

1. Implementación serial

```
Semana 11$ g++ chrono.cc primosSerial.cpp -w
rigovil@rodrigo:~/Escritorio/UCR/I Semestre 2020/CI-0117$ ./a.out
Ingrese el limite superior:
10
6 = 3 + 3
8 = 3 + 5
10 = 3 + 7
Tiempo: 0 seg, 431311707 nseg
```

Se prueba el programa con la medición de tiempo implementada en la clase `Chrono` y se utiliza un n pequeño para comprobar la correcta funcionalidad del programa.

2. Implementación Pthreads

```
Semana 11$ g++ chrono.cc primosPThread.cpp -pthread -w
rigovil@rodrigo:~/Escritorio/UCR/I Semestre 2020/CI-0117$ ./a.out
Ingrese el limite superior:
10
6 = 3 + 3
8 = 3 + 5
10 = 3 + 7
Tiempo: 0 seg, 374560329 nseg
```

Se prueba el programa pero ahora utilizando la implementación con `Pthreads`.

3. Implementación OpenMP

```
Semana 11$ g++ chrono.cc primosOpenMP.cpp -fopenmp -w
rigovil@rodrigo:~/Escritorio/UCR/I Semestre 2020/CI-0117$ ./a.out
Ingrese el limite superior:
10
6 = 3 + 3
8 = 3 + 5
10 = 3 + 7
Tiempo: 0 seg, 236619528 nseg
```

Por último, se corre el programa con la implementación mediante `OpenMP` para comprobar su funcionalidad.

4. Pruebas

Se muestran distintas pruebas utilizando las tres diferentes implementaciones. Se escogió un $n = 100000$ como límite superior y una cantidad de 2^k trabajadores. Se muestran algunas pruebas para llenar la tabla, en las otras solamente se indicará el tiempo de duración. El programa no imprime la solución para cada número par para medir el tiempo de duración real del cálculo.

```
rigovil@rodrigo:~/Escritorio/UCR/I Semestre 2020/CI-0117$ ./a.out
Ingrese el limite superior:
100000
Tiempo: 18 seg, 828946705 nseg
```

Prueba utilizando la implementación serial

```

rigovll@rodrigo:~/Escritorio/UCR/I
Semana 11$ ./a.out
Ingrese el limite superior:
100000
Tiempo: 14 seg, 201462039 nseg

```

Implementación Pthread y 2 trabajadores

```

rigovll@rodrigo:~/Escritorio/UCR/I
Semana 11$ ./a.out
Ingrese el limite superior:
100000
Tiempo: 8 seg, 925839108 nseg

```

Implementación Pthread y 4 trabajadores

```

rigovll@rodrigo:~/Escritorio/UCR/I
Semana 11$ ./a.out
Ingrese el limite superior:
100000
Tiempo: 5 seg, 959709910 nseg

```

Implementación Pthread y 8 trabajadores

```

rigovll@rodrigo:~/Escritorio/UCR/I
Semana 11$ ./a.out
Ingrese el limite superior:
100000
Tiempo: 9 seg, 931235245 nseg

```

Implementación OpenMP y 2 trabajadores

```

rigovll@rodrigo:~/Escritorio/UCR/I
Semana 11$ ./a.out
Ingrese el limite superior:
100000
Tiempo: 6 seg, 737256060 nseg

```

Implementación OpenMP y 4 trabajadores

```

rigovll@rodrigo:~/Escritorio/UCR/I
Semana 11$ ./a.out
Ingrese el limite superior:
100000
Tiempo: 4 seg, 926489890 nseg

```

Implementación OpenMP y 8 trabajadores

5. Tabla comparativa y conclusiones

Serial	Pthreads	OpenMP	Trabajadores
18s, 514935138ns	14s, 542521643ns	9s, 758637697ns	2
18s, 514935138ns	8s, 526936582ns	5s, 956124787ns	4
18s, 514935138ns	5s, 898312308ns	5s, 727516976ns	8
18s, 514935138ns	5s, 778442800ns	5s, 680402289ns	16
18s, 514935138ns	5s, 764609126ns	5s, 808271955ns	32
18s, 514935138ns	5s, 552529213ns	5s, 746974931ns	64

Se evidencia como una implementación en paralelo es considerablemente mejor que la implementación serial. La comparativa entre Pthreads y OpenMP indica que este último es considerablemente mejor con una cantidad de trabajadores pequeña, a partir de 2^3 trabajadores, las diferencias varían muy poco entre las dos distintas implementaciones.