

MINOR ASSIGNMENT-11

Inter-Process Communication: Shared Memory & Message Queue Practical Programming with C (CSE 3544)

Publish on: 21-12-2024

Course Outcome: CO₅

Program Outcome: PO₃

Submission on: 28-12-2024

Learning Level: L₅

Problem Statement:

Experiment with inter-process communication mechanism(IPC) using shared memory and message queue..

Assignment Objectives:

Students will be able to learn how communication between processes is established using the IPC mechanisms, shared memory and message queue.

Answer the followings:

1. Write a program to create a shared memory segment of size 10 bytes. Make 4 attachments to the shared memory segment to the address space of the calling process and print the number of attachments using the structure filed *number of current attachments* present in the structure **shmid_ds** defined in the header **<sys/shm.h>**. Check the number of attachment using the shell provided command **ipcs -m**.

Code here

Specify: input & output

2. Create a C code named **shmwriter.c** to create a shared memory segment of integer size and store 500 to the segment. Create another program named **shmreader.c** to access the stored value from the shared memory segment and display it. Let the **shmreader.c** update the value to 600. Now update the **shmwriter.c** code to get the updated value and display it. You are not allowed to use `semaphore`.

Code here

Specify: input & output

3. Create 2 processes using **fork()**. The child will sent a number to parent using shared memory segment. The parent willl display the received number and doubles it and sends back to the client. The client will display the received number.

Code here

Specify: input & output

4. Design a C code to create a message queue and add 4 messages to the queue. Create a receiver code to receive all the messages from the queue till the queue is empty.

Code here

Specify: input & output

- 5*. Write a C code to create a message queue. Write 6 messages of message the type 10, 30, 46, 67, 78, and 88 onto the queue. Create a receiver code to receive the message depending on the **msgtyp** parameter of the **msgrcv** system call as **msgtyp=-10**, **msgtyp=100**, **msgtyp=-46**, **msgtyp=0**, and **msgtyp=88** respectively.

Code here

Specify: input & output

- 6*. Write a program to read a string **iter** and encrypt the string using a cryptographic technique called caesar cipher with a key value of 5. The encryption can be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, $A = 0, B = 1, \dots, Z = 25$. Encryption of a letter say x by a key k can be described mathematically as $E_k(x) = (x + k) \bmod 26$. After encryption write the key value and encrypted message on to the queue. Create a receiver code to get the message and the key value. Decrypt the received message using the reverse process as $D_k(x) = (x - k) \bmod 26$. After decryption display the message on the receiver side.

Example-1**Text : asdzf****key: 3****Cipher: dvgci****Example-2****Text : ATTACKATONCE****Shift: 4****Cipher: EXXEGOEXSRGI****Code here****Specify: input & output**

Code here

Specify: input & output