T4: Geração de Fractais de Mandelbrot em OpenMP

Eduardo Rafael Hirt

Possibilidades de paralelização

```
// compute frames
double delta = Delta;
for (int frame = 0; frame < frames; frame++) {
  const double xMin = xMid - delta;
  const double yMin = yMid - delta;
  const double dw = 2.0 * delta / width;
```

- Iteração deve ser sequencial devido ao seguinte trecho:

```
    delta *= 0.98;
```

Possibilidades de paralelização

```
for (int row = 0; row < width; row++) {
  const double cy = yMin + row * dw;
```

Possibilidades de paralelização

```
for (int col = 0; col < width; col++) {
  const double cx = xMin + col * dw;
  double x = cx;
  double y = cy;
  int depth = 256;
  double x2, y2;
```

Definição da região crítica:

```
//set critical region
#pragma omp critical
{
  pic[frame * width * width + row * width + col] = (unsigned char)depth;
}
```

Estratégia 1: paralelização nas linhas

```
//start the parallel execution
#pragma omp parallel for schedule(auto)
for (int row = 0; row < width; row++) {
  const double cy = yMin + row * dw;
  for (int col = 0; col < width; col++) {
    const double cx = xMin + col * dw;
    double x = cx;
    double y = cy;
    int depth = 256;
    double x2, y2;
    do {
      x2 = x * x;
      y2 = y * y;
      y = 2 * x * y + cy;
      x = x2 - y2 + cx;
```

Estrategia 1 - resultados:

	Fractal	**F1 – 2** threads	**F1 – 4 threads**
512 32	11.4818s	6.2799s	4.3413s
512 64	21.9186s	11.8888s	8.5013s
1024 32	47.6827s	25.0129s	16.3605s
1024 64	88.9103s	47.1654s	33.7022s

Estratégia 2: paralelização das colunas

```
for (int row = 0; row < width; row++) {
  const double cy = yMin + row * dw;

  //start the parallel execution
  #pragma omp parallel for schedule(auto)
  for (int col = 0; col < width; col++) {
    const double cx = xMin + col * dw;
    double x = cx;
    double y = cy;
    int depth = 256;
    double x2, y2;
    do {
      x2 = x * x;
      y2 = y * y;
      y = 2 * x * y + cy;
      x = x2 - y2 + cx;
```

Estrategia 2 - resultados:

	Fractal	F2 – 2 threads	F2 – 4 threads
512 32	11.4818s	7.7123s	6.6959s
512 64	21.9186s	14.5267s	11.3634s
1024 32	47.6827s	29.0454s	24.5017s
1024 64	88.9103s	54.8704s	44.6627s