RPS™
learn..evolve

WIPRO NGA Program – 25SUB4508_LSP

Capstone Project Presentation – 13th Feb & 14th Feb 2026

Project Title Here - Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Presented by -

www.rpsconsulting.in

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Introduction :

Managing large volumes of files and directories manually across systems is time-consuming and error-prone. Identifying specific content within distributed storage environments can be difficult and inefficient without automation.

The **Distributed Content Discovery & Inspection Utility (DCDIU)** allows users to recursively traverse directories, search for specific keywords within files, and inspect file contents efficiently from a centralized interface.

It uses a C++ POSIX-based backend with a modular client-server architecture to provide reliable directory scanning, deep content analysis, structured logging, and robust error handling across Unix-like systems.

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Problem Statement :

In modern computing environments, organizations and system administrators face several challenges, such as:

-- Difficulty in locating specific files within deeply nested directory structures.

-- Time-consuming manual inspection of large volumes of files to find relevant content.

-- Inefficient keyword searching across distributed or decentralized storage systems.

-- Lack of centralized tools to perform recursive directory traversal and deep content analysis.

-- Limited visibility into file contents without opening each file individually.

-- A robust, automated utility is essential to ensure efficient, accurate, and scalable content discovery across systems.

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Objective :

➔ Primary Objectives

-- Develop a secure command-line utility for recursive directory traversal and content discovery.

-- Enable searching and inspection of files using keyword-based deep scanning.

-- Provide a centralized system to manage file discovery and inspection efficiently.

➔ Secondary Objectives

-- Ensure portability across Unix-like and POSIX-compliant systems.

-- Maintain structured logging for monitoring and debugging.

-- Support multi-client handling using a modular client-server architecture.

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Key Features:

- Client–Server Architecture

Clear separation between request initiation (client) and processing (server).

- Remote Directory Traversal

Recursive traversal of directories from a configurable base path.

- Deep Content Discovery

Scans file contents for specific strings or patterns across all traversed files.

- File Inspection Mode

Allows targeted inspection and retrieval of file content using absolute paths.

- Structured Response Handling

Server sends processed results back to the client in a readable format.

- Robust Error Handling & Logging

Centralized exception handling and logging for reliability.

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Client –Server Communication Feature:

Description:

This feature enables communication between the client and the server using **TCP sockets**. It forms the foundation of all other system features.

Functional Requirements:

- The server shall listen for incoming client connections.
- The server shall prompt the client for authentication credentials (username and password).
- The client shall provide valid login credentials to the server.
- The server shall authenticate the client using secure hashed password verification.
- Only authenticated clients shall be allowed to execute commands.
- The client shall send text-based commands to the server after successful authentication.

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

- The server shall process commands and send responses back to the client.
- The connection shall remain active until the client issues an exit command..
- The connection shall remain active until the client issues an exit command.

Inputs

Client command strings (e.g., TRAVERSE, SEARCH, INSPECT, EXIT)

Processing

- Socket creation and connection establishment
- Command transmission and reception
- Request parsing on the server side

learn..evolve

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Requirement Tag	Requirement Description
Functional Requirements	
DCDIU_FR_01	The utility shall be capable of performing recursive directory traversals on a remote target server, with an optional configurable base path.
DCDIU_FR_02	DCDIU must support deep-content discovery by scanning for specific strings, phrases, or patterns across all files in the target directory and returning a comprehensive match list.
DCDIU_FR_03	The system shall allow for targeted retrieval of specific files when an absolute path is provided, enabling the user to view the raw data stream.
DCDIU_FR_04	The utility must provide a 'Content Inspection' mode, allowing users to select and display the full data contents of any file identified during the discovery phase.
DCDIU_FR_05	The framework shall implement an exception handling module to display descriptive system messages when traversal or discovery operations fail.
DCDIU_FR_06	A command-line driven interface (CLI) shall be provided, featuring a structured menu for all supported discovery and inspection operations.
DCDIU_FR_07	The system must integrate a diagnostic logging engine with four standardized severity levels: FATAL, INFO, WARNING, and DEBUG.

learn..evolve

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Non-Functional Requirements	Requirement Description
DCDIU_NF_01	Portability: The utility shall be compatible with all standard Unix-like and POSIX-compliant Operating Systems.
DCDIU_NF_02	Modularity: The system architecture shall utilize well-defined interfaces to ensure scalability and the addition of future data-parsing modules.

