**HPC Assignment No. 04**

**Batch:** B2

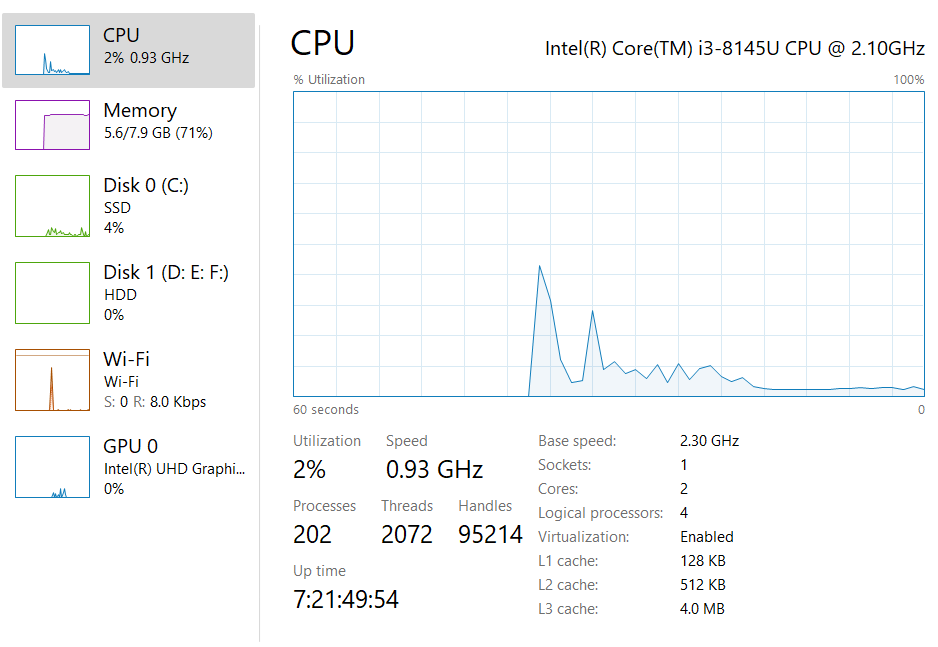
**PRN No:** 2019BTECS00033

**Name**: Teknath jha

**Date**: 12th of sept 2022

* **Title:**

**MY SYSTEM CONFIGURATION :**



1. Analyse and implement a Parallel code for below programs using OpenMP considering synchronization requirements. (Demonstrate the use of different clauses and constructs wherever applicable)

*/\**

*Author - Teknath jha*

*PRN    - 2019BTECS00033*

*HPC-Lab- 4*

*Q1: Analyse and implement a Parallel code for below programs using OpenMP*

*considering synchronization requirements. (Demonstrate the use of different*

*clauses and constructs wherever applicable).*

*\*/*

*// PARALLEL*

#include <omp.h>

#include <stdio.h>

#include <stdlib.h>

int fib(int n) {

  int i, j;

  if (n < 2)

    return n;

  else {

#pragma omp task shared(i)

    i = fib(n - 1);

#pragma omp task shared(j)

    j = fib(n - 2);

#pragma omp taskwait

    return i + j;

  }

}

int main() {

  int n = 6, result;

*//  scanf("%d",n);*

#pragma omp parallel

  {

#pragma omp single

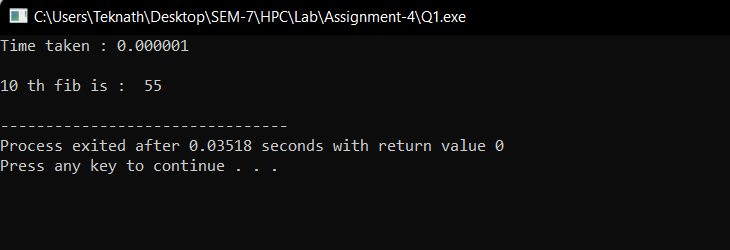
    result = fib(n);

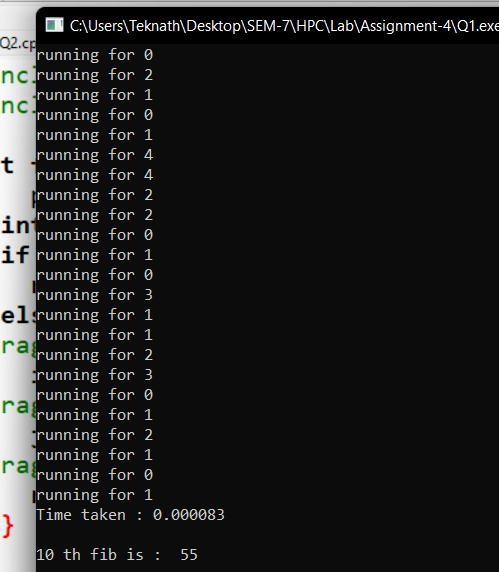
  }

  printf("Nth fib is :  %d\n", result);

}

**Output :**

****

****

**Q2)** Analyse and implement a Parallel code for below programs using OpenMP considering synchronization requirements. (Demonstrate the use of different clauses and constructs wherever applicable). Producer Consumer Problem:

*/\**

*Author - Teknath jha*

*PRN    - 2019BTECS00033*

*HPC-Lab- 4*

*Q2: Analyse and implement a Parallel code for below programs using OpenMP*

*considering synchronization requirements. (Demonstrate the use of different*

*clauses and constructs wherever applicable). Producer Consumer Problem:*

*\*/*

#include <stdio.h>

#include <stdlib.h>

int mutex = 1, full = 0, 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);

    if (n == 1) {

      if ((mutex == 1) && (empty != 0)) {

        producer();

      }

      else {

        printf("Buffer is full!");

      }

      if (n == 2) {

        if ((mutex == 1) && (full != 0)) {

          consumer();

        }

        else {

          printf("Buffer is empty!");

        }

      }

      else if (n == 3) {

        exit(0);

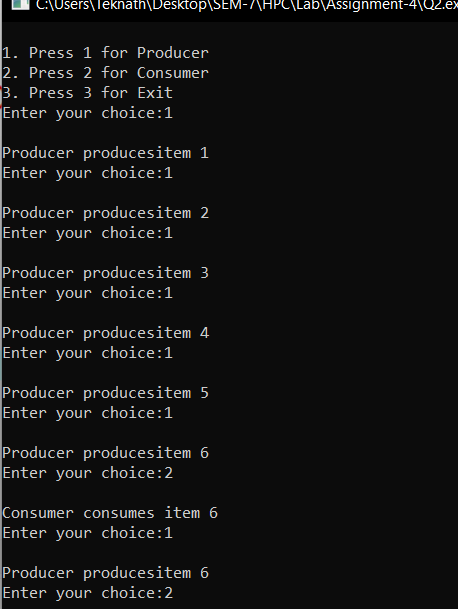
      }

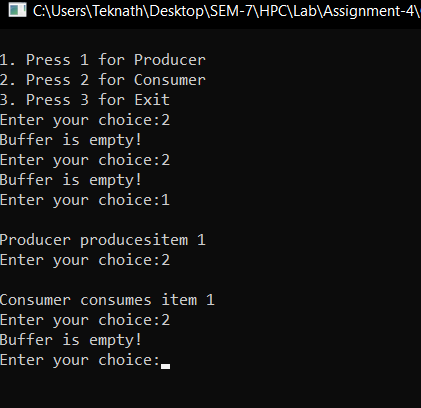
    }

  }

}

**Output:**

****

****