
Assignment 1: Java RMI String Concatenation

Step 1 (1st Terminal):

- javac -version → Check Java compiler version.
- javac *.java → Compile all .java files.
- rmiregistry → Start RMI registry to allow remote objects.

Step 2 (2nd Terminal):

- java Server → Start the RMI server that will host the remote object.

Step 3 (3rd Terminal):

- java Client → Run the client to connect to the server.
- Enter two strings → The client sends them to the server to concatenate.

Assignment 2: CORBA Reverse String

Step 1 (1st Terminal):

- ls → List files in the directory.
- idlj -fall ReverseModule.idl → Generate Java code from IDL (Interface Definition Language).
- javac *.java ReverseModule/*.java → Compile all Java and generated files.
- orbd -ORBInitialPort 1050& → Start CORBA naming service on port 1050.
- java ReverseServer -ORBInitialPort 1050 -ORBInitialHost localhost& → Start the CORBA server.

Step 2 (2nd Terminal):

- java ReverseClient -ORBInitialPort 1050 -ORBInitialHost localhost → Start the client.
- Enter string: student → Sends string to server, which returns the reversed string.

Assignment 3: MPJ Express – Parallel Array Sum

In Terminal:

- export MPJ_HOME=... → Set the MPJ (Message Passing in Java) installation path.
- export PATH=... → Add MPJ to system path.
- javac -cp $MPJ_HOME/lib/mpj.jar ArrSum.java → Compile program using MPJ library.
- ls → Check files.
- mpjrun.sh -np 1 ArrSum → Run with 1 process (no parallelism).
- mpjrun.sh -np 2 ArrSum → Run with 2 processes to split the array and compute in parallel.

Assignment 4: Socket Programming – Client-Server

Step 1 (1st Terminal):

- python3 server.py → Start the server to wait for a client and respond.

Step 2 (2nd Terminal):

- python3 client.py → Start client to connect to server and exchange data.

Assignment 5: Token Ring Protocol

In Terminal:

- javac TokenRing.java → Compile the Java program.
- java TokenRing → Run the program to simulate token ring.
- Enter number of nodes: 10 → Define number of nodes in the ring.
- Enter sender: 2, Enter receiver: 8 → Send data from node 2 to 8 using the token.
- Repeat with sender 5 and receiver 1 → Test another transfer.

Assignment 6: Bully and Ring Election Algorithms

Step 1 (1st Terminal): Bully Algorithm

- javac Bully.java → Compile bully election program.
- java Bully → Start the election system.

- Choice -1: Initialize, enter number of processes: 5 → Set up 5 processes.
- Choice -2: Display processes.
- Choice -4: Take down process 4 and 3 (simulate failure).
- Choice -3: Bring back process 3.
- Choice -5: Start election by process 2.
- Choice -6: Exit.

Step 2 (2nd Terminal): Ring Algorithm

- javac Ring.java, java Ring → Compile and run Ring election program.
- Choice -1: Initialize 4 processes.
- Choice -2: Display processes.
- Choice -4: Bring down process 4.
- Choice -3: Bring up process 4 again.
- Choice -5: Start election from process 3.
- Choice -6: Exit.
