

Universidad Nacional Autónoma de México
Facultad de Ciencias

Cómputo Concurrente

Profesora: Gilde Valeria Rodríguez Jiménez

Ayudantes: Luis Angel Leyva Castillo

Rogelio Alcantar Arenas

Gibrán Aguilar Zuñiga

Cuestionario Practica 5: Spinlocks

Carlos Daniel Cortés Jiménez

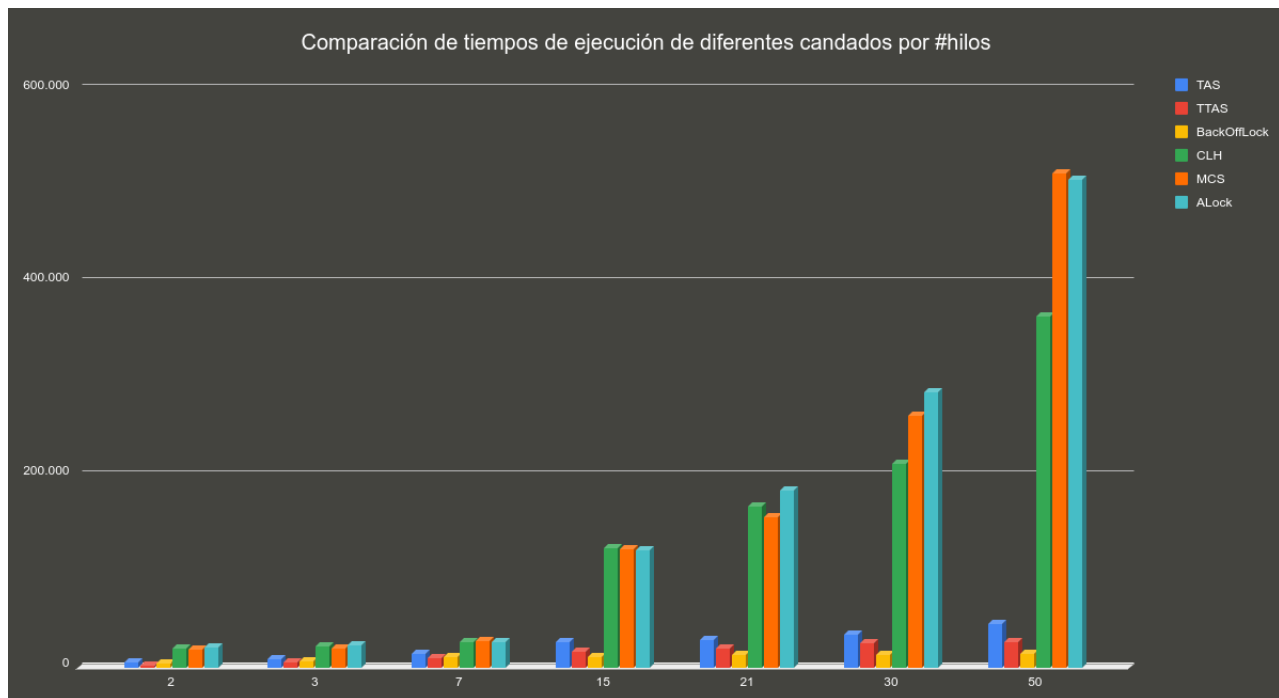
10 de Abril 2024

1. Tabla comparativa de resultados

1.1. Tabla de resultados

	TAS	TTAS	BackOffLock	CLH	MCS	ALock
2	5.078	2.090	4.132	19.994	18.597	20.993
3	8.785	5.020	6.226	21.743	19.373	22.569
7	14.123	10.002	10.703	26.945	27.169	26.810
15	26.828	16.379	11.008	123.835	122.635	121.123
21	29.027	20.096	12.886	167.048	155.717	183.840
30	33.800	25.208	12.901	211.337	261.125	285.661
50	45.730	26.204	13.754	364.073	512.760	505.620

1.2. Gráfica comparativa



1.3. Especificaciones de la computadora

Modelo de hardware	Dell Inc. G7 7588
Memoria	16.0 GiB
Procesador	Intel® Core™ i7-8750H CPU @ 2.20GHz × 12
Gráficos	NV137 / Mesa Intel® UHD Graphics 630 (CFL GT2)
Capacidad del disco	1.1 TB
Nombre del SO	Ubuntu 22.04.4 LTS
Tipo de SO	64 bits

```
Nombre del modelo: Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz
Familia de CPU: 6
Modelo: 158
Hilo(s) de procesamiento por núcleo: 2
Núcleo(s) por «socket»: 6
«Socket(s)»: 1
Revisión: 10
CPU MHz máx.: 4100.0000
CPU MHz mín.: 800.0000
BogoMIPS: 4399.99
```

```
*-core
  descripción: Placa base
  producto: 075F7T
  fabricante: Dell Inc.
  id físico: 0
  versión: A00
  serie: /9J96LR2/CNMC00892015A/
```