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Functional Requirements Mapping:

Requirement ID	Description	Implementation Module
DCDIU_FR_01	Client–Server Connection	TCP Socket
DCDIU_FR_02	Recursive Directory Traversal	DirectoryTraverser
DCDIU_FR_03	Deep Content Scanning	ContentScanner
DCDIU_FR_04	File Inspection by Absolute Path	FileInspector
DCDIU_FR_05	Request Parsing & Delegation	ClientHandler / Server
DCDIU_FR_06	Structured Response Delivery	Server → Client Communication
DCDIU_FR_07	Error Handling	ExceptionHandler
DCDIU_FR_08	Logging & Observability	Logger

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Function Module :

1. Directory Traversal Module:
-- Recursive scanning of directories and subdirectories
-- Collect and manage file paths dynamically

2. Content Scanning Module:
-- Search for specific keywords within files
-- Perform deep content discovery across all scanned files

3. File Inspection Module:
-- Display full file content on request
-- Enable detailed inspection of selected files

4. Logging & Exception Module:
-- Maintain structured logs (INFO, WARNING, DEBUG, FATAL)
-- Handle system errors gracefully using ExceptionHandler

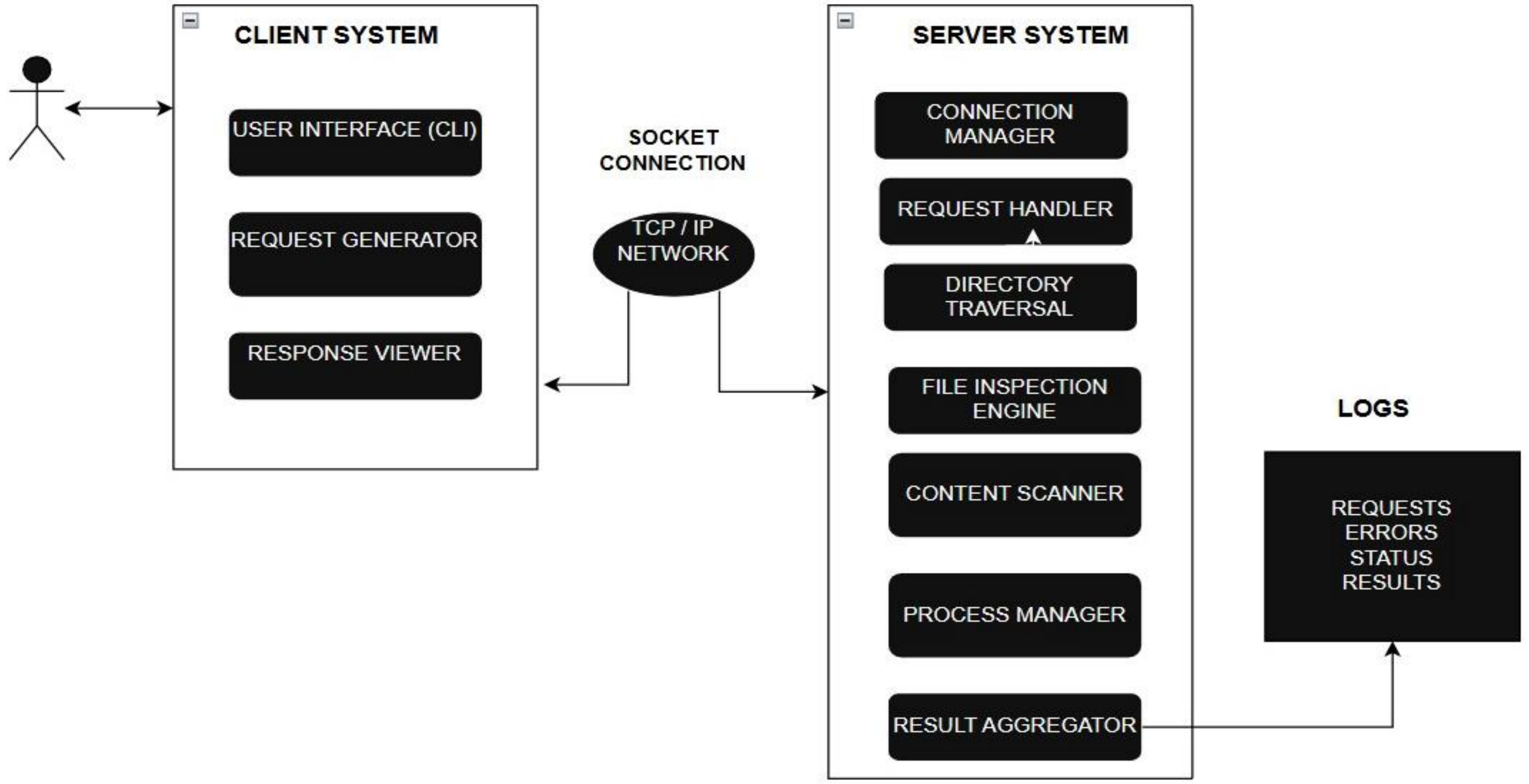
5. Multi-Client Handling Module:
-- Handle multiple client connections concurrently
-- Manage independent client requests using threading

6. Communication Module:
-- Establish client-server socket connection
-- Exchange directory path and keyword data securely

7. System Monitoring Module:
-- Maintain detailed activity logs for auditing
-- Monitor execution status and error reporting

