Experiment 3 – Inter-process Communication

Learning Objective: Demonstrate a program for Inter-process communication

Tools: Java

Theory:

Inter-process Communication

A process is a program in execution. When we say that two computers of a distributed system are communicating with each other, we mean that two processes, one running on each computer, are in communication with each other. In a distributed system, processes executing on different computers often need to communicate with each other to achieve some common goal. For example, each computer of a distributed system may have a resource manager process to monitor the current status of usage of its local resources, and the resource managers of all the computers might communicate with each other from time to time to dynamically balance the system load among all the computers. Therefore, a distributed operating system needs to provide inter-process communication (lPC) mechanisms to facilitate such communication activities.

Inter-process communication basically requires information sharing among two or more processes. The two basic methods for information sharing are as follows:

1. Original sharing, or shared-data approach
2. Copy sharing, or message-passing approach

In the shared-data approach, the information to be shared is placed in a common memory area that is accessible to all the processes involved in an IPC. The shared-data paradigm gives the conceptual communication pattern illustrated in Figure 3.1(a). On the other hand, in the message-passing approach, the information to be shared is physically copied from the sender process's address space to the address spaces of all the receiver processes, and this is done by transmitting the data to be copied in the form of messages (a message is a block of information). The message-passing paradigm gives the conceptual communication pattern illustrated in Figure 3.1(b). That is, the communicating processes interact directly with each other.

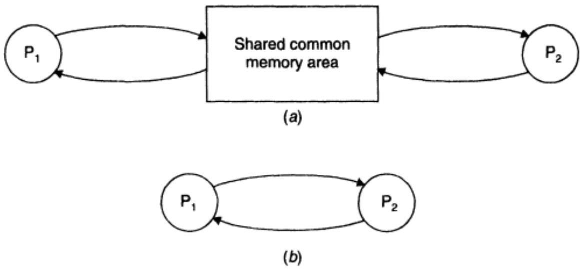


Fig. 3.1 The two basic interprocess communication paradigms: (*a*) The shared-data approach. (*b*) The message-passing approach.

Since computers in a network do not share memory, processes in a distributed system normally communicate by exchanging messages rather than through shared data. Therefore, message passing is the basic IPe mechanism in distributed systems.

A message-passing system is a subsystem of a distributed operating system that provides a set of message-based IPe protocols and does so by shielding the details of complex network protocols and multiple heterogeneous platforms from programmers. It enables processes to communicate by exchanging messages and allows programs to be written by using simple communication primitives, such as send and receive. It serves as a suitable infrastructure for building other higher level lPC systems, such as remote procedure call and distributed shared memory

Model of interprocess communication

The models of interprocess communication are as follows:

Shared Memory Model

Shared memory is the memory that can be simultaneously accessed by multiple processes. This is done so that the processes can communicate with each other. All POSIX systems, as well as Windows operating systems use shared memory.

1. **Advantage of Shared Memory Model**

   Memory communication is faster on the shared memory model as compared to the message passing model on the same machine.

1. **Disadvantages of Shared Memory Model**

   Some of the disadvantages of shared memory model are as follows:

   - All the processes that use the shared memory model need to make sure that they are not writing to the same memory location.
   - Shared memory model may create problems such as synchronization and memory protection that need to be addressed.

Message Passing Model

Multiple processes can read and write data to the message queue without being connected to each other. Messages are stored on queue until their recipient retrieves them. Message queues are quite useful for interprocess communication and are used by most operating systems.

1. **Advantage of Messaging Passing Model**

   The message passing model is much easier to implement than the shared memory model.

2. **Disadvantage of Messaging Passing Model**

   The message passing model has slower communication than the shared memory model because the connection setup takes time.

Characteristics Of Inter-process Communication

There are mainly five characteristics of inter-process communication in a distributed environment/system.

1. **Synchronous System Calls:**

   In the synchronous system calls both sender and receiver use blocking system calls to transmit the data which means the sender will wait until the acknowledgment is received from the receiver and receiver waits until the message arrives.

2. **Asynchronous System Calls:**

   In the asynchronous system calls, both sender and receiver use non-blocking system calls to transmit the data which means the sender doesn't wait from the receiver acknowledgment.

