundamentals

Practical Application

Enable the analysis, implementation, and troubleshooting of system-level and network-based applications with hands-on skills.

Analysis | Implementation | Troubleshooting

Capstone Project

Culminate in a distributed client-server application demonstrating concurrency and robust network communication.

Client-Server | Concurrency | Network Communication

Program Structure

2 Months Online
Foundational & Core Concepts

1 Month Offline
Project & Industry Immersion

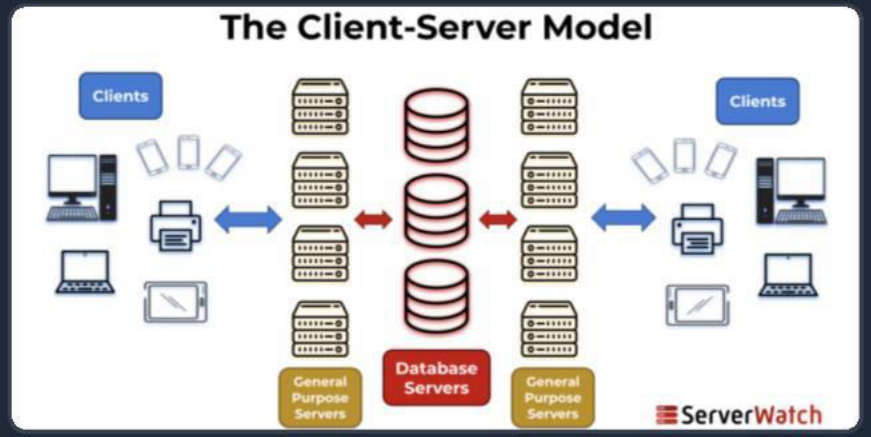


Diagram illustrating a typical client-server model.

Month 1: OS Fundamentals - Processes & Scheduling

Week 1: Foundations of OS & Process Management

- **Operating System Introduction:** Understand OS roles (resource mgmt), types (batch, time-sharing), and structures like monolithic vs. microkernels.
- **Process Concepts:** Define a process, its states (ready, running, waiting), and the role of the Process Control Block (PCB).
- **Inter-Process Communication (IPC):** Learn how processes communicate via pipes (simple channels) and shared memory (direct data exchange).
- **Hands-on Lab:** Explore process states using `ps` and `top`. Implement basic IPC in C/Python to see data exchange in action.

Week 2: Threads & Efficient CPU Utilization

- **Threads:** Contrast lightweight threads with processes. Compare user-level vs. kernel-level threads and multithreading models.
- **CPU Scheduling:** Grasp scheduling goals (throughput, response time) and analyze algorithms like FCFS, SJF, and Round Robin.
- **Hands-on Lab:** Develop multi-threaded programs in C/Java/Python. Simulate scheduling algorithms to compare performance.

Sources: Programming Code Running Computer Screen Terminal Developing Programming, CPU Scheduling in Operating Systems - GeeksforGeeks

Month 1: OS Fundamentals - Memory & File Systems

Week 3: Memory Management (Paging & Segmentation)

Core Concepts

Contiguous Allocation: Simple method where each process gets a single contiguous block of memory. Leads to fragmentation.

Fragmentation:

Internal External

Paging: Solves external fragmentation by dividing logical memory into pages and physical memory into frames. Utilizes a Page Table and fast cache TLB.

Segmentation: Memory is divided into variable-sized segments based on program structure (code, data, stack), supporting a user's view of memory.

Swapping: Processes are temporarily moved from main memory to secondary storage to improve multiprogramming.

Hands-on Lab

Week 4: Virtual Memory & File Systems

Core Concepts

Virtual Memory: Allows programs larger than physical memory using Demand Paging (loading pages only when needed).

Page Replacement Algorithms:

FIFO LRU

Thrashing: A state where the system spends more time swapping pages than executing, causing severe performance degradation.

File System Interface: Defines how the OS interacts with files, including directory structures (tree-structured, acyclic-graph) and access methods (sequential, direct).

Hands-on Lab

Month 2: Network Fundamentals - Models & Data Link

Week 5: Network Models & Physical Foundation

Core Concepts: Network Fundamentals

Components & Topologies: Explore hardware like routers, switches, and hubs. Understand arrangements: Bus, Star, and Ring topologies.

Network Models: Introduction to the OSI and TCP/IP models that standardize network communication.

Physical Layer Aspects: Focus on transmission media (Copper, Fiber, Wireless), signal encoding, and data rate.

Week 6: Data Link Layer & Ethernet Protocol

Core Concepts: Data Link Communication

Data Link Functions: Covers Framing, error detection (CRC), and flow control.

MAC Sublayer & Ethernet: Understand Media Access Control, the CSMA/CD access method, Ethernet standards, and unique hardware MAC addresses.

Month 2: Network Fundamentals - Network & Transport Layers

Week 7: The Network Layer - Interconnecting Networks

Logical Addressing & Routing

IPv4 Understanding the 32-bit addressing system for unique device identification.

CIDR How Classless Inter-Domain Routing enables efficient IP allocation and reduces routing table size.

