Tentative Schedule and Module Assignments

Module Assignments and Responsibilities

Zafeer Mahmood

Modules Assigned: 1. Core Browser Engine, 3. Networking and HTTP Loader

Steps:

- - Design the CLI for entering commands (open, exit, history, etc.).
- - Implement main control loop to parse user input and invoke appropriate actions.
- - Handle process creation and inter-process communication for new tabs.
- - Implement raw socket connection to retrieve HTML text from a URL.
- - Send retrieved HTML to stdout and handle basic errors (e.g., 404).
- - Integrate networking with tab creation logic.

Kumail Ali

Modules Assigned: 2. Tab Management (Multi-Tab System)

Steps:

- - Set up forking logic to open each tab as a new process.
- - Manage communication via pipes between main browser and tab processes.
- - Implement signal-based control (SIGSTOP, SIGCONT) for scheduling simulation.
- - Log the lifecycle of each tab (creation, suspension, termination).

Okasha Shoaib

Modules Assigned: 5. Cache Management, 6. Download Manager

Steps:

- - Set up shared memory region to hold cached web pages (raw HTML).
- - Implement semaphore synchronization for safe access to the cache.
- - Implement a basic lookup to detect cache hits and store new entries.
- - Build a producer-consumer queue to simulate downloading tasks.
- - Allow multiple resources to be handled in parallel by consumer threads.

Saad Qazi

Modules Assigned: 6. Download Manager, 7. History Management

Steps:

- - Collaborate with Okasha on download manager threading and logging.
- - Implement file-based or in-memory history tracker.
- - Use reader-writer locks to ensure safe concurrent access.
- - Provide CLI support to display browsing history.

Tentative Weekly Schedule

Week 1: Setup GitHub repo, plan module interfaces, research relevant system calls.

Week 2: Zafeer: CLI and controller loop; Kumail: Basic tab fork with pipe setup.

Week 3: Zafeer: Raw socket HTTP request; Kumail: IPC integration; Saad & Okasha: Start Download Manager.

Week 4: Okasha: Cache manager in shared memory; Saad: History logging with reader-writer locks.

Week 5: Connect all components (tabs + networking + cache); simulate browser usage.

Week 6: Testing, debugging, and final documentation; handle deadlocks and edge cases.