Inter-process communication Software Technologies - Lecture 9

Saurabh Barjatiya

International Institute Of Information Technology, Hyderabad

5^{th} April, 2010

Message queues - Introduction

Message queues can be used to pass messages from one program to another asynchronously. The two programs need not even be running at same time to exchange information using message queues.

There is limit on size of each message and also number of messages that can be in queue. Read man page of msgsnd(), msgrcv() or msgctl() to know about how to use message queues.

Message queues - Caution

Do not include size of long int message_type

One of the very common mistakes that people do when they use message queues for first time is they use `sizeof(struct my_message)` and not `sizeof(struct my_message) - sizeof (long int)`. For some strange reason size of long int message_type need not be included in arguments to function msgsnd() and msgrcv().

Semaphores - Introduction

Semaphores can be used for synchronization. If we want to perform any atomic operation or synchronize two process/threads then semaphore provide very efficient and clean way to achieve proper results.

Read man pages of semctl(), semget() and semop() to know about what all can be accompalished using semaphores.

Shared memory - Introduction

Shared memory can be used to shared large chunks of information between processes running on same system. This is used a lot between GUI programs and X server to exchange information so that GUI display information exchange is not very expensive.

Read man pages of shmat(), shmctl(), shmdt() and shmget() to know about shared memory functionalities provided by Linux OS.