

Práctica Pthreads2

En esta segunda práctica de pthreads se utilizan locks y técnicas para coordinar los threads productores y consumidores, además de gestionar las secciones críticas del programa mediante exclusión mutua. En las 6 versiones del programa histograma se utilizan 3 imágenes con diferente resolución y calidad.

Procedemos a explicar las versiones:

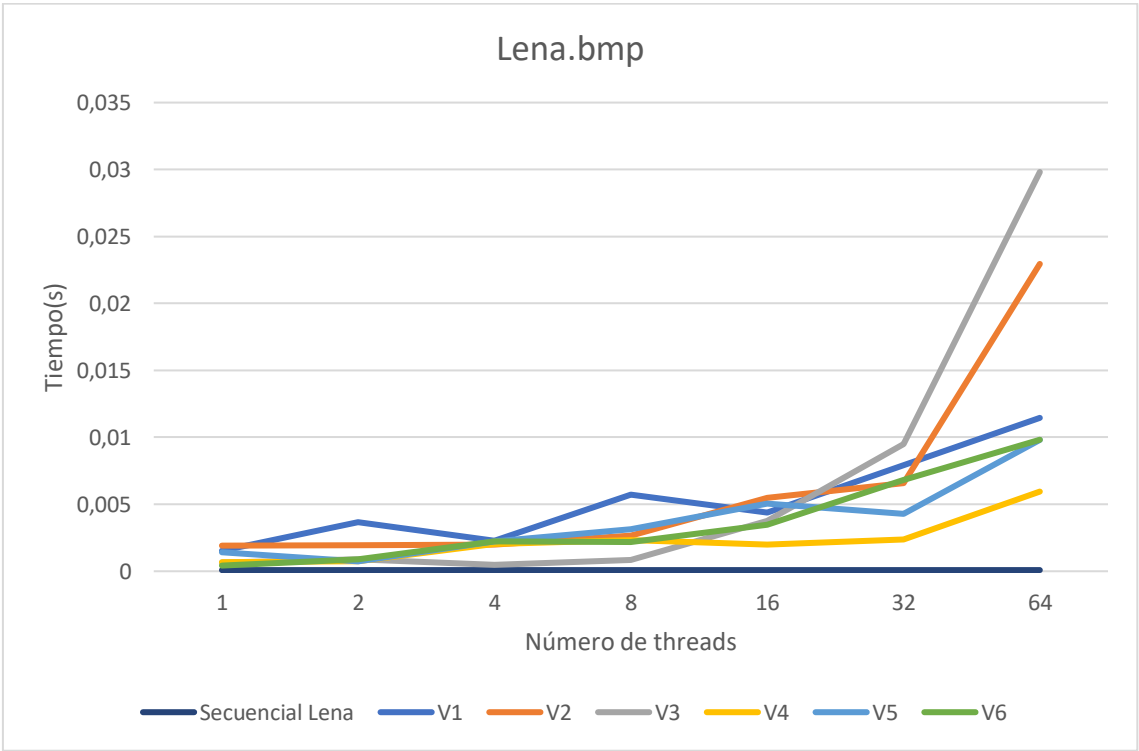
- Versión 1: Los threads que hemos pasado por argumentos deben acceder en exclusión mutua al array de histograma, cada thread ejecuta un bloque del bucle que utilizamos para recorrer la imagen pero el cálculo principal lo realiza el master.
- Versión 2: Misma versión que el anterior pero utilizando 256 mutex en vez de solo 1.
- Versión 3: Cada thread calcula una parte parcial del array iHist, utilizando una estructura de datos compartida. El cálculo final lo vuelve a calcular el master.
- Versión 4: Se realiza el cálculo de iHist en paralelo, por lo que se tiene que garantizar la terminación de hist primero, utilizando barreras.
- Versión 5: Repartimos iHist entre los threads de forma paralela y después calculamos hist.
- Versión 6: Para resolver el desbalanceo de carga de la versión anterior, se implementa un particionado estático entrelazado de iHist.

