Unit-II

Message Passing

Introduction

Two basic methods for for information sharing as as follows

1. Shared Data Approach

Figure: Shared Data Approach

2. Message Passing Approach

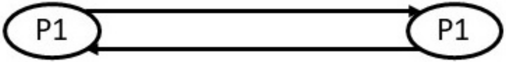


Figure: Message Passing Approach

Desirable Features of a good Message Passing System

1. Simplicity
2. Uniform Semantics
   - Local Communication
   - Remote Communication
3. Efficiency
4. Reliability
5. Correctness
   Issues related to correctness are
   - Atomicity
   - Ordered Delivery
   - Survivability
6. Flexibility
7. Security
8. Portability

Issues in IPC by Message Passing I

A message is a block of information formatted by a sending process in such a manner that it is meaningful to the receiving process.

In the designing of an IPC protocol for message-passing system, the following important issues need to be considered.

Issues in IPC by Message Passing II

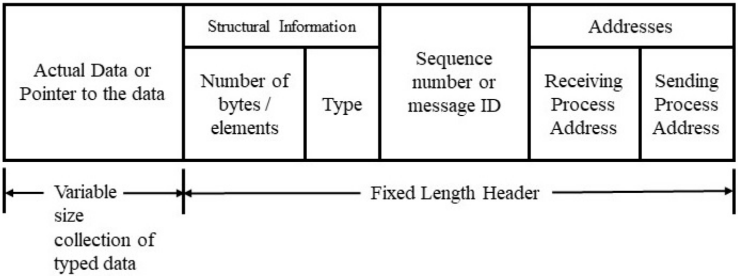


Figure: A Typical Message Structure

Issues in IPC by Message Passing III

In designing of an IPC for MPS, the following important issues need to be considered:

1. Who is the sender?
2. who is the receiver?
3. Is there one receiver or many receivers?
4. Is the message guaranted to have been accepted by its receiver?
5. Does the sender need to wait for a reply?
6. What should be done is a catastrophic event such as a node crash of a communication link failure occurs during the course of communication?

Issues in IPC by Message Passing IV

7. What should be done if the receiver is not ready to accept the message:Will the message be discarded or stored in a buffer? In case of buffering,what shoul be done if athe buffer is full?
8. is there are several outstanding messages for a receiver, can it choose the order in which to service the outstanding messages?

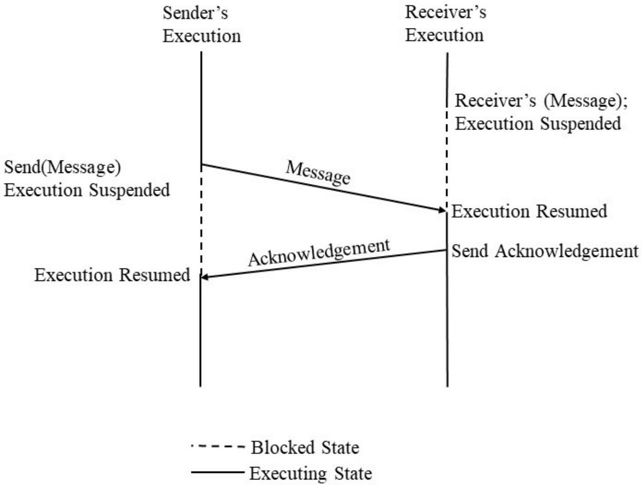
Synchronization I

A central issue in the communication structure is the synchronization. The semantics used synchronization may be broadly classified as

- Blocking
- Non-Blocking

When both the send and receive primitives of a communication between two processes use blocking semantics, the communication is said to be synchronous; otherwise it is asynchronous.

Synchronization II



Synchronization III

Figure: Synchronous Mode of Communication with send and receive primitives having blocking type semantics.