Coprocesador de gráficos	NVIDIA GeForce GTX 1060
Marca de la tarjeta gráfica	NVIDIA
Descripción de la tarjeta gráfica	nvidia geforce gtx

2. Cuestionario

1. ¿Para que sirve el método Yield? (yield())

El metodo yield() sirve para poder ceder el procesador cuando otros hilos que tienen mas prioridad puedan hacer uso de este, y puedan terminar su ejecucion, especialmente los usamos cuando se tiene varios hilos que este tratando de acceder a los recursos.

2. ¿Qué es un atributo atómico? (En la biblioteca atomic de Java)

Un atributo atomico de java es una variable que nos permite realizar operaciones seguras sin la interferencia de otros hilos y además sin que necesitemos el uso de candados para la sincronizacion entre hilos.

3. Ventajas de usar atributos atomicos

- Podemos evitar la posibilidad de que haya condiciones de carrera
- Podemos evitar el tener que usar candados explicitos para sincronizar a los atributos atomicos
- Hacen que el rendimiento mejore
- las operaciones que se realizan se de manera seguro cuando hay muchos hilos

4. Desventajas de usar atributos atómicos

- Si la complejidad del problema es muy grande, el uso de atributos atomicos puede que no se tan seguro y nos haga requerir del uso de candados
- Puede haber la posibilidad que nos encontremos con situaciones donde haya deadlock
- El uso de atributos atomicos puede hacer que al momento de realizar cambios en el codigo haga que sea mas dificil de entender

5. ¿Que locks cumplen con la propiedad de Justicia?

- CLHLock: Notemos que se esta implementando una cola de espera con nodos, esto puede garantizar que los hilos se ejecuten en el orden en que llegaron para poder adquirir el candado
- MCSLock: asi como con CLHLock, se implemeta una cola de espera con nodos, a cada hilo se le dará la oportunidad de adquirir el candado.

6. ¿Qué locks cumplen con la propiedad libre de Hambruna (starvation-free)?

- CLHLock: Dado que se implemeta una cola de espera, cada hilo tendra la oportunidad de adquirir el candado eventualmente.
- MCSLock: Al igual que en CLHLock, como se esta implementando una cola de espera, a cada hilo se le dará la oportunidad de adquirir el el candado eventualmente.

7. ¿Cuál es la implementación más eficiente? ¿Porqué crees que es así?

Apesar de que en las pruebas los tiempos de eejcucion el que tuve tiempos mas bajos fue Backofflock, considero que el mejor podria ser MSCLock, debido a que se esta usando una cola de espera pero con menos operaciones de escritura en la memoria compartida, esto pueden tener una mejor escalabilidad en sistemas donde se tengas una alta concurrencia así como reducir la contención en la memoria compartida cuando tenemos un sistema con múltiples procesadores.

8. Por último, describe un problema en el que se pueda utilizar un candado visto en esta práctica

Supongamos que trabajamos en un banco y necesitamos administrar un sistema de gestión de cuentas bancarias. Los hilos en este contexto representan las acciones de depositar y retirar dinero de cuentas bancarias compartidas. La sección crítica se refiere a la acción de actualizar el saldo disponible de la cuenta bancaria.

Aquí es donde entran en juego los candados que vimos durante la práctica. Podemos asegurar la exclusión mutua y prevenir que dos o más transacciones actualicen el saldo de la cuenta bancaria al mismo tiempo.

Cada transacción (hilo) adquiere el candado de la cuenta bancaria respectiva antes de poder actualizar el saldo disponible. Una vez que la acción ha sido completada, el hilo libera el candado. De esta manera, garantizamos que solo un hilo pueda actualizar el saldo de una cuenta en un momento específico, evitando así inconsistencias en los datos.