A continuación se ofrecen los resultados obtenidos (en segundos):

Lena.bmp

Tiempos	1	2	4	8	16	32	64
V1	0,001531	0,003683	0,002284	0,005733	0,004396	0,00794	0,011453
V2	0,001913	0,00193	0,001989	0,002681	0,005478	0,00658	0,02295
V3	0,000674	0,000875	0,000478	0,00082	0,003777	0,009478	0,029813
V4	0,000674	0,000731	0,00206	0,002308	0,002001	0,002395	0,005949
V5	0,001418	0,000737	0,002223	0,00315	0,005037	0,004286	0,009808
V6	0,000422	0,000911	0,002219	0,002194	0,003489	0,006826	0,009819

Secuencial Lena	0,0000844
-----------------	-----------

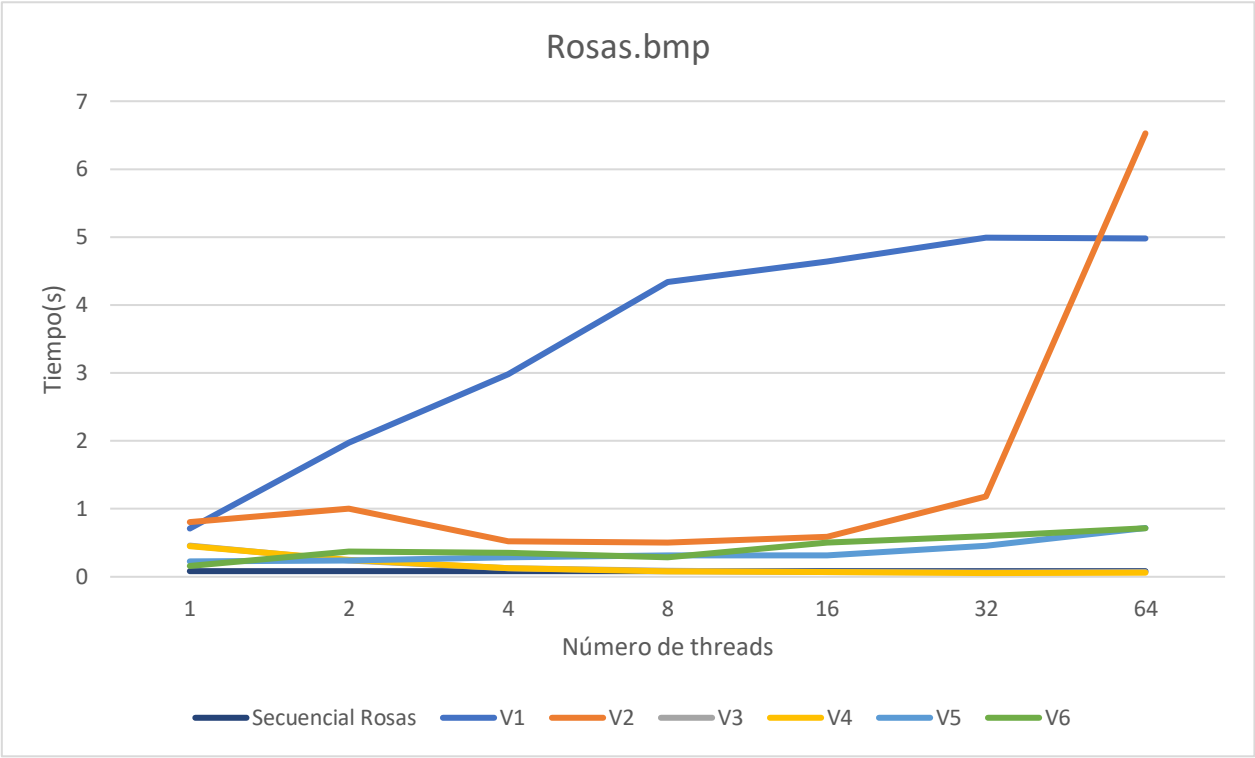


Speedup	1	2	4	8	16	32	64
V1	0,05512737	0,0229161	0,03695271	0,01472179	0,01919927	0,01062972	0,00736925
V2	0,04411918	0,04373057	0,04243338	0,03148079	0,01540708	0,01282675	0,00367756
V3	0,12522255	0,09645714	0,17656904	0,10292683	0,02234578	0,00890483	0,00283098
V4	0,12522255	0,11545828	0,04097087	0,03656846	0,04217891	0,03524008	0,01418726
V5	0,05952045	0,11451832	0,03796671	0,02679365	0,01675601	0,01969202	0,00860522
V6	0,2	0,09264544	0,03803515	0,03846855	0,02419031	0,01236449	0,00859558

Rosas.bmp

Tiempos	1	2	4	8	16	32	64
V1	0,711001	1,973911	2,984769	4,339064	4,644371	4,991872	4,980349
V2	0,807112	1,00735	0,518802	0,503956	0,585304	1,183497	6,527139
V3	0,459365	0,239517	0,125447	0,092116	0,072147	0,065795	0,066144
V4	0,451032	0,245969	0,127362	0,083843	0,069763	0,055074	0,062182
V5	0,226979	0,236347	0,289243	0,313897	0,314078	0,459725	0,718379
V6	0,160079	0,36859	0,351337	0,291568	0,507966	0,596379	0,714795