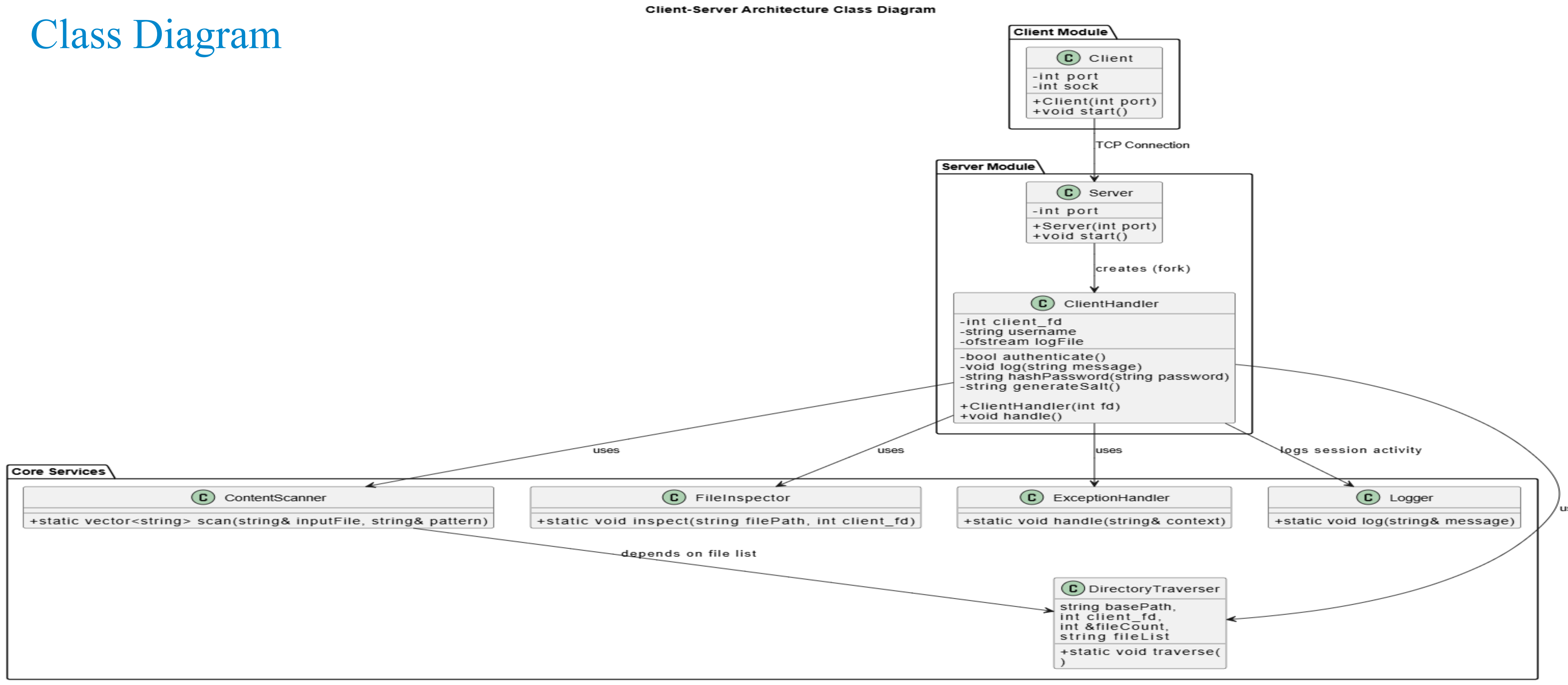
Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

System Architecture:



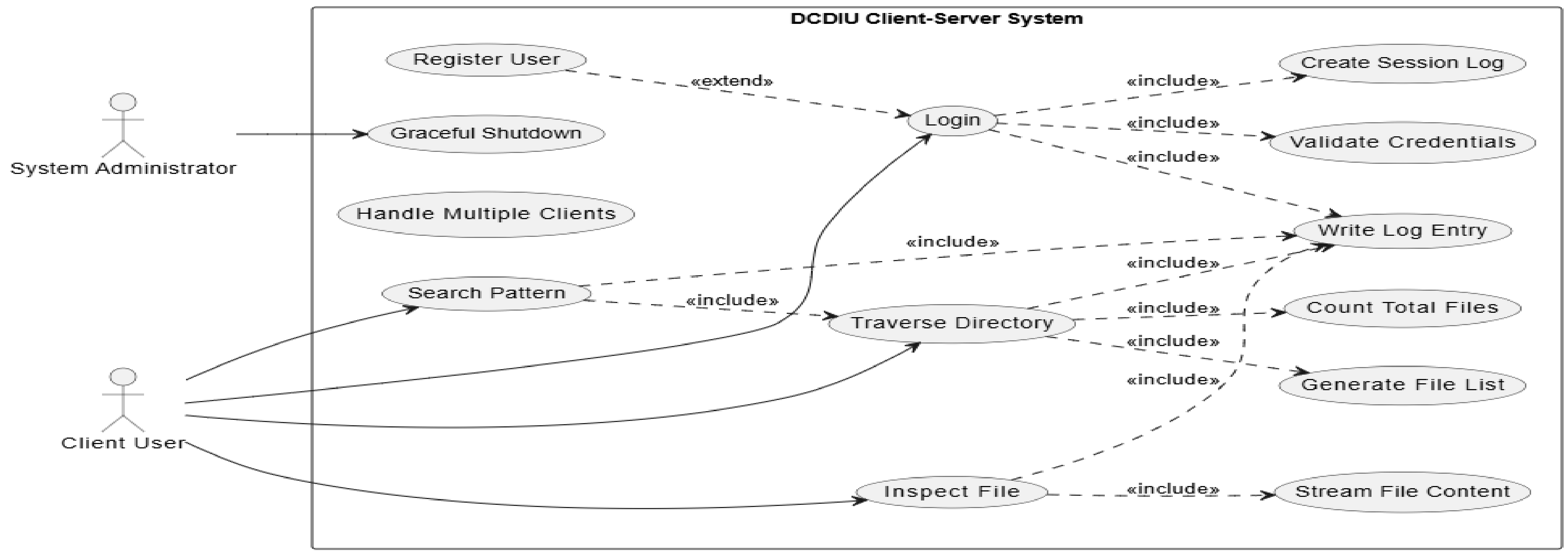
Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Class Diagram



Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Use-Case Diagram:



Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Build Instructions

1.1 Prerequisites

- Linux / WSL / macOS
- g++ (C++17 compatible)
- make
- OpenSSL Development Library

Install OpenSSL:

```
bash
sudo apt update
sudo apt install libssl-dev
```

1.2 Build

From the project root directory:

```
bash
make clean
make
```

This generates:

- build/server
- build/client

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Running the Application

1.1 Start the Server

Open Terminal 1:

```bash
./build/server
```

The server starts listening on the configured port.

1.2 Start Clients

Open Terminal 2 / Terminal 3:

```bash
./build/client
```

Multiple clients can connect simultaneously.

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

How to Use the Application

This section describes the actual execution flow and output of the DCDIU system as observed during testing.

1.1 Starting the Client

Run the client application using:

```
bash
./build/client
```

On successful connection, the client prompts for registration or login.

Example:

```
Username: Mershika
Password: (Enter Your password)
Account created
```

This confirms that the authentication module is working correctly.

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

1.2 Main Menu

After login, the client displays the main menu

```
----------------------
   Display Menu
----------------------
1. Traverse
2. Search
3. Inspect
4. Exit
```

Users can select an option by entering the corresponding number.

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

1.3 Directory Traversal (Option 1)

When the user selects option 1 (Traverse), the system requests a directory path.
Example:

```
Choice: 1
Enter directory path: /home/student/25SUB4508
```

The server performs recursive traversal and sends the results to the client.

Sample Output:

```
Directory: /home/student/25SUB4508
Directory: /home/student/25SUB4508/25sub4508_56117
Directory: /home/student/25SUB4508/25sub4508_56117/Capstone
File: /home/student/.../Custom_system_call_using_Dockerfile_1_.pdf
File: /home/student/.../CA_SOFTWARE.pptx
```

This confirms successful directory scanning and file detection.

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

1.4 Search (Option 2)

Allows users to search for specific files or keywords within the selected directory.

Steps:

1. Select option 2
2. Enter search keyword
3. View matched results

1.5 Inspect (Option 3)

The Inspect feature analyzes selected files and displays detailed information such as size, type, and content patterns.

Steps:

1. Select option 3
2. Provide file or directory path
3. Review inspection report

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

1.6 Exit (Option 4)

Safely terminates the client session and disconnects from the server.

