Operating Systems Practical Training Course Guide

1. Project Background

Modern academic conferences and journals widely use online peer-review systems (such as OpenReview and EasyChair). A review system must support:

- Authors uploading papers and revised versions
- Reviewers downloading papers and submitting reviews
- Editors assigning reviewers and making final decisions
- Administrators maintaining and backing up the system

To help students understand key concepts of operating systems—including file systems, directory structures, concurrent access, caching, and network communication—this course project requires students to build a simplified but complete academic peer-review system.

2. Project Requirements

The system adopts a Client–Server architecture, implemented in C++. All data (papers, reviews, user roles, version history, etc.) must be stored in the server-side file system. The client provides a command-line interface (CLI) and connects to the server over the network to perform various operations.

The system includes four user roles, each with different permissions. After login, the client CLI should provide role-appropriate commands:

Role	Functions
Author	Upload papers, submit revisions, view review status, download review comments
Reviewer	Download papers, upload review reports, view review status
Editor	Assign reviewers, view review status, make final decisions
Admin	User management, backup management, system status monitoring

File System Requirements:

- Includes superblock, inode table, data block area, and free-block bitmap
- Supports multi-level directories, file creation/deletion, file read/write, and path resolution
- Uses a free-bitmap mechanism for data-block allocation
- Must include a configurable LRU block cache with statistics (hits, misses, replacements)

- Supports backup creation, listing, and restoration

Server–Client Network Architecture Requirements:

- All operations must be initiated from the client and processed by the server
- Server loads and maintains the file system and executes business logic
- Server must support multiple clients concurrently
- Implements authentication and permissions
- Client CLI parses commands, sends protocol messages, and displays responses
- The system may use text-based or binary communication protocols, but the format must be clear, explicit, and extensible.
- Client cannot directly access local files

Optional extensions:

- Backup via file snapshots
- Automated reviewer assignment (conflict-of-interest detection, domain matching, etc.)

3. Task Allocation

- **Requirements Analysis & System Design:** Completed collaboratively by all team members.
- **System Development & Implementation:** Each team selects a technical lead responsible for task distribution and internal/external coordination. Other members collaborate based on their expertise.
- **Testing & Optimization:** Conducted collaboratively by all team members.
- **Documentation Writing:** Completed collaboratively, ensuring clarity and accuracy. The document must clearly describe the internal role distribution within the team.

4. Timeline

- **Week 14**: Submit team formation application, clearly specifying project goals, task allocation, and schedule.
- **Week 16**: Submit mid-term check (no progress requirement, only confirmation).
- **Week 18**: Project evaluation—including PPT presentation and code demonstration—followed by submission of project materials. It is recommended to record the demonstration video in advance for presentation.

5. Evaluation Criteria

- Completeness of system functionality

- Technical complexity and implementation quality
- Teamwork and communication
- Documentation quality

6. Submission Materials

Team application must be submitted to the TAs' email addresses **before December 7, 2025.**

Mid-term check report must be submitted to the TAs' email addresses **before December 21, 2025**.

After final acceptance, each group must package all project-related files and email them **before January 14, 2026**, to yuhanyi@scut.edu.cn, CC to the TA emails.

Email title format: **"操作系统大作业– XX"** where **XX is the group number**.

Submission must include:

- All files required for running the project (source code + data files)
- Project experiment report, following the required template

The printed report must be submitted to B1c-501 **before January 16, 2026**.

Do not use staples for binding.

Only one copy needs to be submitted per group.

7. Additional Notes

- Students may seek guidance and advice from instructors at any time.
- Students must complete the project independently; plagiarism is strictly prohibited.
- A GUI is not required; a terminal (CLI) interface is sufficient.
- Remote login for different user roles may be simulated locally by opening multiple terminal windows.