



National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Computing

CS 330: Operating Systems

BSCS 8C

Lab 12: Synchronized File Management System

CLO4 (Develop programs to interact with OS components through its API)

Time: 10:00 AM – 1:00 PM

Instructor: Dr. Fahad Javed



Lab 12: File Management Server

Instructions

- You are allowed to form groups of up to 3 students per group.
- The code should be developed within the team and any form of plagiarism will automatically result in zero for both the groups involved without any questions asked.
- Each student is responsible to understand the code being submitted under his or her name. Division of work is not explicitly required, however, each group member will be given average viva marks of the group. This implies that if a group member gets zero in viva and another gets full marks then both will score 50%.
- The submission deadline is 2PM next Friday. Viva will be held during the lab hours.
- Any questions and comments on the lab must reach the faculty before 10AM on Friday through email or in person meeting. No questions will be answered after 10 AM on Friday.

Introduction

The purpose of this lab is to make your server built in lab 10 multi-user safe. In lab 10 you developed a server which provides remote access to file system you developed in lab 7. In this lab we will protect the files from reader writer problem. The description of reader-writer problem is described in section 6.6 of your text book. You may use monitors, locks or semaphore libraries provided by the language APIs to implement the tasks

Objectives

By the end of this lab you will learn practical uses of synchronization and implement it on reader-writer problem.

Tasks

There are two task and you may choose any one of the two. The task is to implement mutual exclusion of file access. Multiple threads can attempt to access your files.

Task 1: is to implement a queue for readers. Multiple threads can attempt to read a file and some can request a write.

1. Your task is to make sure that while the file is being read no writer is allowed while maintaining the order of writes as discussed in class.
2. Any request to open the file in write mode is considered to be in active writing till the user closes the file.



National University of Sciences and Technology (NUST) School of Electrical Engineering and Computer Science

3. Similarly any request to open the file in read mode is considered to be in active read till the user closes the file.
4. Multiple users can read the file concurrently but writes will be mutually exclusive.

Task 2: is to implement a limit on the user name to access a file.

1. Multiple users can access your system.
2. Though it is not required that you implement any type of security, but any user may not access more than 5 files.
3. If more than 5 requests are placed then the requesting thread must wait.
4. Each file can only be access by 3 users, be it for read or write.

Requirements

1. You must provide two programs, a server and a client.
2. The client must allow the user to specify the ip address of the server
3. The client will allow user to first specify the user name.
4. The client must provide an interface to apply the operations developed in lab 10.
5. The client must give errors when the server is not available.
6. The client must display the response of the actions performed.
7. The server must respond to multiple requests at the same time (this will require threads)
8. The server must bind to port 95.
9. The server and clients can be run on different machines.

Deliverables

Submit

1. Complete code
2. Sample data file (sample.dat) consisting of files and directories to show your output.
3. User guide to teach how to use your system. The user guide must include description of your directory structure.