COMP489: Distributed Computing - Assignment 2

Test Plan

by Alex Perrin

July 1, 2025

Testing Environment

For my development environment, I'm using Ubuntu 24.04.2 LTS running in WSL on Windows 11.

```
alex@alex-desktop:~$ cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=24.04
DISTRIB_CODENAME=noble
DISTRIB_DESCRIPTION="Ubuntu 24.04.2 LTS"
```

Install Java 8

The `defualt-jre` and `defualt-jdk` ubuntu packages will install Java 21, this version of Java does not support the libraries for CORBA. We'll need to use Java 8.

```
$ sudo apt install openjdk-8-jre openjdk-8-jdk
```

Set `PATH` to the Java 8 installation.

```
$ export PATH=/usr/lib/jvm/java-8-openjdk-amd64/bin:$PATH
```

Verify versions for `java`, `javac`, and `idlj`

```
alex@alex-desktop:~$ java -version
openjdk version "1.8.0_432"OpenJDK Runtime Environment (build 1.8.0_432-8u432-b06-1~24.04-b06)OpenJDK 64-Bit Server VM (build 25.432-b06, mixed mode)
```

```
alex@alex-desktop:~$ javac --version
javac 1.8.0_452
```

```
alex@alex-desktop:~$ idlj -version
IDL-to-Java compiler (portable), version "3.2"
```

Setup PostgreSQL Database Management System

Install the following ubuntu packages.

```
$ sudo apt install postgresql
```

Verify PostgreSQL installation:

```
alex@alex-desktop:~$ psql --version
psql (PostgreSQL) 16.9 (Ubuntu 16.9-0ubuntu0.24.04.1)
```

Set the password for postgres user.

```
$ sudo -u postgres psql -c "ALTER USER postgres PASSWORD 'postgres';"
```

Create the fileshare database.

```
$ sudo -u postgres createdb fileshare
```

Download PostgreSQL JDBC Driver.

```
$ wget https://jdbc.postgresql.org/download/postgresql-42.7.6.jar
```

Assignment file directory

All assignment program files are located in the `comp489-a2.zip` archive.

`database-schema.sql` is the database schema.

`FileShare.idl` is the IDL interface definition.

`FileShareServer.java` is the application database server.

`FileShareClient.java` is the application P2P client.

`build.sh` , `init.sh` , `start-server.sh` , `start-client.sh` are the testing scripts.

`testfile.txt` is the example file containing “Hello, World!”

`FileShare` , `Stub` , `Helper` , `Holder` , `Operations` , `POA` compiled IDL-to-Java

All java and compiled class files included for reference.

```
alex@alex-desktop:~$ cd comp489-a2
alex@alex-desktop:~/comp489-a2$ ls -1
FileShare.class
FileShare.idl
FileShare.java
'FileShareClient$1.class'
'FileShareClient$FileRequestHandler.class'
```

```
'FileShareClient$SocketServerRunnable.class'
FileShareClient.class
FileShareClient.java
FileShareHelper.class
FileShareHelper.java
FileShareHolder.java
FileShareImpl.class
FileShareOperations.class
FileShareOperations.java
FileSharePOA.class
FileSharePOA.java
FileShareServer.class
FileShareServer.java
_FileShareStub.class
_FileShareStub.java
build.sh
database-schema.sql
init.sh
postgresql-42.7.6.jar
start-client.sh
start-server.sh
testfile.txt
```

1. Compile the application files:

Compile IDL, generates FileShare, Stub, Helper, Holder, Operations, POA in current directory.

```
$ idlj -fall FileShare.idl
```

Compile Java with PostgreSQL driver

```
$ javac -cp ".:postgresql-42.7.6.jar" FileShareServer.java FileShareClient.java
```

`build.sh` script

```
alex@alex-desktop:~/comp489-a2$ ./build.sh
./_FileShareStub.java:153: warning: IORCheckImpl is internal proprietary API and may be removed i
n a future release
  com.sun.corba.se.impl.orbutil.IORCheckImpl.check(str, "_FileShareStub");                    ^Not
e: ./FileSharePOA.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
1 warning
```

