

Tema 3 – OpenMp: Calculul paralel a histogramei unei imagini

1. Compilați și executați programul de mai jos.

```
#include <iostream>
#include <omp.h>
#include <time.h>

void main()
{
    // Alocarea de memorie pentru imagine
    int imgSize = 10000;
    unsigned char *img = new unsigned char[imgSize*imgSize];

    // Alocare de memorie pentru histogramă
    int *hist = new int[256];

    // Inițializarea imaginii
    for (int i = 0; i < imgSize; i++)
        for (int j = 0; j < imgSize; j++)
            img[i*imgSize + j] = i%256;

    // Inițializarea histogramei
    for (int i = 0; i < 256; i++)
        hist[i] = 0;

    clock_t start = clock();

    // Calculul histogramei
    for (int i = 0; i < imgSize; i++)
        for (int j = 0; j < imgSize; j++)
            hist[img[i*imgSize + j]]++;

    clock_t stop = clock();

    // Afișarea histogramei
    for (int i = 0; i < 256; i++)
        std::cout << hist[i] << " ";
    std::cout << std::endl;

    // Afișarea timpului de execuție
    std::cout << "Time: " << double(stop - start) / CLOCKS_PER_SEC << std::endl;

    // Dealocarea memoriei
    delete img;
    delete hist;

    system("PAUSE");
}
```

2. Paralelizați calculul histogramei din programul de mai sus utilizând OpenMP.

Indicații:

- Se alocă memorie și se inițializează câte o histogramă „privată” fiecărui fir de execuție
- Se folosește directiva *omp parallel for* pentru a asocia fiecărui fir de execuție o porțiune din imagine
- Fiecare fir de execuție va calcula histograma porțiunii de imagine ce-i corespunde
- În final se calculează histograma globală însumând histogramele „private” calculate de fiecare fir de execuție