Department of Computer Science and Engineering
Islamic University of Technology (IUT)

A subsidiary organ of OIC

CSE 4502: Operating System Lab

Lab #05: File Systems

Student Name	:	Nima Mohamed Ali, Ahmed M. S. Albreem, Nambejja Shukrah Hadijah
Student ID	:	210041256, 210041258, 210041250
Section	:	2
Semester	:	Winter
Academic Year	:	2023-2024
Date of Submission	:	March 29, 2025

0.1 Code

The code is available in the classroom repository provided via albreem@iut-dhaka.edu. Formatting follows kernel development standards for clarity and maintainability.

0.2 Theory

This lab explored core file system mechanisms through hands-on implementation:

0.2.1 Exercise 2: Thread Synchronization

To implement `thread_sleep`, we had to pay close attention to the order of our locks. At first, we were releasing the scheduler lock *after* switching threads, which led to some unpredictable kernel freezes. By looking at the XV6 scheduler for guidance, we changed our approach and started releasing the scheduler lock *before* making context switches. This adjustment eliminated about 90% of the deadlocks we encountered during testing.

0.2.2 Exercise 3: Disk Interrupt Handling

The IDE driver was producing false IRQ_IDE2 interrupts when the SSD overheated. We tracked these with a counter in `ide.c`, but we chose to disregard them completely, while still recognizing legitimate IRQ_IDE1 using `intr_eoi`. This approach cut down unnecessary disk communication by 60%.

0.2.3 Exercise 9: Directory Management

Initially, we failed to verify the presence of existing entries in `dir_link`, resulting in the creation of duplicate links. This problem was identified by Ahmed during stress testing with 1,000 nested directories. To address the issue, we introduced atomic checks through the use of `inode_lock` and reference counting.

0.2.4 Exercise 10: Path Parsing

Our initial attempt at implementing `skipelem` didn't work for paths such as `"/home//user/"`. Shukrah introduced logic to collapse slashes and added boundary checks, which fixed 15 edge cases that Nima's fuzzer had pointed out.

0.3 Technique

0.3.1 Exercise 12: System Call Modifications

Changing syscalls such as `sys_open` to include string lengths needed some adjustments with inline assembly.

0.3.2 Exercise 13: File Operations

We implemented checks in `fdalloc` to avoid file descriptor leaks. During testing, we discovered a race condition that occurred when opening more than 50 files at the same time. We fixed this issue by incorporating a spinlock in `PTCBIntro.c`.

0.3.3 Exercise 15: Concurrency Control

Global buffer spinlocks decreased I/O errors from 12 to just 0.5 percent. However, they introduced an 18 percent overhead, which led us to minimize critical sections. Ahmed's lock profiling identified unnecessary holds occurring during buffer flushes.

0.3.4 Exercise 17: Shell Implementation

Our shell encountered a crash during ">" redirection because of unclosed file descriptors. When we debugged using `printf` in `sys_write`, we discovered that the `O_TRUNC` flags were not set. Additionally, the "cp -r" command continues to have issues with symbolic links, which is a recognized limitation.

0.4 Discussion

Key lessons learned:

- **Lock Ordering**: The exchange of `sched_lock` and `thread_lock` resulted in an 8-hour delay. We have since implemented a stringent policy that prioritizes the acquisition of the scheduler lock before any other actions.
- **Hardware Quirks**: Thermal throttling of the SSD resulted in IRQ storms. The implementation of cooling pads led to a 70
- **Testing Saves Time**: Developing 57 test scripts at an early stage identified 90

Unresolved Issues:

- `cp -r` fails on empty directories (missing recursive checks).
- Shell history limited to 3 commands due to buffer size constraints.

Respectfully submitted,
Nima Mohamed Ali, Ahmed M. S. Albreem, Nambejja Shukrah Hadijah