**Muhammad Abdullah**

**SE(4A) | 19F-0916**

Operating System Lab

Synchronization Mutex Locks, Semaphores, Conditional variables

**Question # 01**

**CPP CODE:**

**#include <iostream>**

**#include <pthread.h>**

**#include<unistd.h>**

**using namespace std;**

**//Question 1**

**int GlobalVariable;**

**pthread\_mutex\_t Locked;**

**pthread\_cond\_t Condition;**

**void\* Recieved ( void\* val)**

**{**

**string Convert = (char \*) val;**

**cout<<"Message has been recieved : "<<Convert<<endl;**

**pthread\_exit(NULL);**

**}**

**void\* threadReleased()**

**{**

**pthread\_mutex\_lock (&Locked);**

**pthread\_t thread;**

**char\* Message = "Humpty Dumpty Sat On A Wall\nBecause usko ghar nahi rehney detey ";**

**pthread\_create(&thread,NULL,Recieved, (void \*)Message);**

**pthread\_join(thread,NULL);**

**++GlobalVariable;**

**pthread\_cond\_signal(&Condition);**

**pthread\_mutex\_unlock(&Locked);**

**}**

**void\* Sent(void \* val)**

**{**

**threadReleased();**

**}**

**void\* threadJoin()**

**{**

**pthread\_mutex\_lock(&Locked);**

**while(GlobalVariable==0)**

**{**

**pthread\_cond\_wait(&Condition,&Locked);**

**pthread\_mutex\_unlock(&Locked);**

**}**

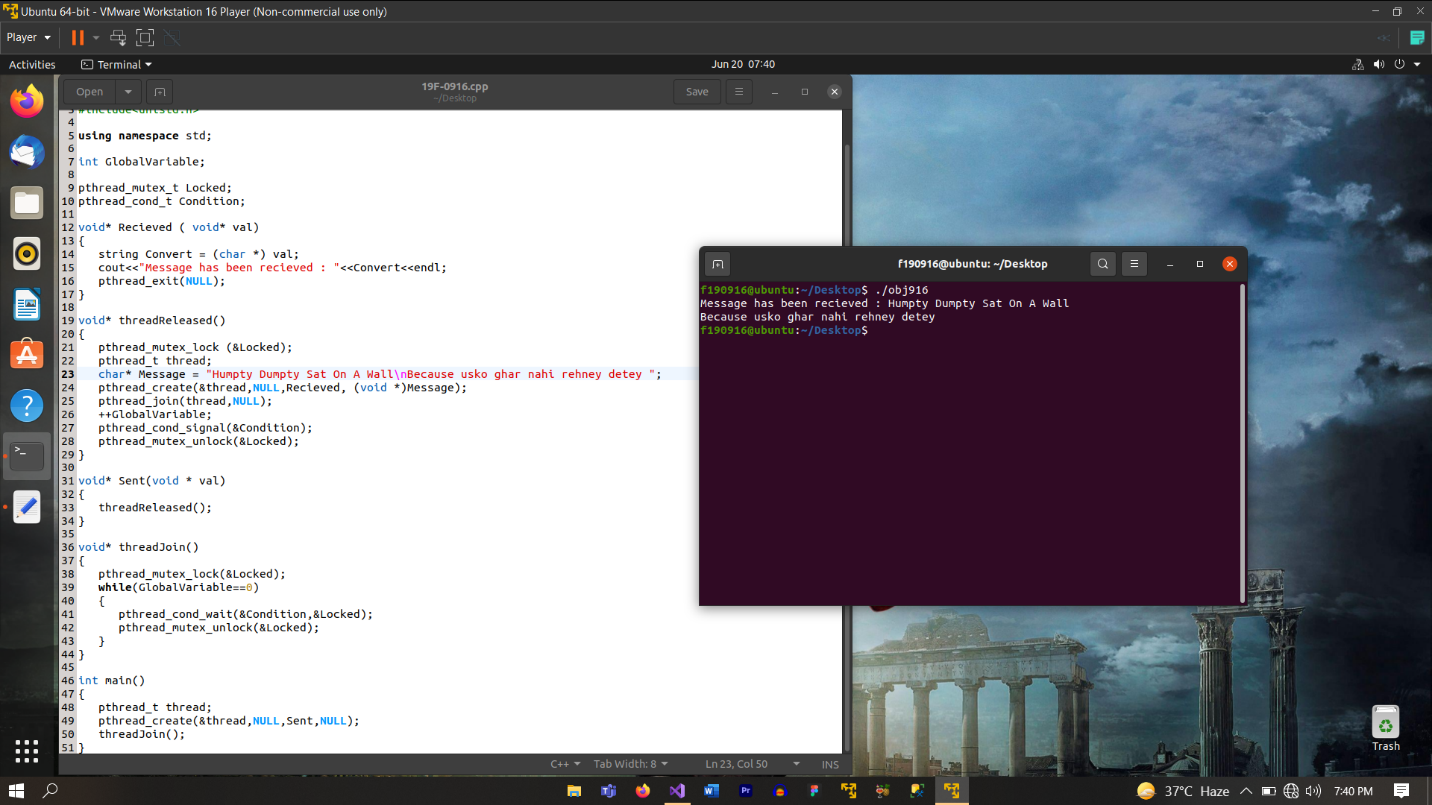
**}**

**int main()**

**{**

**pthread\_t thread;**

**pthread\_create(&thread,NULL,Sent,NULL);**

** threadJoin();**

**}**

**Question # 02**

**Cpp code:**

**#include <iostream>**

**#include <pthread.h>**

**#include<unistd.h>**

**using namespace std;**

**// Question 2**

**pthread\_mutex\_t Locked;**

**void\* Alpha (void \* val)**

**{**

**pthread\_mutex\_lock(&Locked);**

**int OB=0,NB=0,CB=100,AB=10000;**

**OB=CB;**

**NB=OB+AB;**

**CB=NB;**

**cout<<"We are in Parent Thread" <<endl;**

**cout<<"Balance After Parent Deposited it :" << CB << endl;**

**pthread\_mutex\_unlock(&Locked);**

**pthread\_exit(NULL);**

**}**

**void\* Bravo (void \* val)**

**{**

