



**Tecnológico
de Monterrey**

TE2004B.501

Diseño de sistemas embebidos avanzados

Matrix multiplication in open MP

Alumna:

Fernanda Monserrat Galvan Romero

A00344712

Ingeniería en Robótica y Sistemas Digitales

Tec de Monterrey, Campus Guadalajara

05/11/2022.

Link al repositorio de Github:

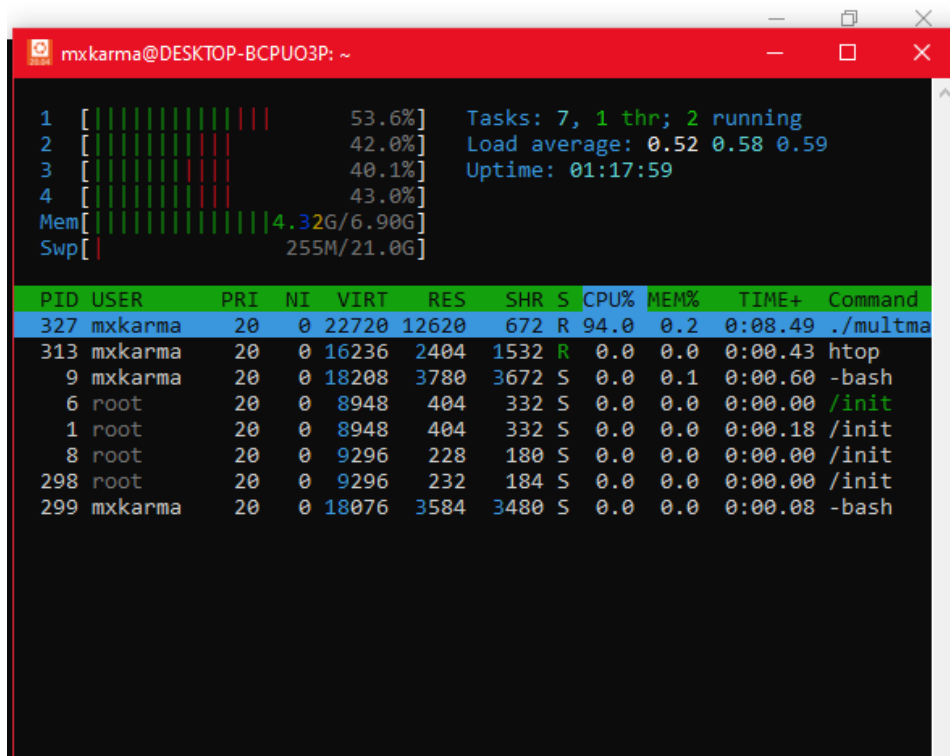
<https://github.com/mxgalvxn/parallel-programming-ITESM/tree/main/practice02>

En la siguiente gráfica se muestra el promedio del tiempo de ejecución con diferentes threads