Secuencial Rosas	0,083287
------------------	----------

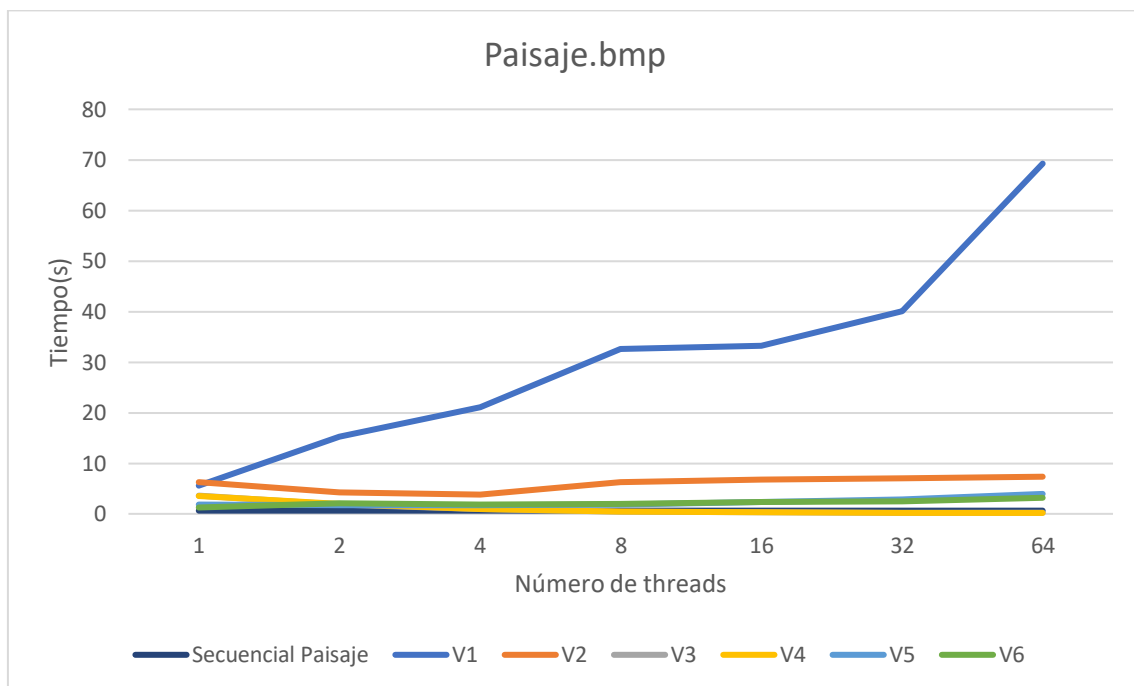


Speedup	1	2	4	8	16	32	64
V1	0,11714048	0,0421939	0,027904	0,01919469	0,01793289	0,01668452	0,01672313
V2	0,10319138	0,08267931	0,16053716	0,16526641	0,14229699	0,07037365	0,01276011
V3	0,18130898	0,34772897	0,66392182	0,90415346	1,15440697	1,26585607	1,25917695
V4	0,18465874	0,33860771	0,65393917	0,99336856	1,19385634	1,51227439	1,3394069
V5	0,36693703	0,35239288	0,2879482	0,26533226	0,26517935	0,181167	0,11593741
V6	0,52028686	0,22596109	0,2370573	0,28565206	0,16396176	0,13965448	0,11651872

Paisaje.bmp

Tiempos	1	2	4	8	16	32	64
V1	5,576608	15,279794	21,024611	32,636114	33,225095	40,06039	69,261626
V2	6,324132	4,236186	3,800121	6,306305	6,77815	7,008112	7,340352
V3	3,599949	1,886005	0,943416	0,470595	0,359828	0,248297	0,221294
V4	3,524017	1,974877	0,941897	0,475008	0,339192	0,243739	0,177937
V5	1,819728	1,770295	1,709427	1,884212	2,349064	2,831389	3,941191
V6	1,256154	2,035857	1,867555	1,914625	2,292981	2,521338	3,176316

Secuencial Paisaje	0,6301704
---------------------------	-----------



Speedup	1	2	4	8	16	32	64
V1	0,11300246	0,04124207	0,02997299	0,01930899	0,0189667	0,01573051	0,00909841
V2	0,09964536	0,14875891	0,16582904	0,09992704	0,09297085	0,08992014	0,08585016
V3	0,17504981	0,33412976	0,66796662	1,33909285	1,75131007	2,53797025	2,84766148
V4	0,17882161	0,31909349	0,66904386	1,32665218	1,8578575	2,58543114	3,54153661
V5	0,34629923	0,35596915	0,36864423	0,33444772	0,26826447	0,22256581	0,15989339
V6	0,50166652	0,30953569	0,3374307	0,32913516	0,27482583	0,24993492	0,19839663

