CS5204 Concurrency Web-server

Name: Felipe Campoverde
Date: 12/04/2025

Task 1: Background Study [15 points]

Concurrency: in an operating system refers to the ability to execute multiple tasks in overlapping time periods, often across multiple CPUs or interleaved on a single CPU. This improves performance, but it also introduces challenges such as race conditions, deadlocks, and inconsistent shared state when threads access resources without proper synchronization. In a web server, concurrency is crucial because each client request needs to be handheld independently; without threads, the server would block on long operations and become unusable under load. However, concurrent web servers must be carefully designed so that threads do not corrupt shared data (such as request queues) or starve one another. Preventing these problems requires disciplined use of synchronization primitives mutexes, condition variables, and well-structured producer - consumer patterns, as well as scheduling policies that guarantee fairness. When developing a concurrent web server, it is important to design a safe work distribution mechanism, avoid blocking the main accept loop, control access to shared buffers, and ensure that long-running requests do not starve smaller ones. These principles allow the server to remain correct, efficient, and robust even under high parallel load.