LABWORK 1

OPEN MP

TRAN Quy Ban
Department of ICT

INTRODUCTION

The Labwork 1 converts sequential CPU Code into parallel using OpenMP. The speed is increased a lot.

RESULT

USTH ICT Master 2018, Advanced Programming for HPC. Warming up... Starting labwork 1 labwork 1 CPU ellapsed 4061.6ms 1 3756.82 2233.33 1266.54 980.15 798.66 675.17 622.28 533.49 474.010 429.111 395.212 367.913 347.614 336.915 302.716 436.417 499.218 465.919 440.920 416.821 399.522 380.723 364.724 349.925 336.926 323.927 321.028 302.429 293.330 286.031 277.032 270.733 428.634 415.235 421.336 414.737 417.038 417.139 420.140 418.841 414.042 414.343 410.444 408.045 403.546 395.947 385.948 382.349 375.750 370.751 364.752 367.453 364.354 355.455 363.056 358.057 365.558 365.059 371.060 368.861 373.162 373.063 374.9labwork 1 ellapsed 483.4ms

SOURCE CODES

```
void Labwork::labwork1_OpenMP(int threads) {
    int pixelCount = inputImage->width * inputImage->height;
    outputImage = static_cast<char *>(malloc(pixelCount * 3));
    omp_set_num_threads(threads);
    for (int j = 0; j < 100; j++) { // let’s do it 100 times, otherwise it’s too fast
        # pragma omp parallel for
        for (int i = 0; i < pixelCount; i++) {
            outputImage[i * 3] = (char) (((int) inputImage->buffer[i * 3] +
                                          (int) inputImage->buffer[i * 3 + 1] +
                                          (int) inputImage->buffer[i * 3 + 2]) / 3);
            outputImage[i * 3 + 1] = outputImage[i * 3];
            outputImage[i * 3 + 2] = outputImage[i * 3];
        }
    }
}
```