Buffering I

The synchronous and asunchronous mode of communication correspond respectively to the two extremes of buffering: a Null Buffer or No Buffering and a buffer with unbounded capacity. Other two commonly used buffering strategies are Single-buffering and finite bound or multiple message buffers. These four types of buffering strategies are;

- Null Buffer or No Buffering
- Single Message Buffer
- Unbounded-Capacity Buffer
- Finite Bound or Multiple Message Buffer

Buffering II

Fig. (a) Message Transfer in synchronous send with no buffering strategy

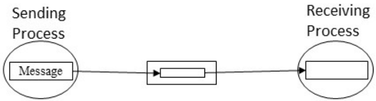


Fig. (a) Message Transfer in synchronous send with single message buffering strategy

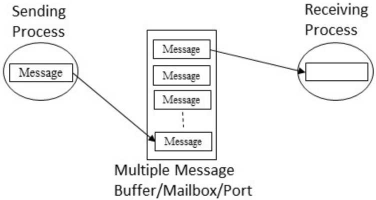


Fig. (a) Message Transfer in asynchronous send with multiple message buffering strategy

Multidatagram Messages

- Datagram
- MTU
- Single Datagram Messages
- Multi Datagram Messages

Encoding and Decoding of Message Data

A message data should be meaningful to the receiving process. This implies that, ideally, the structure of the program object should be preserved while they are being transmitted from address space pf the sending process to the address space of the receiving process. This obviously is not possible in a heterogeneous systems in which the sending and receiving processes are on different computers of different architectures. However, even in homogeneous systems, it is very difficult to achieve this goal mainly because of two reasons:

1. An absolute pointer value loses
2. Different program objects occupy varying amount of storage space.

Due to above mentioned problem encoding and decoding is done.

One of the following two representations may be used for

Process Addressing

Another important issue in message based communication is addressing(or naming)of the parties involved in an interaction.
MPS usually supports two types of process addressing.

- Explicit Addressing
- Implicit Addressing

Failure Handling I

While distributed system may offer potential for parallelism, it is also prone to partial failure such as node crash or a communication link failure. Therefore, during interprocess communication, such failures may lead the following problems:

1. Loss of Request Message
2. Loss of Response Message
3. Unsuccessful Execution of the Request

Failure Handling II

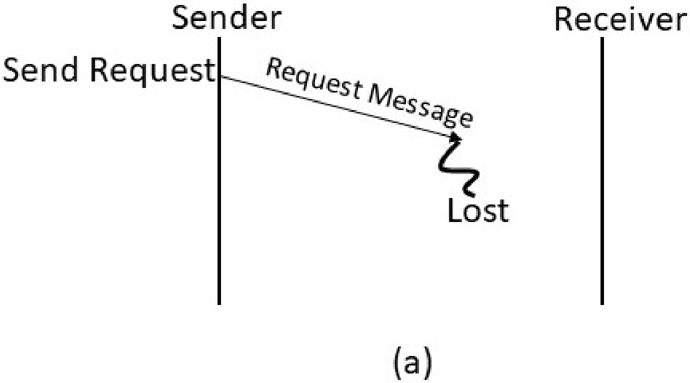


Figure: Request message is lost

Failure Handling(Continues...)

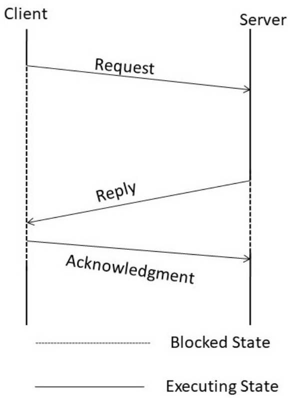


Figure: Three-message reliable IPC protocol

Failure Handling(Continues...)