Client Commands

Command	Description
/exit	Exit the client application

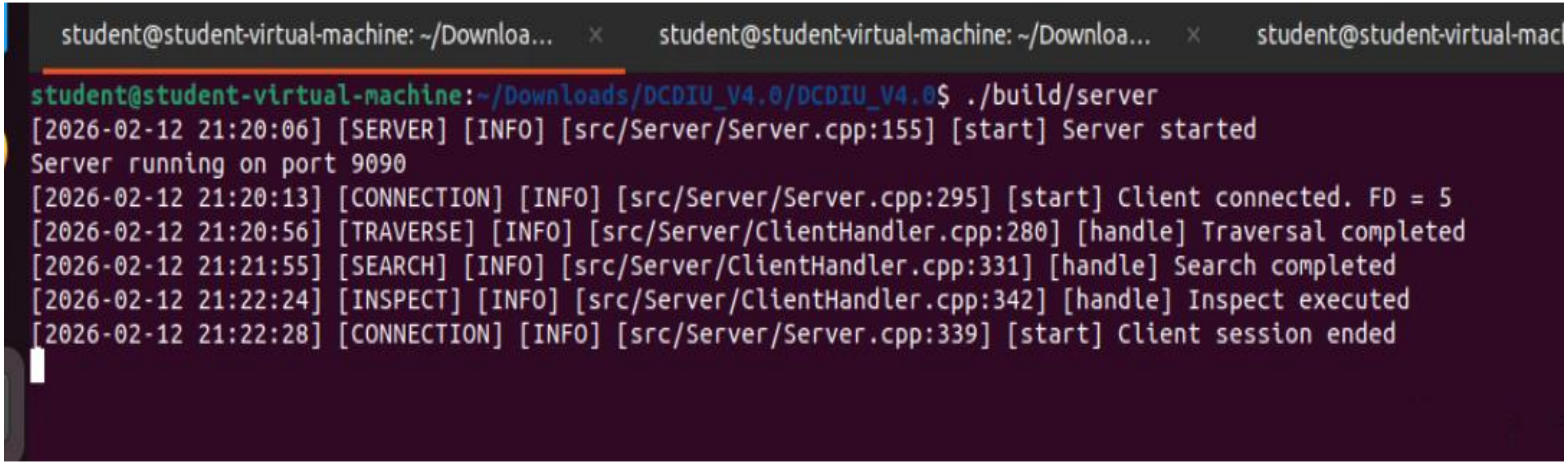
Logging & Observability

Log files are stored in:

-logs/username_pid.log

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Output: Connecting to the server



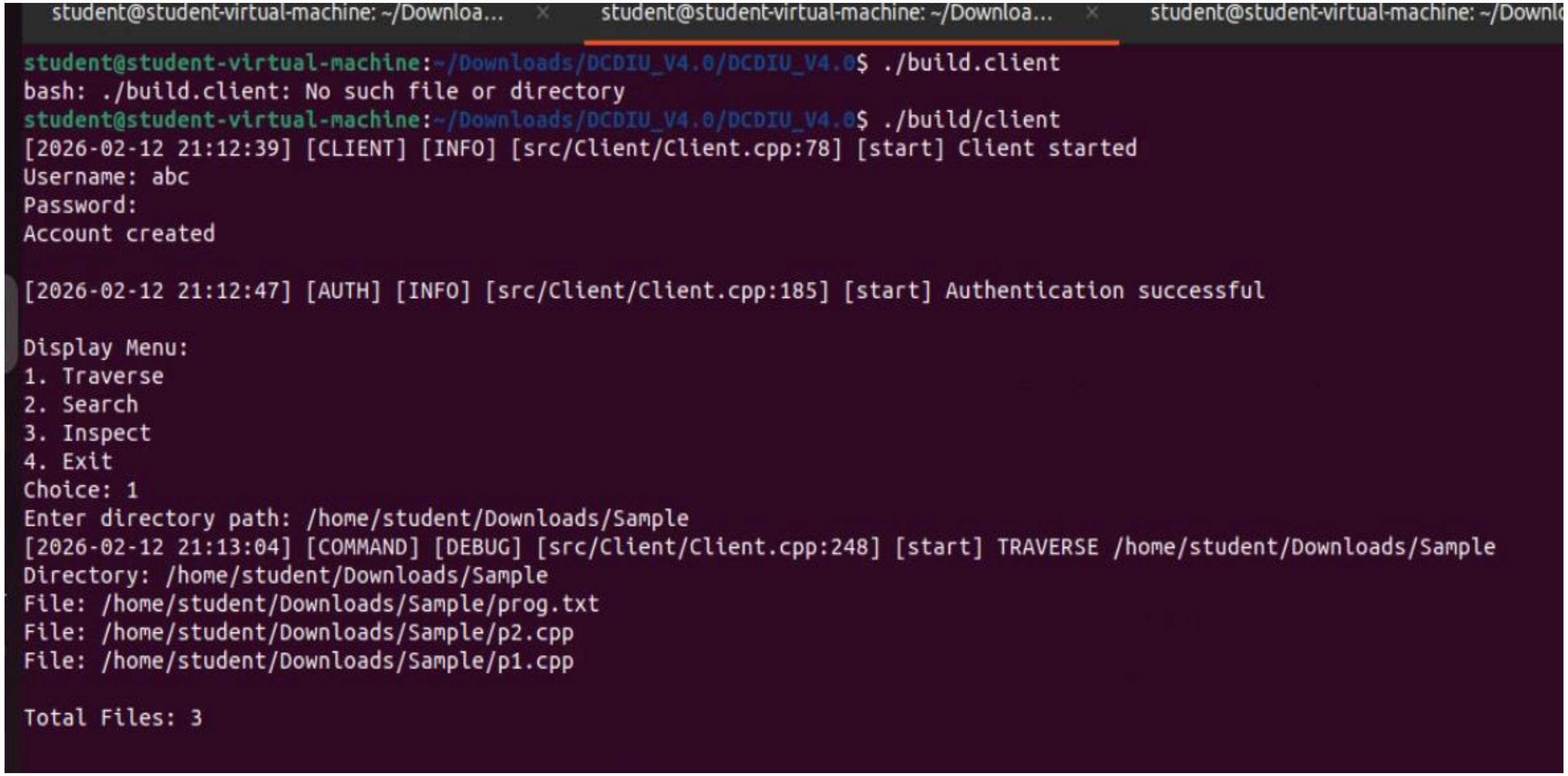
student@student-virtual-machine: ~/Downloa...   student@student-virtual-machine: ~/Downloa...   student@student-virtual-mac

```
student@student-virtual-machine:~/Downloads/DCDIU_V4.0/DCDIU_V4.0$ ./build/server
[2026-02-12 21:20:06] [SERVER] [INFO] [src/Server/Server.cpp:155] [start] Server started
Server running on port 9090
[2026-02-12 21:20:13] [CONNECTION] [INFO] [src/Server/Server.cpp:295] [start] Client connected. FD = 5
[2026-02-12 21:20:56] [TRAVERSE] [INFO] [src/Server/ClientHandler.cpp:280] [handle] Traversal completed
[2026-02-12 21:21:55] [SEARCH] [INFO] [src/Server/ClientHandler.cpp:331] [handle] Search completed
[2026-02-12 21:22:24] [INSPECT] [INFO] [src/Server/ClientHandler.cpp:342] [handle] Inspect executed
[2026-02-12 21:22:28] [CONNECTION] [INFO] [src/Server/Server.cpp:339] [start] Client session ended
```

