Divyansh Chhabria

Senior Undergraduate, Computer Science and Engineering, IIT Kanpur

@ chhabriadivyansh@gmail.com ☎ +91 9399258709 divc13 divyanshchhabria divc13.github.io/divc/

EDUCATION

Indian Institute of Technology Kanpur
B.Tech, CSE — *CPI: 9.6/10*
2021 - Present — Kanpur, India

Vindhyachal Academy, Dewas (M.P.)
XII, CBSE — *Percentage: 95.8%*
2021 — Dewas, (M.P.)

Vindhyachal Academy, Dewas (M.P.)
X, CBSE — *Percentage: 97.4%*
2019 — Dewas, (M.P.)

TECHNICAL COMPETITIONS

ISC Student Cluster Competition 2024 (HPC)

- Ranked **8th globally** as part of 8-member IITK team
- Profiled RegCM using **IPM**, **TAU**, and **Intel VTune**
- Identified ideal **ppn** for fastest run via **strong scaling**
- Optimized code by restructuring and using **SIMD directives**, reducing runtime from 6330s to 5143s

The Logosphere Hackathon 2023

- Won **1st place** with $5000 in a 2-membered team
- Created and refactored Next.js app to **Web3 DApp**
- Integrated **Blockfrost**, **Cardano**, **Pinata (IPFS)**, **Postgres**, **Fluree**, and **GraphQL** using Logosphere's API

India Terminal 2023, Citadel | Citadel Securities

- Won **1st place** with $7500 in a 2-membered team
- Developed algorithms for tower defense strategy game & competed in single-elimination tournament

SCHOLASTIC ACHIEVEMENTS

- Awarded **Academic Excellence Award** twice for academic performance by IIT Kanpur (2022, 2023)
- Achieved **SPI** of **perfect 10** in the sixth semester
- Secured **AIR 548 (CRL)** in JEE Advanced 2021
- Qualified **KVPY** in **SA** (2020) & **SX** (2021) streams
- Awarded status of **NTSE Scholar** (2019) by NCERT
- Recognized as Madhya Pradesh **State Topper** in **NSEC** 2021 and **NSEP** 2021 by HBCSE
- Qualified **Regional Mathematics Olympiad** 2019

COURSEWORK

* : ongoing

Programming for Performance* | Principles of Database Systems* | Linux Kernel Programming | Compiler Design | Operating Systems | Computer Networks | Parallel Computing | Computer Organization | Advanced Algorithms | Data Structures & Algorithms | Probablity | Software Development & Operations | Intro. to Machine Learning

WORK EXPERIENCE

Systems Engineer

Quadeye Securities LLP — *May 2024 - July 2024*

- Used **gcov** tool to identify code coverage for each unit test in a C++ codebase
- Utilized **libclang** to generate **ASTs** for source files, statically analyzing them to build efficiently updatable **include graph**, **potential call graph**, & **reference map**
- Designed a **tree-diff algorithm** to semantically identify changes in the codebase
- Used a cyclic approach of **tracing metadata** by running unit tests, **updating ASTs** to last commit, & **ranking tests** based on code changes since **last commit**
- Used **CMakeFile API** to identify translation units that would be linked together
- Identified the **minimum translation units** required for compilation after code changes on a **fresh build** to detect potential compiler or linking errors

Visual Mathematics Content Developer

Vizuara Technologies Pvt. Ltd. — *Oct 2022 - Nov 2022*

- Utilized **Manim** library of Python to animate mathematical concepts
- Created **30+ animated videos** on Trigonometry and Surface Areas & Volumes

KEY PROJECTS

Full Fork

Course Project | CS614 | Prof. Debadatta Mishra — *Jan'24-Apr'24*

- Implemented new system call in **linux kernel** to **clone multi-threaded processes**
- Modified kernel to stop all threads except the leader when it sends **SIGSTOP**
- Ensured the last stopping thread directly **notifies the leader**, not to the parent
- Cloned the leader and entered its context by hooking into **schedule_tail**, and created a thread group similar to original by calling **kernel_clone** repeatedly
- Copied execution states of original threads to new threads and resumed them

Parsel Tongue

Course Project | CS335 | Prof. Swarnendu Biswas — *Jan'24-Apr'24*

- Developed compiler for a statically typed subset **Python** targeting **x86_64** code
- Used **Flex** for lexical analysis, and **Bison** for syntactic anlysis, generating **AST**
- Implemented **symbol table**, **register allocation**, **3AC** and **x86 code generation**
- Supported **classes**, **multilevel inheritence**, **function overloading** and **recursion**

Functionalities of gemOS

Course Project | CS330 | Prof. Debadatta Mishra — *Aug'23-Nov'23*

- Implemented **du** utility and dynamic memory functions **memalloc** & **memfree**
- Implemented **trace buffer**, **strace**, and **ftrace** for dynamic trace information
- Implemented **mmap**, **munmap** and **mprotect**, and added **lazy allocation support**
- Implemented **cfork** system call with **copy-on-write** policy & **CoW fault handler**

CSE Bubble

Course Project | CS220 | Prof. Urbi Chatterjee — *Mar'23-Apr'23*

- Built processor with MIPS-like ISA with **single-cycle fetch**, **decode** & **execute**
- Implemented **ALU** using a top-down approach for **R-**, **I-**, and **J-type** instructions
- Designed **finite state machine** for the **control signals** to execute the processor

Unified Portal for Hall Automation

Course Project | CS253 | Prof. Indranil Saha — *Jan'23-Apr'23*

- Developed a software for digitalizing mess, canteen, and booking services
- Adhered to **waterfall model**, while documenting all stages including **requirement specifications, design, implementation, testing,** and **user manual**
- Used **Django Framework** for backend development, **Django-Test** for unit-testing, **Selenium** for integration-testing attaining over **90%** test coverage