Nandha Chokkan

(510) 574 6296 • nandhac18@gmail.com • linkedin.com/in/nandhac

EDUCATION

Virginia Tech | Bachelor of Science in Computer Science *Expected May 2023*

- In-major GPA: 3.34/4.0
- Relevant Coursework: Software Des. & Data Structures, Intro to Computer Organization I/II, Data Structures and Algorithms, Computer Systems, Human-Computer Interaction

SKILLS

C, C++, Java, Python, HTML/CSS | Node.js, React.js, RESTful APIs | CI/CD | AWS EC2, IAM, S3 | Unix | Docker | Git, Gitlab

PROJECTS

Electronic theses and dissertations (ETDs) Chapter Summarization *January 2023 - Current*

- Working with graduate research assistant to create a webpage that accepts chapter text from ETDs and generates summaries, with the goal of helping readers identify portions of interest in long documents
- Using the Pytorch and pysummarization libraries as well as existing language models to create a summarization framework that can used by future students, with an easy-to-use front end

Personal Web/Video Server - CS 3214 *December 2021*

- Built a personal web server with React.js on a Docker container which can serve files, stream MP4s
- Implemented token-based authentication API using JWT, HMAC
- Supported multiple clients simultaneously (single-process, threaded approach)
- Protocol independent: IPv4 and IPv6 clients supported with two separate sockets
- Robust: survives AFL++ fuzzing for 1 hour without crashes or hangs

High Performance Storage Allocator - CS 3214 *November 2021*

- Designed a dynamic storage allocator for C programs; custom implementation of malloc(), free(), and realloc() routines; 89/100 performance index
- Built around testing scenarios: high throughput (30-50 Kops)
- Uses deferred block coalescing, segregated free lists increasing in powers of 2

Fork-Join based Threadpool - CS 3214 *October 2021*

- Implemented a fork-join threadpool in C with work stealing for better load balancing and lower synchronization requirements
- Used critical sections (locks) and condition variables to signal the arrival of a new "future" - task to be completed by threadpool
- Distinguishes external and internal task submissions (subtask submitted by a thread)

Customizable Shell - CS 3214 *September 2021*

- Implemented a simple job control shell with support for: foreground/background processes, pipes/IO redirection, exclusive shell access
- Additional features: process groups, signal handling, built in commands (kill, jobs, stop, fg, bg)
- Customizable prompt support and history command

EXTRACURRICULAR

- *Member*, VT AI/ML