learn..evolve

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Output: Choice 1)Traverse

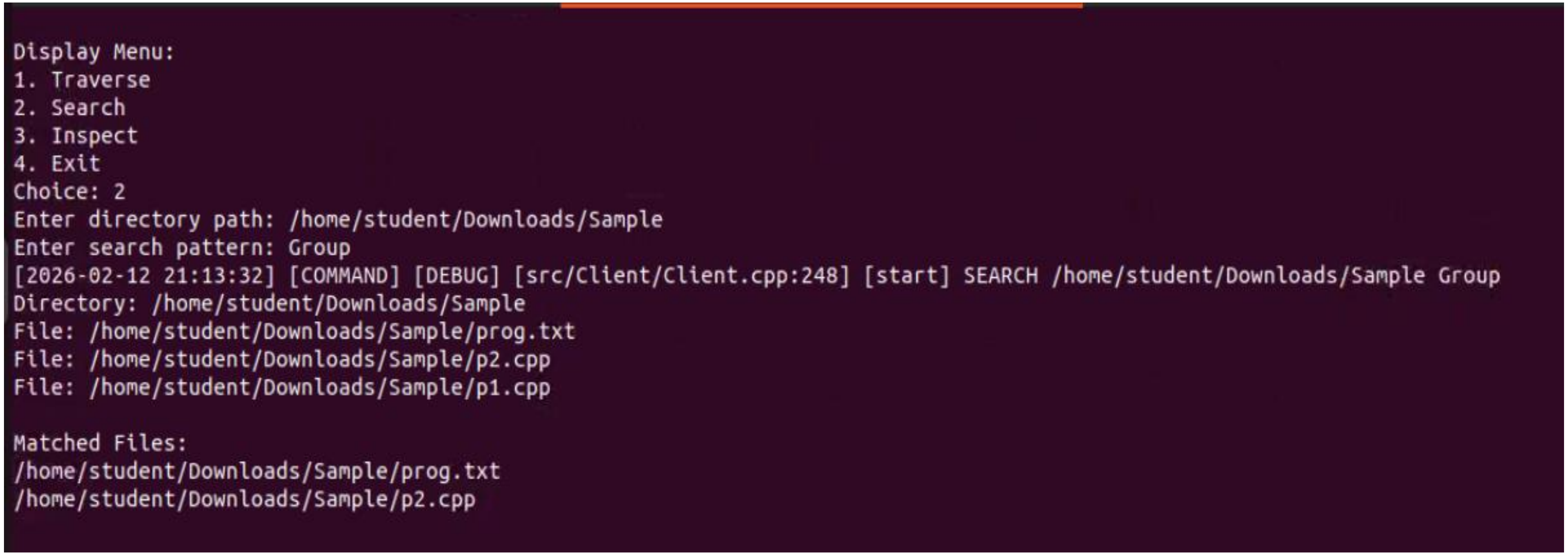

student@student-virtual-machine: ~/Downloa...
student@student-virtual-machine: ~/Downloa...
student@student-virtual-machine: ~/Downlo

```
student@student-virtual-machine:~/Downloads/DCDIU_V4.0/DCDIU_V4.0$ ./build.client
bash: ./build.client: No such file or directory
student@student-virtual-machine:~/Downloads/DCDIU_V4.0/DCDIU_V4.0$ ./build/client
[2026-02-12 21:12:39] [CLIENT] [INFO] [src/Client/Client.cpp:78] [start] Client started
Username: abc
Password:
Account created

[2026-02-12 21:12:47] [AUTH] [INFO] [src/Client/Client.cpp:185] [start] Authentication successful

Display Menu:
1. Traverse
2. Search
3. Inspect
4. Exit
Choice: 1
Enter directory path: /home/student/Downloads/Sample
[2026-02-12 21:13:04] [COMMAND] [DEBUG] [src/Client/Client.cpp:248] [start] TRAVERSE /home/student/Downloads/Sample
Directory: /home/student/Downloads/Sample
File: /home/student/Downloads/Sample/prog.txt
File: /home/student/Downloads/Sample/p2.cpp
File: /home/student/Downloads/Sample/p1.cpp

Total Files: 3
```