Subnetting Dividing a large network into smaller, manageable sub-networks for better organization and security.

Routing Concepts of Static (manual) and Dynamic (protocol-driven) routing to select network paths.

ARP How the Address Resolution Protocol maps logical IP addresses to physical MAC addresses.

Hands-on Lab: Network Diagnostics

- Use `ping` and `traceroute` to test connectivity and trace network paths.

Week 8: Transport & Application Layers - End-to-End Services

End-to-End Communication

TCP Connection-oriented protocol providing reliable, ordered data transfer with flow control. Ideal for web, email.

UDP Connectionless, lightweight protocol for fast, low-overhead transmission. Used for streaming, DNS, VoIP.

Ports & Mux Using port numbers for process-to-process communication and multiplexing multiple applications over one network link.

DNS & HTTP/S Application protocols translating domain names and powering the web with secure communication.

Hands-on Lab: Application Analysis

- View active connections and ports with `netstat`.

Phase 2: Project Application & Industry Immersion

This phase marks a pivotal transition into an **intensive offline experience**, designed to consolidate theoretical knowledge with practical application. It's a period of deep engagement, direct mentorship, and collaborative team formation.

- **Applied Learning:** Directly implement concepts from OS and Network fundamentals.
- **Direct Mentorship:** Benefit from expert guidance and personalized feedback.
- **Team Formation:** Collaborate with peers, simulating real industry project environments.

Concurrency & Deadlock Management

Concurrency: Handling multiple tasks seemingly at the same time, crucial for maximizing resource utilization and system responsiveness.

Race Conditions: Occur when multiple threads access shared data and the final result depends on the unpredictable timing of their execution.

Synchronization Mechanisms:

- **Mutexes** Protect critical sections, ensuring one-thread-at-a-time access.
- **Semaphores** Signaling mechanisms to control access to a pool of resources.

Deadlocks: A state where processes are blocked indefinitely, each

Operating System & Network Security

Basic OS Security:

- **Authentication** Verifying user or process identity.
- **ACLs** Controlling access to objects like files and directories.

Network Security Fundamentals:

- **Firewalls** Monitor and control network traffic based on security rules.
- **VPNs** Create secure, encrypted connections over public networks.
- **Encryption** Securing data from unauthorized access.

Month 3: Capstone Project - Development & Refinement

Week 10: Building the Core

Server-Side Logic

- Implement logic with **Socket Programming (TCP)** for reliable connections.
- Handle concurrent clients via **Multi-threading/processing**.
- Ensure efficient file reading/writing on the server and client.

Client-Side Application

- HTTP Develop robust connection, authentication, and file transfer (upload/download) capabilities.

Hands-on Focus

- Writing core client/server code for robust connections and foundational file transfer logic.

Week 11: Enhancing Reliability

Robustness & Security

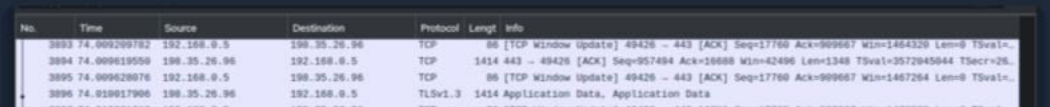
- Implement comprehensive error handling for network and file issues.
- Add basic security features like user authentication to protect data.
- Discuss and apply strategies for performance optimization.

Advanced Debugging

- Utilize advanced tools like **gdb** and **Wireshark** to diagnose complex concurrency and network problems.

Hands-on Focus

- Refining project code, conducting rigorous testing for edge cases, and optimizing for stability and performance.



