**ARYAMAN MISHRA**

**19BCE1027**

**LAB 4**

**Program 1:**

**Critical,Master and Single**

#include<stdio.h>

#include<omp.h>

int main(){

int x=0,y=0,z=0;

#pragma omp parallel shared(x) shared(y) shared(z)

{

#pragma omp critical

{

x=x+1;

printf("This is in Critical section value of x is %d and thread no is %d\n",x,omp\_get\_thread\_num());

}

#pragma omp master

{

y=y+1;

printf("This is in Master section value of y is %d and thread no is %d\n",y,omp\_get\_thread\_num());

}

#pragma omp single

{

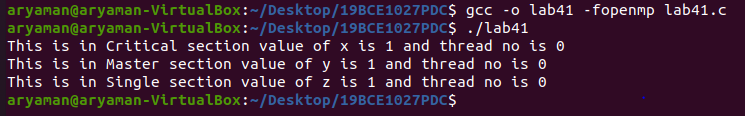
z=z+1;

printf("This is in Single section value of z is %d and thread no is %d\n",z,omp\_get\_thread\_num());

}

}

}



**Program 2:**

**Critical:Producer and Consumer Problem**

#include <omp.h>

#include <stdio.h>

#include <stdlib.h>

int mutex = 1;

int full = 0;

int empty = 10, x = 0;

void producer()

{

--mutex;

++full;

--empty;

x++;

printf("\nProducer produces"

"item %d",

x);

++mutex;

}

void consumer()

{

--mutex;

--full;

++empty;

printf("\nConsumer consumes "

"item %d",

x);

x--;

++mutex;

}

int main()

{

int n, i;

printf("\n1. Press 1 for Producer"

"\n2. Press 2 for Consumer"

"\n3. Press 3 for Exit");

#pragma omp critical

for (i = 1; i > 0; i++) {

printf("\nEnter your choice:");

scanf("%d", &n);

switch (n) {

case 1:

if ((mutex == 1)

&& (empty != 0)) {

producer();

}

else {

printf("Buffer is full!");

}

break;

case 2:

if ((mutex == 1)

&& (full != 0)) {

consumer();

}

else {

printf("Buffer is empty!");

}

break;

case 3:

exit(0);

break;

}

}

} 