#### Inter-process Communication

# Aim

To implement programs for inter process communication

# Description

**tok():** is use to generate a unique key.

**shmget():** int shmget(key\_t,size\_tsize,intshmflg); upon successful completion, shmget() returns an identifier for the shared memory segment.

**shmat():** Before you can use a shared memory segment, you have to attach yourself to it using shmat(). void \*shmat(int shmid ,void \*shmaddr ,int shmflg);

shmid is shared memory id. shmaddr specifies specific address to use but we should set it to zero and OS will automatically choose the address.

**shmdt():** When you’re done with the shared memory segment, your program should detach itself from it using shmdt(). int shmdt(void \*shmaddr);

**shmctl():** when you detach from shared memory,it is not destroyed. So, to destroy shmctl() is used. shmctl(int shmid,IPC\_RMID,NULL);

**Algorithm Sender:-**

1)Start

2) Create a shared memory segment using shmget with a specific key and size.

3) Attach the shared memory segment to the current process using shmat.

4) Display "Enter a string: "

5) Read the input string from the user.

6) Copy the input string into the shared memory segment.

7) Stop

**Algorithm Receiver:-**

1)Start

2) Create a shared memory segment using shmget with a specific key and size.

3) Attach the shared memory segment to the current process using shmat.

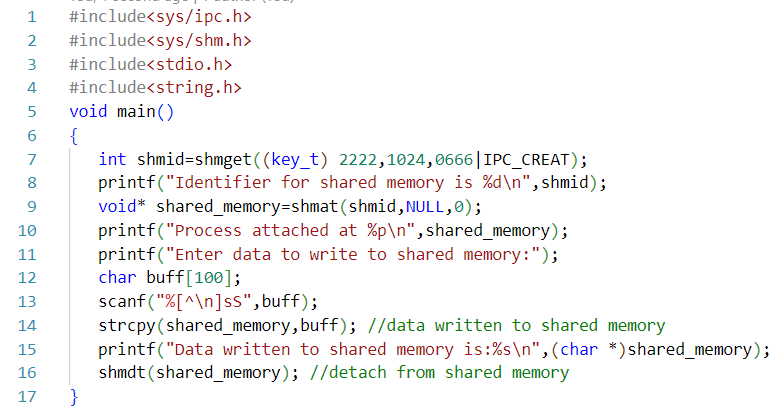
4) Read data from the shared memory segment.

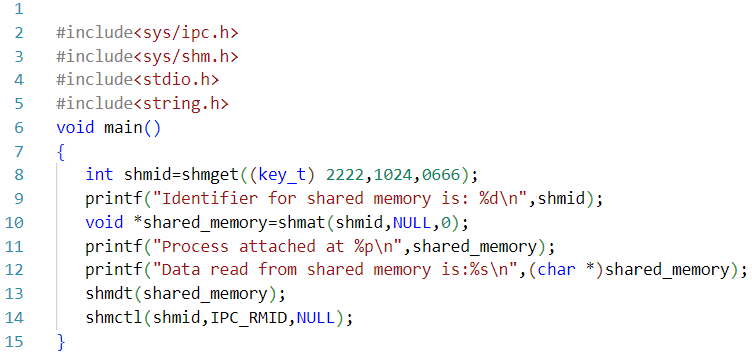
5) Display the read data to the console.

6) Stop

# Code

To write into shared memory:



To read from shared memory

# Output



# Result

The programs have been executed and output has been verified.