RPS learn..evolve

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Output: Choice 2) Search

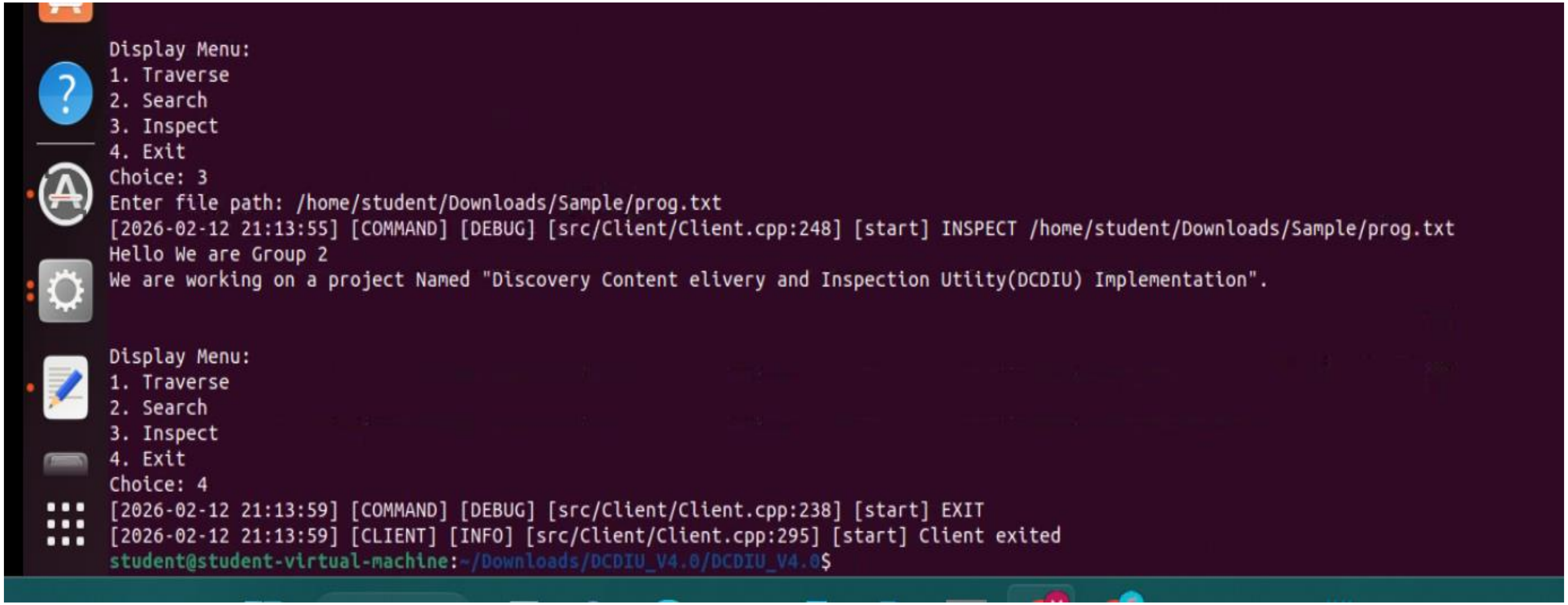


```
Display Menu:
1. Traverse
2. Search
3. Inspect
4. Exit
Choice: 2
Enter directory path: /home/student/Downloads/Sample
Enter search pattern: Group
[2026-02-12 21:13:32] [COMMAND] [DEBUG] [src/Client/Client.cpp:248] [start] SEARCH /home/student/Downloads/Sample Group
Directory: /home/student/Downloads/Sample
File: /home/student/Downloads/Sample/prog.txt
File: /home/student/Downloads/Sample/p2.cpp
File: /home/student/Downloads/Sample/p1.cpp

Matched Files:
/home/student/Downloads/Sample/prog.txt
/home/student/Downloads/Sample/p2.cpp
```