9. Escribe lo aprendido en esta practica asi como diferencias respecto a las anteriores.

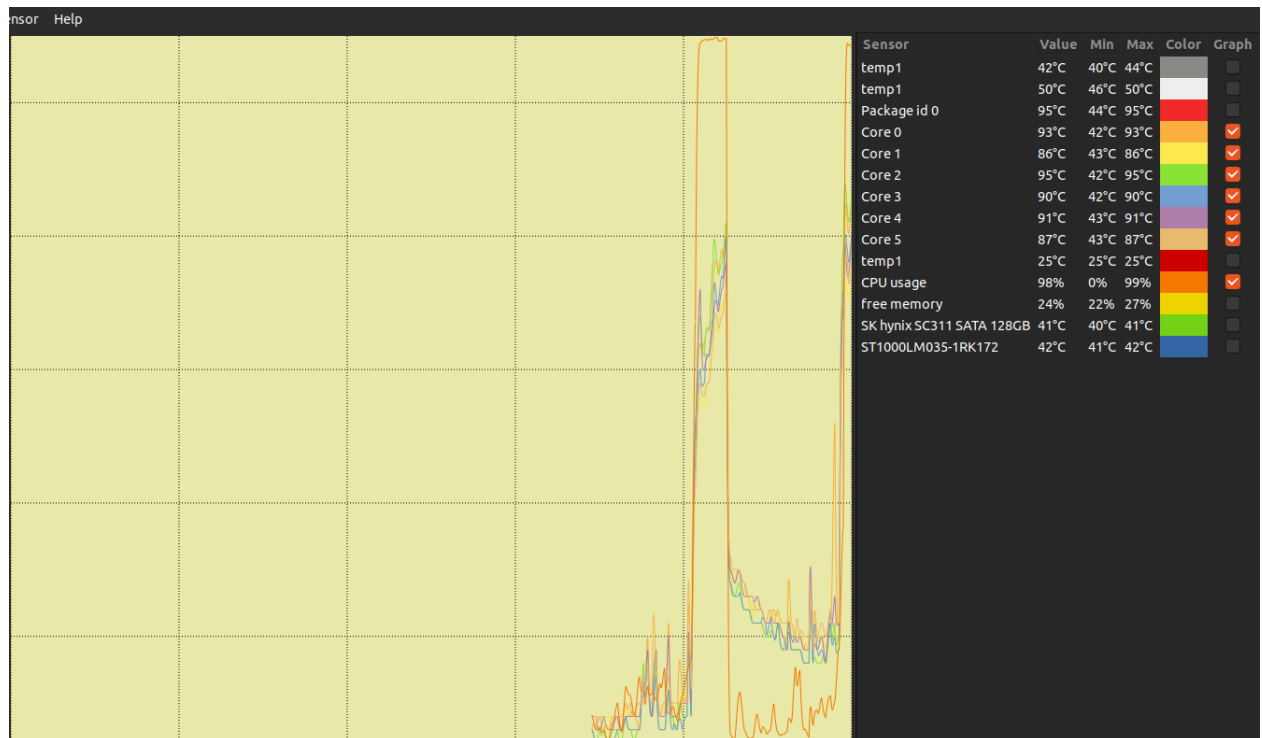
Me parecio entretenida, y a su vez interesante ver el rendimiento de cada candando, y como puede variar los tiempos de ejecución dependiendo de los componentes de la computadora. Además de tener cuidado de no usar excesivamente el metodo `yield()`, asi como variables atomica puede llegar a ser un gasto de recursos innecesarios, y obtener condiciones de carrera respectivamente.