Conclusiones

Para la imagen "Lena.bmp" ninguna de las versiones es capaz de alcanzar a la versión secuencial del programa. Esto puede ser debido a que es una imagen muy pequeña en la que no se puede aprovechar el paralelismo ya que el cómputo necesario para crear este paralelismo es mayor que el cómputo del histograma en sí. En cualquier caso, de todas las versiones paralelizadas la mejor es la V4.

Utilizando la imagen "Rosas.bmp" obtenemos resultados dispares en las cuales seguimos teniendo una versión secuencial mejor que el resto de versiones, salvo en las versiones V3 y V4, las cuales empiezan a destacar a partir de los 8 threads, obteniendo una velocidad incluso mayor a la del programa secuencial.

En la imagen "Paisaje.bmp" pasa exactamente lo mismo que en el caso anterior, y es que las versiones V3 y V4 consiguen ejecutar el programa de una manera más rápida, consiguiendo un speedup incluso mayor al de la imagen "Rosas.bmp".

Como conclusión general podemos decir que las versiones V3 y V4 son las más optimizadas para nuestras imágenes, de las cuales solo podemos obtener una mejora de rendimiento en el caso de "Rosas.bmp" y "Paisaje.bmp" a partir de los 8 threads en el primer caso, y en algún momento entre 4 y 8 threads para el segundo.

La versión 4 del programa es la que mejor escala cuando la imagen es mayor y más compleja, por lo que podríamos concluir que para imágenes muy pequeñas, la mejor versión es la secuencial, pero en cuanto esta imagen empieza a requerir mayor cómputo, la versión 4 ofrece una mejora de rendimiento muy interesante.