Lab Assignment 2

Inter-Process Communication

Max Marks: 10

You are to design an inter-process communication (IPC) utility that allows any two applications to communicate with each other by passing messages. The maximum limit on the number of such applications is 10.

Constraints on your design:

- Single---core system
- No system calls are allowed in this implementation

Deliverables:

1. IPC Utility

Testing Interface:

Initialization Commands

1. init_IPC app1 app2 [app3, …]

The directive initializes IPC utility with message queues setup for each listed application on the command line. The queues will be initialized as empty.

Status Commands

1. LIST [queue name]

The directive “list” has arguments of optionally a queue name. The command will list the elements of the named queue. With no arguments, it will list the elements of all queues, separated by queue with the name of the queue listed.

2. HAS_Message [queuename]

This directive has the argument of a queue name. It returns TRUE if there is a message in the named queue and FALSE if there is nothing in the queue.

Action Commands

1. Send [source] [destination] [message]

This command tells the IPC utility that the source application wants to send a message to the destination application. Source and destination should be valid applications in the system, such as process, memory, filesystem, etc. It will enqueue the message in the appropriate message queue.

2. Retrieve [destination]

This command tells the IPC utility to return the next message for destination. It will dequeue the message and display the contents.

Expected use of this test interface will be like:

- Multiple “Send” commands to put messages in the queues
- An occasional “list” to see what is where
- Specific “retrieve” commands to consume messages.

Tip:

You need to think about the message format.