**pthread\_mutex\_lock(&Locked);**

**int OB=0,NB=0,CB=100,AB=5000;**

**OB=CB;**

**NB=OB+AB;**

**CB=NB;**

**cout<<"We are in Brother Thread" <<endl;**

**cout<<"Balance After Brother Deposited it :" << CB << endl;**

**pthread\_mutex\_unlock(&Locked);**

**pthread\_exit(NULL);**

**}**

**void\* Charlie (void \* val)**

**{**

**pthread\_mutex\_lock(&Locked);**

**int OB=0,NB=0,CB=100,AB=5500;**

**OB=CB;**

**NB=OB+AB;**

**CB=NB;**

**cout<<"We are in Cousin Thread" <<endl;**

**cout<<"Balance After Cousin Deposited it :" << CB << endl;**

**pthread\_mutex\_unlock(&Locked);**

**pthread\_exit(NULL);**

**}**

**int main()**

**{**

**pthread\_mutex\_init(&Locked,NULL);**

**pthread\_t FirstThread;**

**pthread\_t SecondThread;**

**pthread\_t ThirdThread;**

**pthread\_create(&FirstThread,NULL,Bravo,NULL);**

**pthread\_create(&SecondThread,NULL,Alpha,NULL);**

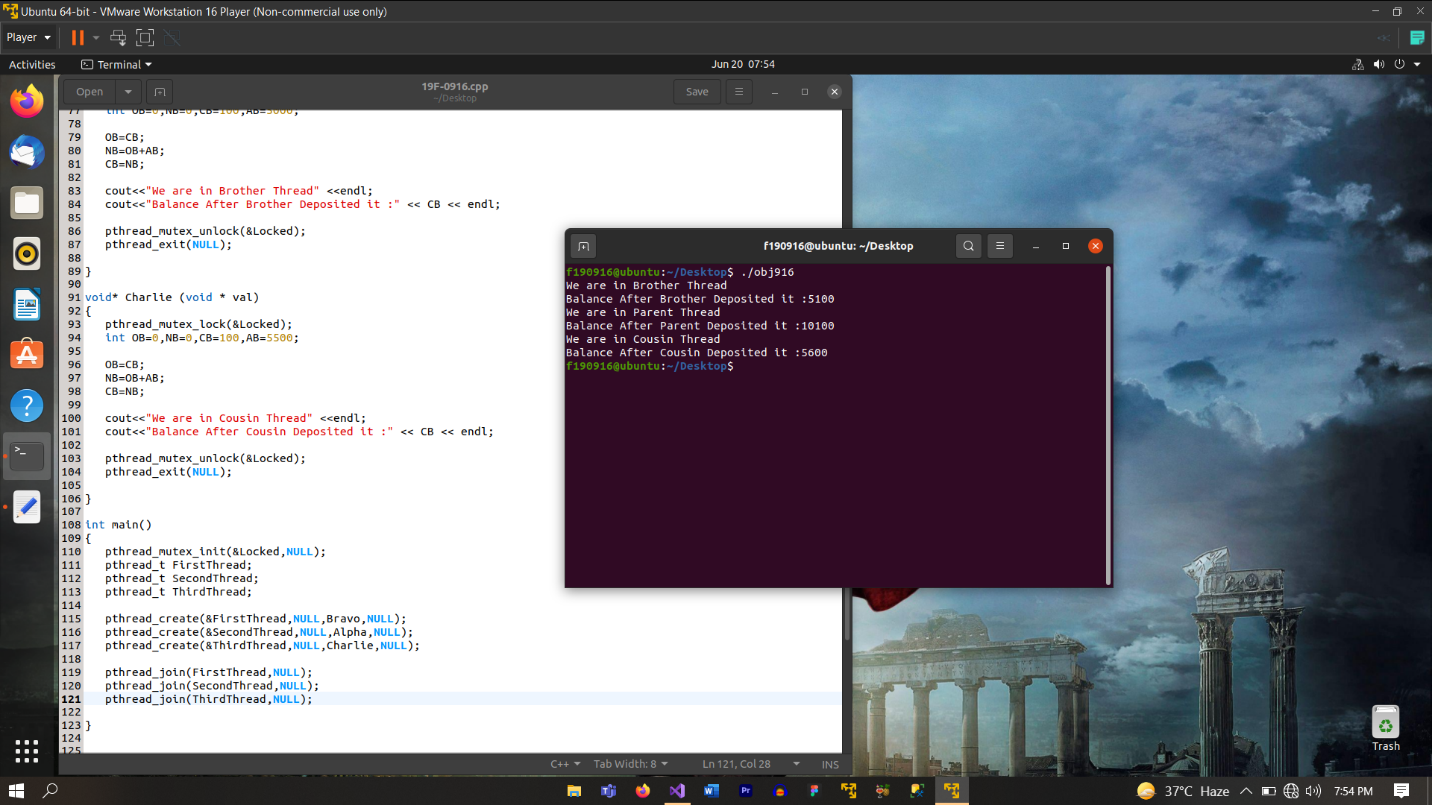
**pthread\_create(&ThirdThread,NULL,Charlie,NULL);**