2. Initialize Database and CORBA Naming Service:

Import the SQL Database Definition Language to establish the database

```
$ sudo -u postgres psql -U postgres -f database-schema.sql
```

Start CORBA naming service

```
$ orbd -ORBInitialPort 1050 &
```

`init.sh` script

```
alex@alex-desktop:~/comp489-a2$ ./init.sh
CREATE DATABASE
You are now connected to database "fileshare" as user "postgres".
CREATE TABLE
```

3. Start Server:

Start FileShareServer instance with the postgres JDBC driver. and connected to the ORB service.

```
java -cp ".:postgresql-42.7.6.jar" FileShareServer -ORBInitialPort 1050
```

`start-server.sh` script

```
alex@alex-desktop:~/comp489-a2$ ./start-server.sh
FileShare Server ready and waiting...
```

4. Start Client(s):

Start FileShareServer instance connected to the ORB service.

```
java FileShareClient -ORBInitialPort 1050
```

`start-client.sh` script

```
alex@alex-desktop:~/comp489-a2$ ./start-client.sh
Client socket server listening on port 36007
```

Test Results

Single Client Tests

Client Input (STDIN)	Client Output (STDOUT)	Server Output (STDOUT)	Result

search testfile	No files found matching: testfile	SEARCH REQUEST: testfile SEARCH SUCCESS: Found 0 files	No files found initially
register testfile.txt	Registered: testfile.txt	REGISTER REQUEST: testfile.txt from 127.0.0.1:42225 REGISTER SUCCESS: Registered testfile.txt from 127.0.0.1:42225	File registered successfully
search testfile	Search results for 'testfile':testfile.txt	SEARCH REQUEST: testfileSEARCH SUCCESS: Found 1 files	Find registered file
download testfile.txt output1.txt	Downloaded: testfile.txt → output1.txt (13 bytes)	OWNER REQUEST: testfile.txt OWNER SUCCESS: testfile.txt → 127.0.0.1:42225	Download from self successful
unregister testfile.txt	Unregistered: testfile.txt	UNREGISTER REQUEST: testfile.txt from 127.0.0.1:42225 UNREGISTER RESULT: Removed testfile.txt from 127.0.0.1:42225	File unregistered successfully
search testfile	No files found matching: testfile	SEARCH REQUEST: testfileSEARCH SUCCESS: Found 0 files	No files found after unregister

Multi-Client P2P Tests

Client A Actions	Client B Actions	Client A Output	Client B Output	Server Output	Result
Start client A	Start client B	Client socket server listening on port 44715	Client socket server listening on port 42225	Server accepts both connections	Two clients connected
-	register testfile.txt	-	Registered: testfile.txt	REGISTER REQUEST: testfile.txt from 127.0.0.1:42225	Client B registers file
search testfile	-	Search results for 'testfile':testfile.txt	-	SEARCH REQUEST: testfile	Client A finds B's file
download testfile.txt output1.txt	-	Downloaded: testfile.txt → output1.txt (13 bytes)	Served file: testfile.txt (13 bytes)	OWNER REQUEST: testfile.txt OWNER SUCCESS: testfile.txt → 127.0.0.1:42225	Client A downloads from B
-	download testfile.txt output2.txt	-	Served file: testfile.txt (13 bytes) Downloaded: testfile.txt → output2.txt (13 bytes)	OWNER REQUEST: testfile.txt	Client B downloads from self

Input Validation Tests

Client Input (STDIN)	Client Output (STDOUT)	Result
register	Usage: register <filename>	Invalid command format rejected

`unregister`	`Usage: unregister <filename>`	Invalid command format rejected
`search`	`Usage: search <query>`	Invalid command format rejected
`download`	`Usage: download <filename> [output_filename]`	Invalid command format rejected
`register nonexistent.txt`	`File not found: nonexistent.txt`	Cannot register non-existent file
`download nonexistent.txt`	`File not found: nonexistent.txt`	Cannot download non-existent file
`invalid_command`	`Unknown command: invalid_commandType 'quit' to exit or use one of the available commands.`	Invalid commands rejected
`quit`	`Goodbye!`	Application exits gracefully