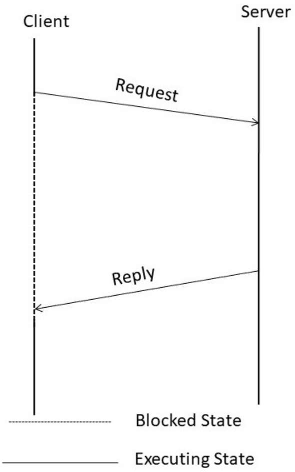


Figure: Two-message reliable IPC protocol

Failure Handling(Continues...)

Consider an example of fault-tolerant communication between a client and a server

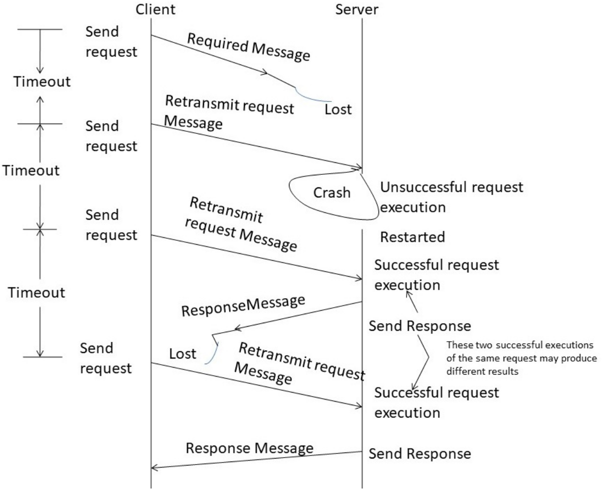


Figure: fault-tolerant communication between a client and a server

Idem-potency and Handling of Duplicate Messages I

Consider the following routine of a server process that debits a specified amount from a bank and returns the balance amount to a requesting client.

Idem-potency and Handling of Duplicate Messages II

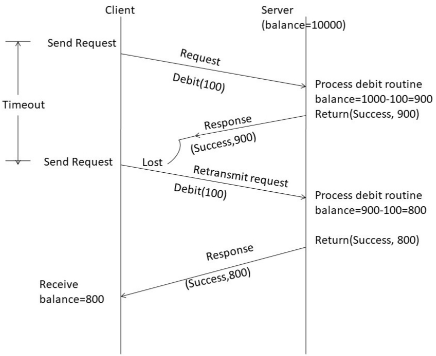


Figure: A Non-idempotent routine

Idem-potency and Handling of Duplicate Messages III

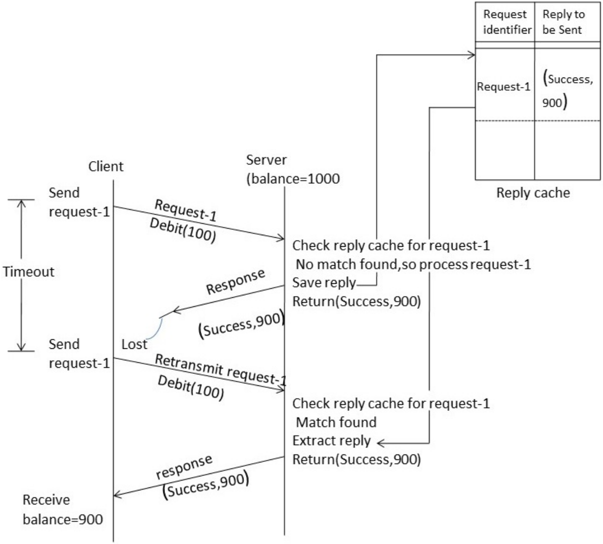


Figure: Exactly once semantics

Group Communication I

Depending on single or multiple senders and receivers, the following are the three types of group communications;

1. One-to-Many
2. Many-to-One
3. Many-to-Many

Group Communication II

1. One-to-Many
   - Group Management
   - Group Addressing
   - Message Delivery to Receiver Process
   - Buffered and Unbuffered Multicast
   - Send-to-All and Bulletin-Board Semantics
   - Flexible Reliability in Multicast Communication
   - Atomic Multicast
   - Group Communication Primitives