**pthread\_join(FirstThread,NULL);**

**pthread\_join(SecondThread,NULL);**

**pthread\_join(ThirdThread,NULL);**

**}**

****

**Question # 03**

**Cpp Code:**

**#include <iostream>**

**#include <pthread.h>**

**#include<unistd.h>**

**using namespace std;**

**#include <semaphore.h>**

**sem\_t Semaphore;**

**int GlobalVariable=1;**

**void\* Alpha(void\* val)**

**{**

**sem\_wait(&Semaphore);**

**int temp = GlobalVariable;**

**temp++;**

**GlobalVariable=temp;**

**cout<<" Global Variable in Thread 1 : "<< GlobalVariable << endl;**

**sem\_post(&Semaphore);**

**pthread\_exit(NULL);**

**}**

**void\* Bravo(void\* val)**

**{**

**sem\_wait(&Semaphore);**

**int temp = GlobalVariable;**

**temp--;**

**GlobalVariable=temp;**

**cout<<" Global Variable in Thread 1 : "<< GlobalVariable << endl;**

**sem\_post(&Semaphore);**

**pthread\_exit(NULL);**

**}**

**int main()**

**{**

**sem\_init(&Semaphore,0,1);**

**pthread\_t FirstThread;**

**pthread\_t SecondThread;**

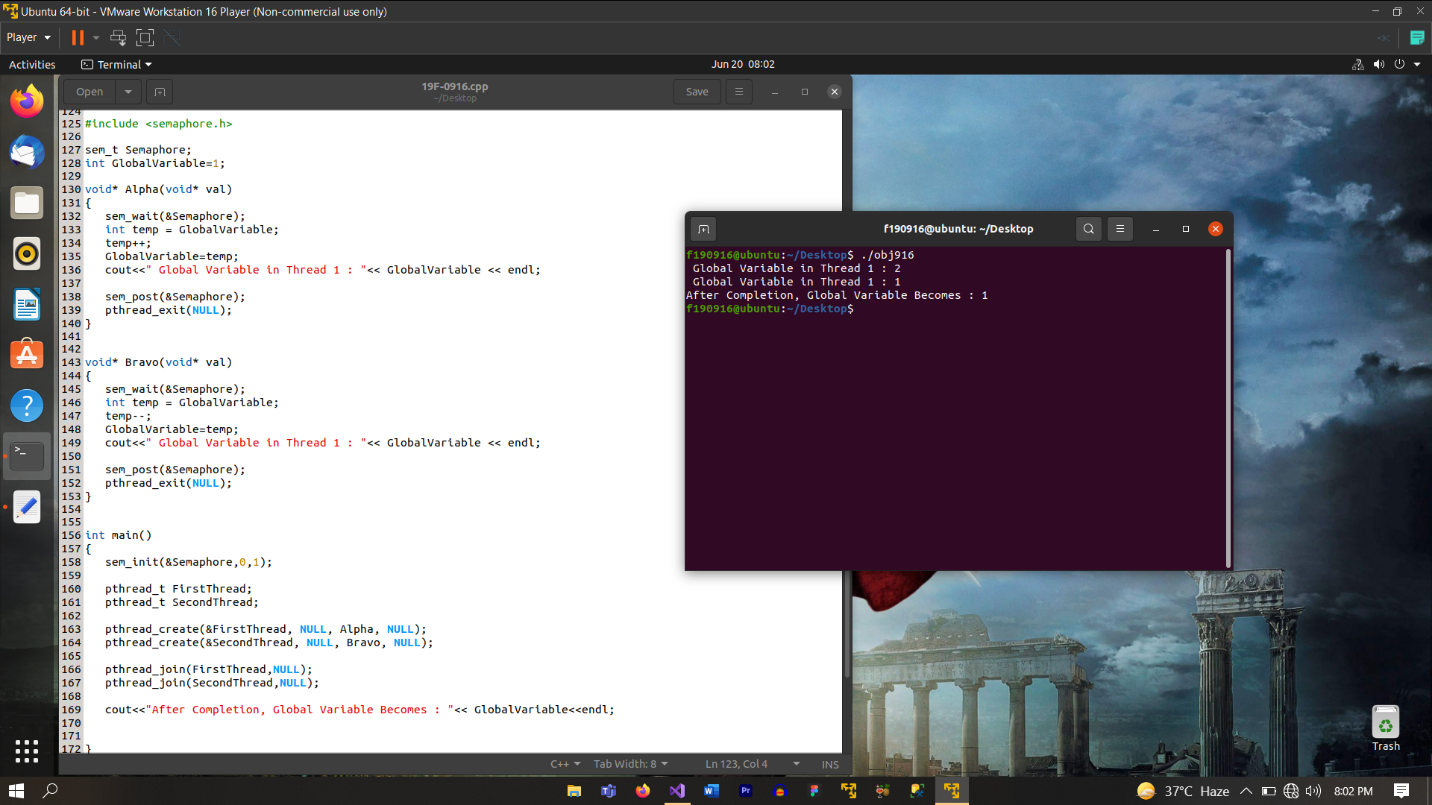
**pthread\_create(&FirstThread, NULL, Alpha, NULL);**

**pthread\_create(&SecondThread, NULL, Bravo, NULL);**

**pthread\_join(FirstThread,NULL);**

**pthread\_join(SecondThread,NULL);**

**cout<<"After Completion, Global Variable Becomes : "<< GlobalVariable<<endl;**

**}**