|  |  |
| --- | --- |
| **Name** | Manish Shashikant Jadhav |
| **UID no.** | 2023301005 |

|  |  |
| --- | --- |
| **Experiment 4A** | |
| **AIM :** | Write a multithreaded program that calculates various statistical values for a list of numbers. This program will be passed a series of numbers on the command line and will then create three separate worker threads. One thread will determine the average of the numbers, the second will determine the maximum value, and the third will determine the median value. |
| **Discussion & Output:** | **Program:**  #include <stdio.h>  #include <pthread.h>  #define MAX\_SIZE 50  int arr[MAX\_SIZE], size, index;  void \*calculateAvg()  {    float sum = 0;    float average;    printf("What will be the size of an array? ");    scanf("%d", &size);    printf("Please enter %d elements:\n", size);    for (index = 0; index < size; index++)    {      scanf("%d", &arr[index]);    }    for (index = 0; index < size; index++)    {      sum = sum + arr[index];    }    average = sum / size;    printf("The average of given elements in an array is: %.2f\n", average);    return NULL;  }  void swap(int \*x, int \*y)  {    int temp = \*x;    \*x = \*y;    \*y = temp;  }  void \*calculateMid()  {    for (index = 0; index < size - 1; index++)    {      int min = index;      for (int j = index + 1; j < size; j++)      {        if (arr[j] < arr[min])          min = j;      }      swap(&arr[min], &arr[index]);    }    float median;    if (size % 2 == 0)    {      median = (arr[(size - 1) / 2] + arr[(size - 1) / 2 + 1]) / 2.0;    }    else    {      median = arr[size / 2];    }    printf("The median of given elements in an array is: %.2f\n", median);    return NULL;  }  void \*calculateMaxi()  {    int maxVal = arr[0];    for (int i = 1; i < size; i++)    {      if (maxVal < arr[i])      {        maxVal = arr[i];      }    }    printf("The maximum of given elements in an array is:: %d\n", maxVal);    return NULL;  }  int main()  {    pthread\_t thread1, thread2, thread3;    pthread\_create(&thread1, NULL, calculateAvg, NULL);    pthread\_join(thread1, NULL);    pthread\_create(&thread2, NULL, calculateMid, NULL);    pthread\_join(thread2, NULL);    pthread\_create(&thread3, NULL, calculateMaxi, NULL);    pthread\_join(thread3, NULL);    return 0;  }  **Output:** |
| **CONCLUSION:** | Hence, by completing this experiment I understood how to write a multithreaded program that calculates various statistical values for a list of numbers. Program will be passed a series of numbers on the command line and will then create three separate worker threads. One thread will determine the average of the numbers, second will determine the maximum value, and the third will determine the median value. |

|  |  |
| --- | --- |
| **Experiment 4B** | |
| **AIM :** | Write a multithreaded program that creates two functions one for adding the element from array and other function to delete last two elements in array and print it. The code runs 5 times, addition of elements happens first and then the deletion. Check necessary conditions like minimum 2 elements remain in array. |
| **Discussion & Output:** | **Program:**  #include <stdio.h>  #include <pthread.h>  #define MAX\_SIZE 8  int array[MAX\_SIZE];  int size = 0;  pthread\_mutex\_t mutex = PTHREAD\_MUTEX\_INITIALIZER;  void \*addValue(void \*arg)  {    int num = \*(int \*)arg;    pthread\_mutex\_lock(&mutex);    if (size < MAX\_SIZE)    {      array[size++] = num;      printf("New Value %d is added to the array. Hence the New array is: ", num);      for (int i = 0; i < size; i++)      {        printf("%d ", array[i]);      }      printf("\n");    }    else    {      printf("Cannot add %d. Array is full.\n", num);    }    pthread\_mutex\_unlock(&mutex);    return NULL;  }  void \*deleteValue(void \*arg)  {    pthread\_mutex\_lock(&mutex);    if (size >= 2)    {      int lastElement1 = array[size - 1];      int lastElement2 = array[size - 2];      size -= 2;      printf("Deleted %d and %d from the array. New array is: ", lastElement1,             lastElement2);      for (int i = 0; i < size; i++)      {        printf("%d ", array[i]);      }      printf("\n");    }    else    {      printf("As there are only 2 elements left in array. Therefore cannot perform delete operation.\n");    }    pthread\_mutex\_unlock(&mutex);    return NULL;  }  int main()  {    pthread\_t addThread, deleteThread;    int numbers[MAX\_SIZE];    printf("Enter up to %d integers: \n", MAX\_SIZE);    int number, index = 0;    while (index < MAX\_SIZE && scanf("%d", &number) == 1 && number != -1)    {      numbers[index++] = number;    }    for (int i = 0; i < index; i++)    {      pthread\_create(&addThread, NULL, addValue, &numbers[i]);      pthread\_join(addThread, NULL);      pthread\_create(&deleteThread, NULL, deleteValue, NULL);      pthread\_join(deleteThread, NULL);    }    pthread\_mutex\_destroy(&mutex);    return 0;  }  **Output:** |
| **CONCLUSION:** | Hence, by completing this experiment I came to know about how to write a multithreaded program that creates two functions one for adding the element from array and other function to delete last two elements in array and print it. The code runs 5 times, addition of elements happens first and then the deletion. Check necessary conditions like minimum 2 elements remain in array. |