RPS learn..evolve

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

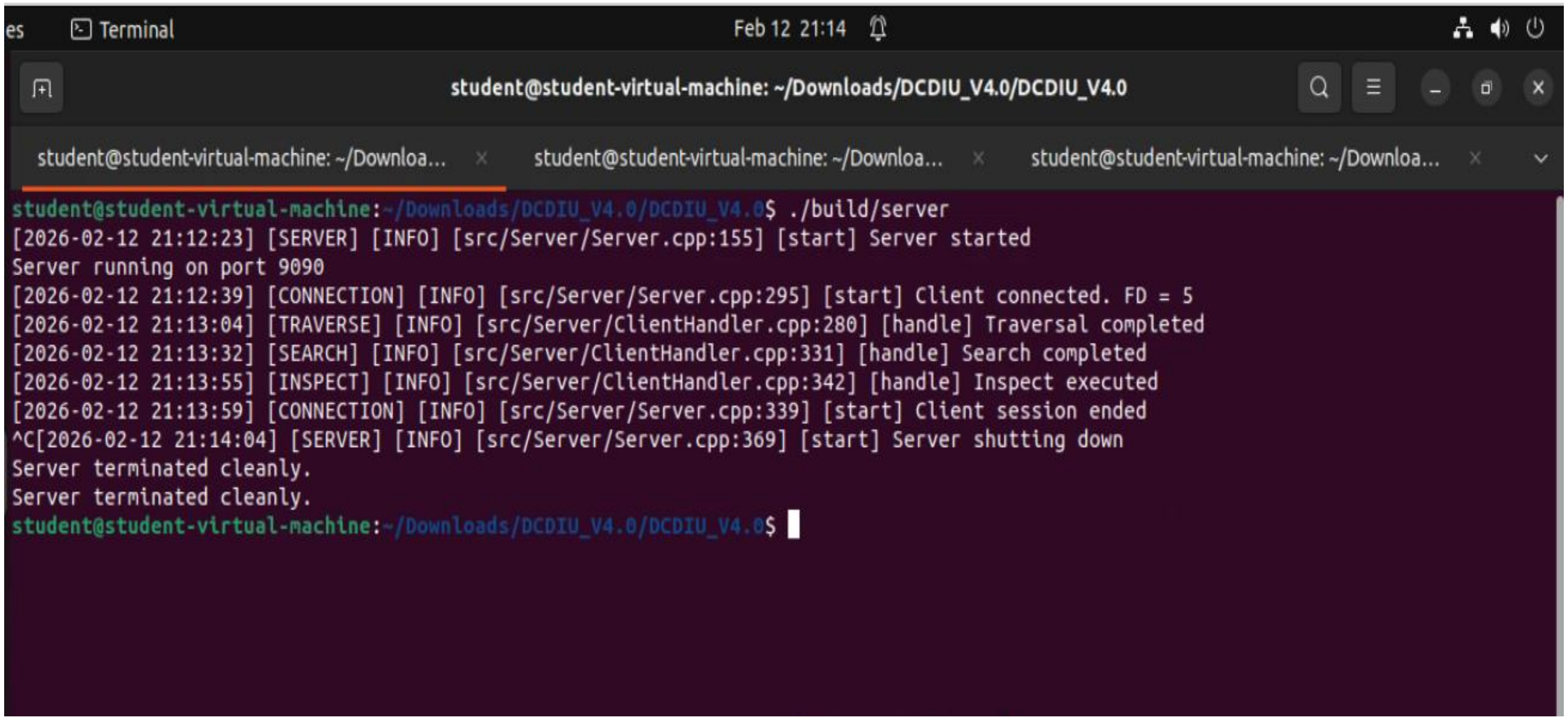
Output: Choice 3) Inspect



```
Display Menu:
1. Traverse
2. Search
3. Inspect
4. Exit
Choice: 3
Enter file path: /home/student/Downloads/Sample/prog.txt
[2026-02-12 21:13:55] [COMMAND] [DEBUG] [src/Client/Client.cpp:248] [start] INSPECT /home/student/Downloads/Sample/prog.txt
Hello We are Group 2
We are working on a project Named "Discovery Content elivery and Inspection Utiity(DCDIU) Implementation".


Display Menu:
1. Traverse
2. Search
3. Inspect
4. Exit
Choice: 4
[2026-02-12 21:13:59] [COMMAND] [DEBUG] [src/Client/Client.cpp:238] [start] EXIT
[2026-02-12 21:13:59] [CLIENT] [INFO] [src/Client/Client.cpp:295] [start] Client exited
student@student-virtual-machine:~/Downloads/DCDIU_V4.0/DCDIU_V4.0$
```

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Output: Disconnecting the server


es
Terminal
Feb 12 21:14
student@student-virtual-machine: ~/Downloads/DCDIU_V4.0/DCDIU_V4.0
student@student-virtual-machine: ~/Downloa...
student@student-virtual-machine: ~/Downloa...
student@student-virtual-machine: ~/Downloa...

```
student@student-virtual-machine:~/Downloads/DCDIU_V4.0/DCDIU_V4.0$ ./build/server
[2026-02-12 21:12:23] [SERVER] [INFO] [src/Server/Server.cpp:155] [start] Server started
Server running on port 9090
[2026-02-12 21:12:39] [CONNECTION] [INFO] [src/Server/Server.cpp:295] [start] Client connected. FD = 5
[2026-02-12 21:13:04] [TRAVERSE] [INFO] [src/Server/ClientHandler.cpp:280] [handle] Traversal completed
[2026-02-12 21:13:32] [SEARCH] [INFO] [src/Server/ClientHandler.cpp:331] [handle] Search completed
[2026-02-12 21:13:55] [INSPECT] [INFO] [src/Server/ClientHandler.cpp:342] [handle] Inspect executed
[2026-02-12 21:13:59] [CONNECTION] [INFO] [src/Server/Server.cpp:339] [start] Client session ended
^C[2026-02-12 21:14:04] [SERVER] [INFO] [src/Server/Server.cpp:369] [start] Server shutting down
Server terminated cleanly.
Server terminated cleanly.
student@student-virtual-machine:~/Downloads/DCDIU_V4.0/DCDIU_V4.0$
```

THANK
YOU