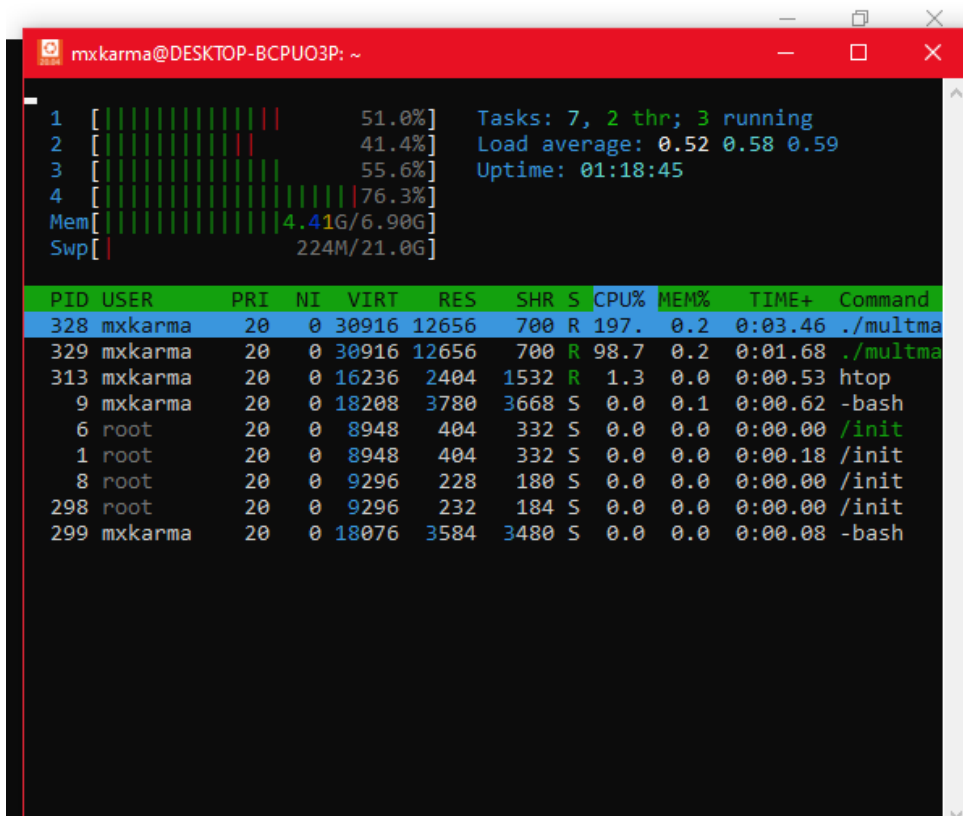


Resultados del htop cuando se ejecutan diferentes threads

1 thread



2 threads



4 threads

```
mxkarma@DESKTOP-BCPU03P: ~  
1 [|||||100.0%]  
2 [|||||100.0%]  
3 [|||||100.0%]  
4 [|||||100.0%]  
Mem[|||||4.04G/6.90G]  
Swp[|284M/21.0G]  
Tasks: 7, 4 thr; 4 running  
Load average: 0.52 0.58 0.59  
Uptime: 01:14:31  


| PID | USER    | PRI | NI | VIRT  | RES   | SHR  | S | CPU% | MEM% | TIME+   | Command  |
|-----|---------|-----|----|-------|-------|------|---|------|------|---------|----------|
| 322 | mxkarma | 20  | 0  | 47308 | 12672 | 700  | R | 341. | 0.2  | 0:09.02 | ./multma |
| 324 | mxkarma | 20  | 0  | 47308 | 12672 | 700  | R | 88.2 | 0.2  | 0:02.28 | ./multma |
| 323 | mxkarma | 20  | 0  | 47308 | 12672 | 700  | R | 86.4 | 0.2  | 0:02.22 | ./multma |
| 325 | mxkarma | 20  | 0  | 47308 | 12672 | 700  | R | 82.9 | 0.2  | 0:02.26 | ./multma |
| 9   | mxkarma | 20  | 0  | 18208 | 3780  | 3668 | S | 0.0  | 0.1  | 0:00.57 | -bash    |
| 313 | mxkarma | 20  | 0  | 16236 | 2404  | 1532 | R | 0.0  | 0.0  | 0:00.20 | htop     |
| 6   | root    | 20  | 0  | 8948  | 404   | 332  | S | 0.0  | 0.0  | 0:00.00 | /init    |
| 1   | root    | 20  | 0  | 8948  | 404   | 332  | S | 0.0  | 0.0  | 0:00.18 | /init    |
| 8   | root    | 20  | 0  | 9296  | 228   | 180  | S | 0.0  | 0.0  | 0:00.00 | /init    |
| 298 | root    | 20  | 0  | 9296  | 232   | 184  | S | 0.0  | 0.0  | 0:00.00 | /init    |
| 299 | mxkarma | 20  | 0  | 18076 | 3584  | 3480 | S | 0.0  | 0.0  | 0:00.08 | -bash    |


```

8 threads

```
mxkarma@DESKTOP-BCPU03P: ~  
1 [||||| 93.8%] Tasks: 6, 1 thr; 1 running  
2 [||||| 92.7%] Load average: 0.52 0.58 0.59  
3 [||||| 91.9%] Uptime: 01:19:05  
4 [||||| 92.0%]  
Mem[||||| 4.41G/6.90G]  
Swp[| 224M/21.0G]  
  
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command  
313 mxkarma 20 0 16236 2404 1532 R 0.0 0.0 0:00.54 htop  
9 mxkarma 20 0 18208 3780 3512 S 0.0 0.1 0:00.65 -bash  
6 root 20 0 8948 404 332 S 0.0 0.0 0:00.00 /init  
1 root 20 0 8948 404 332 S 0.0 0.0 0:00.18 /init  
8 root 20 0 9296 228 180 S 0.0 0.0 0:00.00 /init  
298 root 20 0 9296 232 184 S 0.0 0.0 0:00.00 /init  
299 mxkarma 20 0 18076 3584 3480 S 0.0 0.0 0:00.08 -bash
```

16 threads

```
mxkarma@DESKTOP-BCPU03P: ~  
1 [||||| 82.1%] Tasks: 6, 1 thr; 1 running  
2 [||||| 81.5%] Load average: 0.52 0.58 0.59  
3 [||||| 81.0%] Uptime: 01:19:33  
4 [||||| 82.1%]  
Mem[||||| 4.47G/6.90G]  
Swp[| 224M/21.0G]  
  
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command  
313 mxkarma 20 0 16236 2404 1532 R 0.5 0.0 0:00.60 htop  
9 mxkarma 20 0 18208 3780 3512 S 0.0 0.1 0:00.66 -bash  
6 root 20 0 8948 404 332 S 0.0 0.0 0:00.00 /init  
1 root 20 0 8948 404 332 S 0.0 0.0 0:00.18 /init  
8 root 20 0 9296 228 180 S 0.0 0.0 0:00.00 /init  
298 root 20 0 9296 232 184 S 0.0 0.0 0:00.00 /init  
299 mxkarma 20 0 18076 3584 3480 S 0.0 0.0 0:00.08 -bash
```

Conclusión:

En otras prácticas vimos como el uso ineficiente de threads, pero en este código pudimos observar 2 cosas; el uso correcto de threads y como la paralelización puede ser un recurso bastante útil en códigos con procesos repetitivos que no dependen uno del otro, por ejemplo en este ejercicio podemos paralelizar las multiplicaciones de los diferentes índices de la matriz, lo que nos permitió hacer el proceso más rápido y como los resultados de las multiplicaciones son independientes entre sí, no causó problemas el que varios threads accedieron a la memoria. Si la paralelización nos ahorró una cantidad considerable de tiempo que tal vez no se pueda apreciar bien en la diferencia de milisegundos entre una muestra y otra, a gran escala nos podría ahorrar horas o días.