3. **Message Destination:**

   A local port is a message destination within a computer, specified as an integer. Aport has exactly one receiver but many senders. Processes may use multiple ports from which to receive messages. Any process that knows the number of a port can send the message to it.

4. **Reliability:**

   It is defined as validity and integrity.

5. **Integrity:**

   Messages must arrive without corruption and duplication to the destination.

6. **Validity:**

   Point to point message services are defined as reliable, If the messages are guaranteed to be delivered without being lost is called validity.

7. **Ordering:**

   It is the process of delivering messages to the receiver in a particular order. Some applications require messages to be delivered in the sender order i.e the order in which they were transmitted by the sender

Result and Discussion:

Code:

write.java:

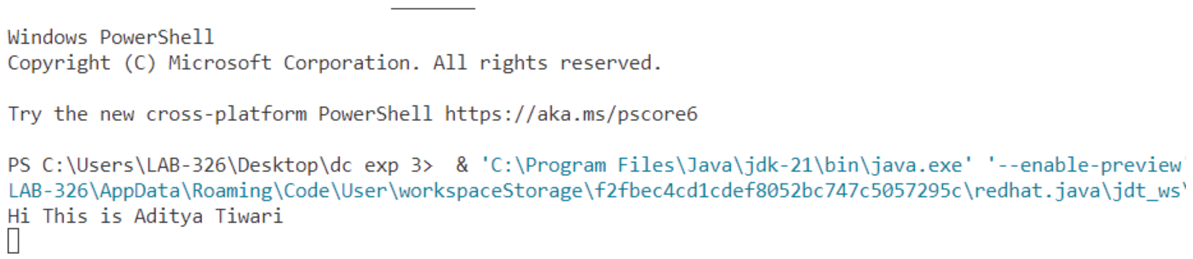
```
package pkg_IPC;

import java.io.*;

public class write {
    public static void main(String[] args) {

    try{
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
        PrintWriter writer = new PrintWriter(new FileWriter("data.txt"));
        String input;
            while ((input = reader.readLine()) != null) {
                writer.println(input);
                writer.flush();
            }
            reader.close();
            writer.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

read.java:

```
package pkg_IPC;

import java.io.*;

public class read {
    public static void main(String[] args) {
        try {
            BufferedReader reader = new BufferedReader(new FileReader("data.txt"));

            String input;
            while ((input = reader.readLine()) != null) {
                System.out.println("Received: " + input);
            }

            reader.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

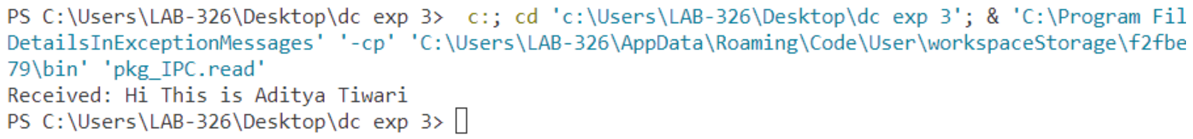
Output:

write.java:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\LAB-326\Desktop\dc exp 3>  & 'C:\Program Files\Java\jdk-21\bin\java.exe' '--enable-preview
LAB-326\AppData\Roaming\Code\User\workspaceStorage\f2fbec4cd1cdef8052bc747c5057295c\redhat.java\jdt_ws'
Hi This is Aditya Tiwari
```

read.java:



```
PS C:\Users\LAB-326\Desktop\dc exp 3>  c:; cd 'c:\Users\LAB-326\Desktop\dc exp 3'; & 'C:\Program Fil
DetailsInExceptionMessages' '-cp' 'C:\Users\LAB-326\AppData\Roaming\Code\User\workspaceStorage\f2fbe
79\bin' 'pkg_IPC.read'
Received: Hi This is Aditya Tiwari
PS C:\Users\LAB-326\Desktop\dc exp 3>
```

Learning Outcomes: The student should have the ability to

LO1: Describe the protocol for Inter process communication.

LO 2: justify that client server are managed properly by the Inter process communication

Course Outcomes: Upon completion of the course students will be able to understand interprocess communication.

Conclusion:

For Faculty Use

Correction Parameters	Formative Assessment [40%]	Timely completion of Practical [40%]	Attendance / Learning Attitude [20%]	
Marks Obtained				