Month 3: Project Showcase & Career Launchpad

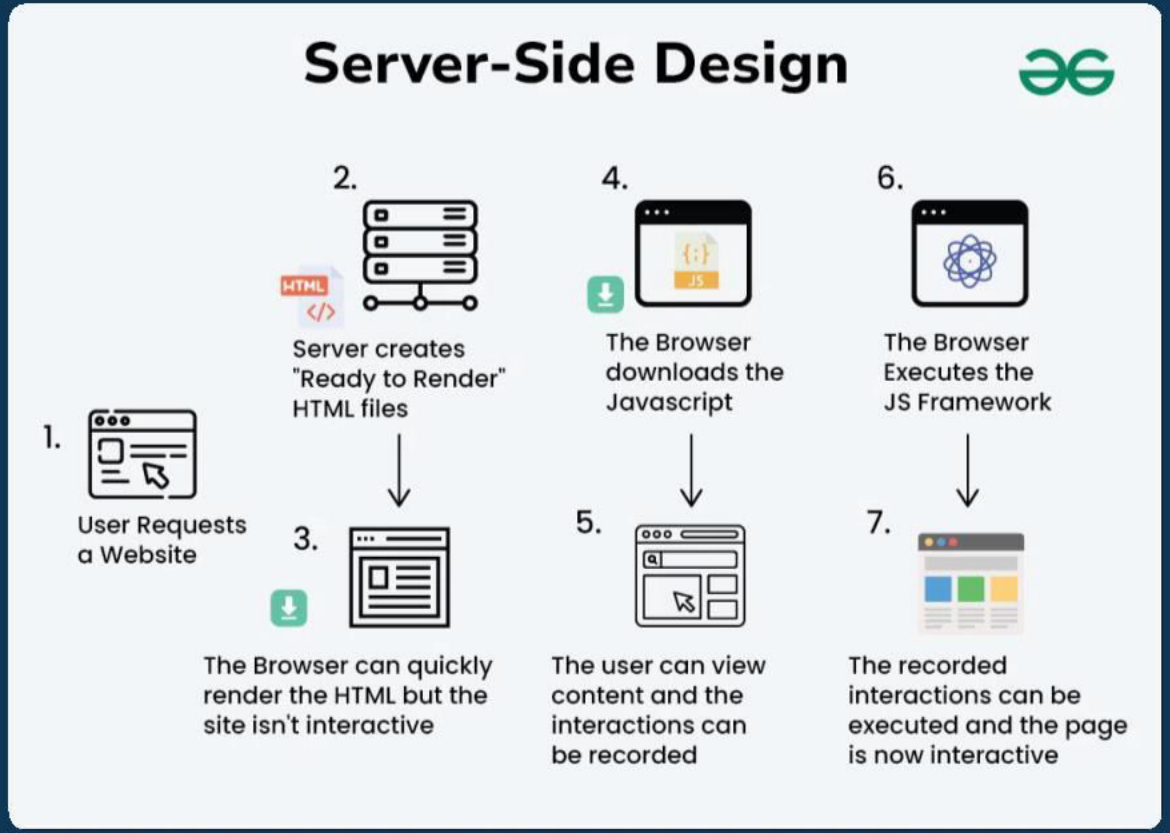
Week 12: Project Showcase: Distributed File Transfer Application

- **Live Demonstration:** Showcasing the functional Distributed Client-Server Application with multi-client support.
- **Applied OS/Network Concepts:**
  Discussion on practical implementation of core principles.

  Concurrency Management of multiple client requests simultaneously.

  OS Principles Process/thread management and efficient file I/O.

  Networking Robust socket programming and TCP data flow control.
- **Design & Future Outlook:**
  Covering design choices, challenges overcome, and future enhancements.



Career Development Launchpad

- **Resume & Portfolio Building** tailored for tech roles.
- **LinkedIn Optimization** & professional networking strategies.

Graduation & Certification

- **Graduation Ceremony** celebrating successful program completion.

Program Summary & Contact

Key Program Takeaways

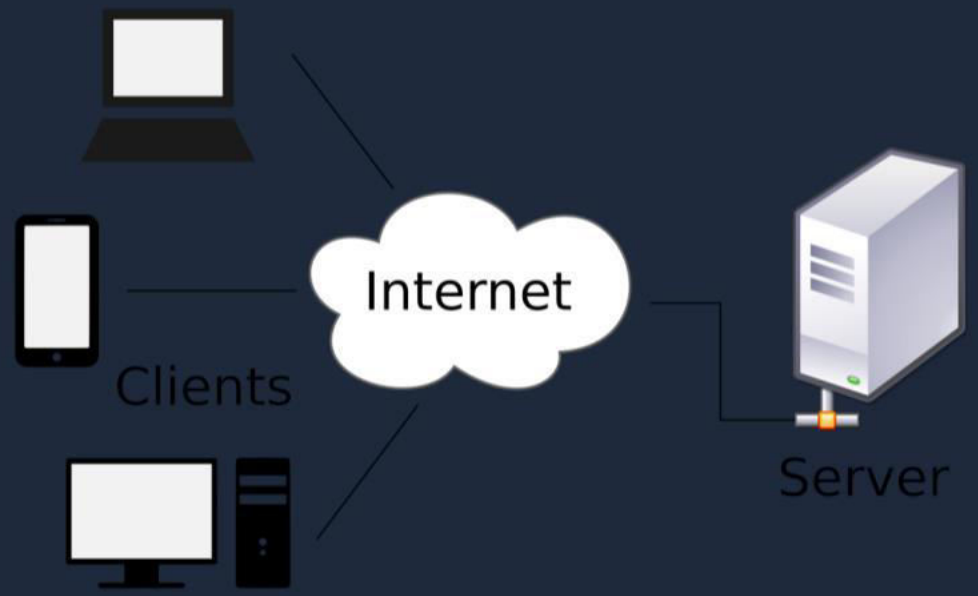
Deep Foundational Understanding
Graduates possess a profound comprehension of OS principles (process/memory management) and a comprehensive grasp of network models and protocols (TCP/IP).

Practical Proficiency
Extensive hands-on experience with system-level programming, IPC, socket programming, and building robust networked applications.

Accelerated Career Readiness
Equipped with a versatile skillset for roles in software/network engineering and systems programming, demanding deep technical insight.

Capstone Project Highlight

The cornerstone of the program is the development of a fully functional **Distributed Client-Server Application**, serving as a practical demonstration of all integrated knowledge.



✓ **Architectural Design:** Implement complex distributed system architectures.