Assignment 7

Auto-Grading with Client/Server Model

Team Members: Mriana Yadkoo, Hashim Tayyab, Amitoj Birah, and Nizbath Ahsan.

Overview

This assignment implements a Client-Server system where students can submit their .c source code files. The server performs the following:

- Compiles the submitted file
- Tests it against predefined input-output test cases
- Sends a detailed grading report back to the client

The client application is a Qt-based GUI that allows the student to:

- Connect to the server
- Upload a .c file along with their username
- Receive and display the grading report

The system is built to simulate an automatic grading environment suitable for classroom/homework submission systems. The program uses fork() to create a child process and execl() to execute the submitted program.

Technologies Used

- **C++** for both client and server logic
- **Qt** for building GUI components
- **Socket programming**
- **OpenCV** for displaying the logo image
- **GCC** for compiling student submitted .c files

System Workflow

1. Client App (client_gui)

- Connects to the server.cpp
- Receives the logo and Page executable
- Displays logo inside the Qt window

- Launches Page GUI for upload

2. Upload App (Page Executable)

- Allows user to select a .c file
- Sends file to the server using a TCP connection
- Waits for grading report and displays it in a dialog box

3. Server (server.cpp)

- Sends the logo and executable Page to the client
- Runs on port 9090
- Receives username and .c file
- Creates a folder with the student's name
- Saves the file and compiles it
- Tests the executable using input/output from test.txt
- Generates a detailed report report.txt and sends report back to the client as well

Challenges faced

Initially we were unable to set a server into a single file as we were using 2 separate files, one for sending the logo and Page executable and other for handling the file submission. To overcome this issue, we added a mode variable in the code. In this client-server system, mode is used to distinguish between the two types of communication the server must handle (file transfer and file handling). Before any actual file exchange, the client app sends an int (value 0 or 1) to indicate the mode.

Limitations

One of the limitations is that it is currently single threaded, handles only one client at a time.

Summary

This program is a client server-based auto grading system written in C++ and implemented using sockets, OpenCV and QT GUI library. It accepts a .c file and runs its code against a test file. The name of test file is passed as a command line argument. The program returns a report.txt file that contains