Experiment 5

**Aim: To implement programs based on thread synchronization.**

**Assignment 1: Write a Program in OpenMP to demonstrate thread synchronization using barrier clause.**

**Solution:**

**#include <stdio.h>**

**#include <omp.h>**

**void main()**

**{**

***omp\_set\_num\_threads(4);***

**#pragma omp parallel**

**{**

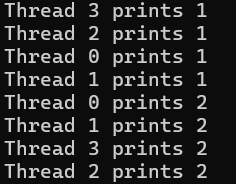
**printf("Thread %d prints 1\n",omp\_get\_thread\_num());**

**# pragma omp barrier**

**printf("Thread %d prints 2\n",omp\_get\_thread\_num());**

**}**

**}**

****

**Assignment 1 (PRACTICE - DO NOT WRITE IN LAB MANUAL): Write a Program in OpenMP to demonstrate thread synchronization without using barrier clause.**

**#include <stdio.h>**

**#include <omp.h>**

**void main()**

**{**

***omp\_set\_num\_threads(4);***

**#pragma omp parallel**

**{**

**printf("Thread %d prints 1\n",omp\_get\_thread\_num());**

**printf("Thread %d prints 2\n",omp\_get\_thread\_num());**

**}**

**}**

***Compilation and execution:***

**gcc -fopenmp barrier\_example.c -o barrier\_example**

**./barrier\_example**

=====================================================================================

**Assignment 2: Write a Program in OpenMP to introduce sequential execution in parallel programming using ordered clause.**

**Solution:**

**#include <stdio.h>**

**#include <omp.h>**

**void main()**

**{**

**#pragma omp parallel for ordered**

**for(int i=1; i<=8; i++)**

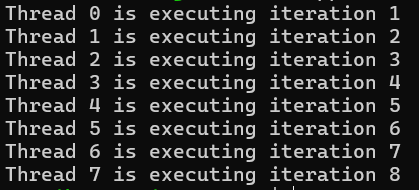
**{**

**#pragma omp ordered**

**printf("Thread %d is executing iteration %d \n",omp\_get\_thread\_num(),i);**

**}**

**}**

****

=====================================================================================

**Assignment 2 (PRACTICE - DO NOT WRITE IN LAB MANUAL): Write a Program in OpenMP without using ordered clause.**

**Solution:**

**#include <stdio.h>**

**#include <omp.h>**

**void main()**

**{**

**#pragma omp parallel for**

**for(int i=1;i<=8;i++)**

**{**

**printf("Thread %d is executing iteration %d\n", omp\_get\_thread\_num(),i);**

**}**

**}**

***Compilation and execution:***

**gcc -fopenmp ordered\_example.c -o ordered\_example**

**./ordered\_example**

**Assignment 3: Write a Program in OpenMP to demonstrate thread synchronization using atomic and critical clause.**

**Solution:**

**#include <stdio.h>**

**#include <omp.h>**

**int main()**

**{**

**int sum\_atomic = 0;**

**int sum\_critical = 0;**

**// Atomic section**

**#pragma omp parallel for**

**for (int i = 0; i < 51; i++) {**

**#pragma omp atomic**

**sum\_atomic += i;**

**}**

**// Critical section**

**#pragma omp parallel for**

**for (int i = 0; i < 51; i++) {**

**#pragma omp critical**

**{**

**for (int j = 0; j < 100; j++) {**

**sum\_critical += i + j;**

**}**

**}**

**}**

**printf("Sum using atomic: %d\n", sum\_atomic);**

**printf("Sum using critical: %d\n", sum\_critical);**

**return 0;**

**}**

****