**Assignment 9**

**Aim –** Implement the process synchronization with the structure of reader and writer process

Code :

#include<semaphore.h>

#include<stdio.h>

#include<stdlib.h>

sem\_t x,y;

pthread\_t tid;

pthread\_t writerthreads[100],readerthreads[100];

int readercount;

void \*reader(void\* param)

{

sem\_wait(&x);

readercount++;

if(readercount==1)

sem\_wait(&y);

sem\_post(&x);

printf("\n%d reader is inside",readercount);

sem\_wait(&x);

readercount--;

if(readercount==0)

{

sem\_post(&y);

}

sem\_post(&x);

printf("\n%d Reader is leaving",readercount+1);

}

void \*writer(void\* param)

{

printf("\nWriter is trying to enter");

sem\_wait(&y);

printf("\nWriter has entered");

sem\_post(&y);

printf("\nWriter is leaving");

}

int main()

{

int n2,i;

printf("Enter the number of readers:");

scanf("%d",&n2);

int n1[n2];

sem\_init(&x,0,1);

sem\_init(&y,0,1);

for(i=0;i<n2;i++)

{

pthread\_create(&writerthreads[i],NULL,reader,NULL);

pthread\_create(&readerthreads[i],NULL,writer,NULL);

}

for(i=0;i<n2;i++)

{

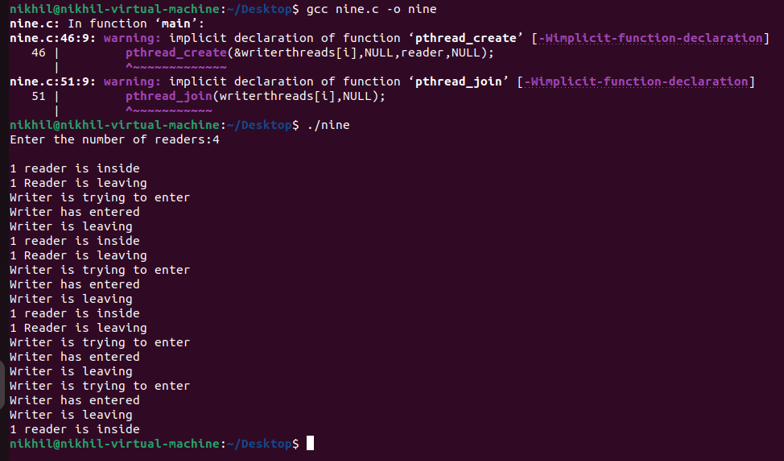
pthread\_join(writerthreads[i],NULL);

pthread\_join(readerthreads[i],NULL);

}

}

**OUTPUT :**



Conclusion : Hence we have successfully implemented reader and writer process.

**THANK YOU!!!!!!!!!!**