RACE CONDITION IMPLEMENTATION

#include<pthread.h>

#include<stdio.h>

#include<unistd.h>

void \*inc();

void \*dec();

int shared=1;//shared variable

int main(){

pthread\_t t1,t2;

pthread\_create(&t1, NULL, inc, NULL);//creating threads

pthread\_create(&t2, NULL, dec, NULL);

pthread\_join(t1, NULL);

pthread\_join(t2, NULL);

printf("Final value of shared variable is %d\n",shared);//print the updated value of shared variable

}

void \*inc(){

int x;//local variable

x=shared;

printf("Thread reads the value of shared variable as %d\n",x);

x++;

sleep(5);

shared=x;

printf("Value of shared variable as updated by thread1 is : %d\n",shared);

}

void \*dec(){

int y;//local variable

y=shared;

printf("Thread reads the value of shared variable as %d\n",y);

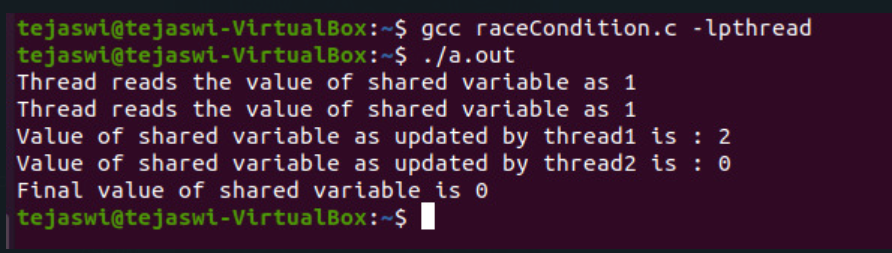
y--;

sleep(5);

shared=y;

printf("Value of shared variable as updated by thread2 is : %d\n",shared);

}



In the above program, inc() function is used to increment the shared variable value and dec() function is used to decrement the shared variabe value.So, initially the value od share variable is one and we increment the value once and decrement it one so finally the value should remain 1.But, we can observe that since we increment and decrement the values locally, they are not updated, creating a race condition and ultimately printing a wrong value of shared variable, i.e. 0.