3. Extra

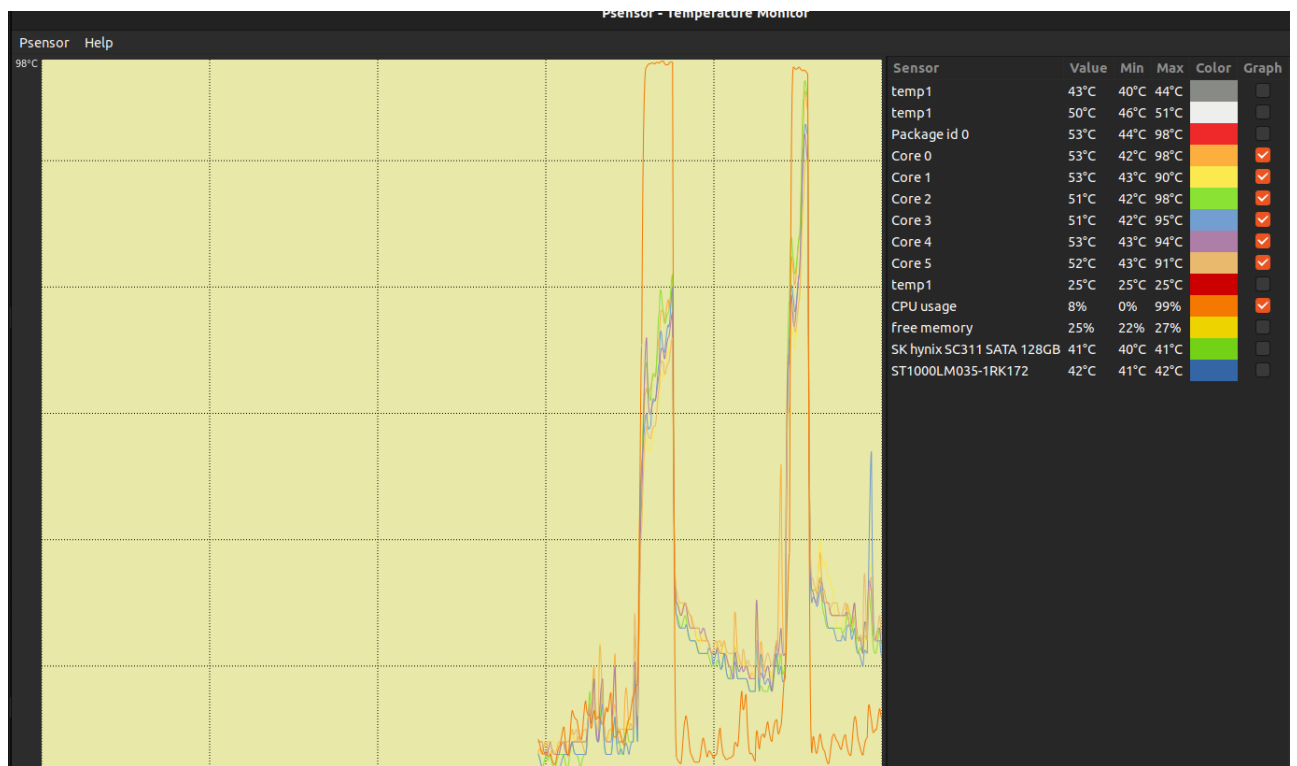
- Toma las temperaturas de tu Computadora y comparalas entre cada algoritmo empleado, puedes usar la cantidad de Hilos Maxima para esto.

En seguida se meustras las temperaturas entre cada candado:

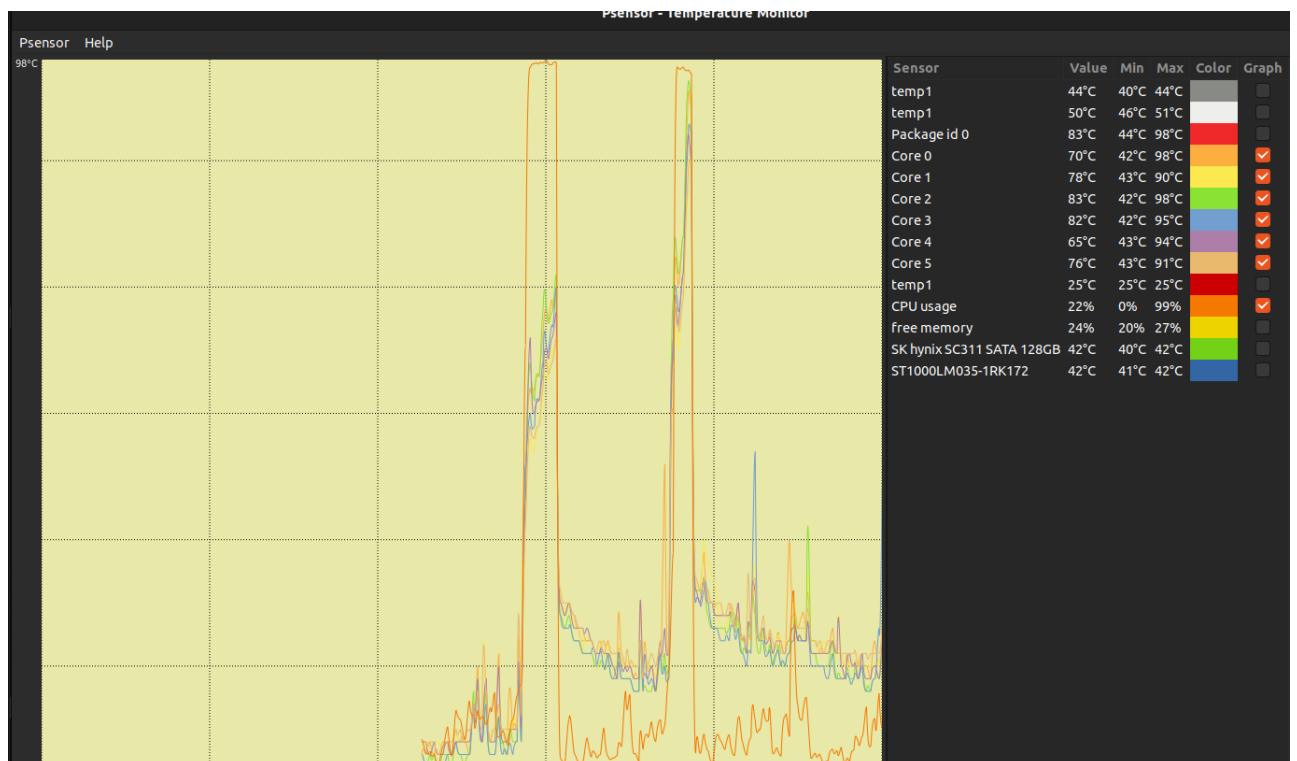
3.1. TASLock



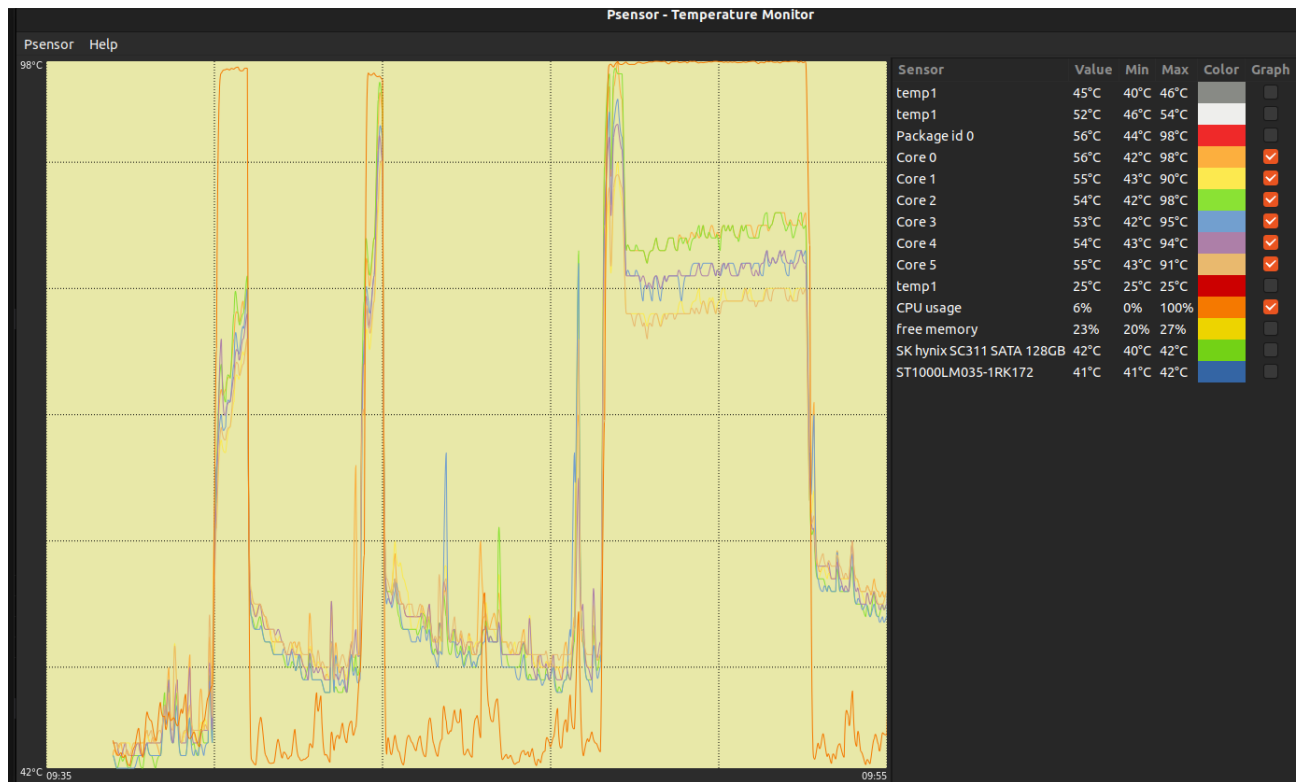
3.2. TTASLock



3.3. BakcoffLock



3.4. CLHLock



3.5. MCSLock



3.6. ALock

