

# Practica 5-a

## lscpu

```
4419 ± lscpu
Arquitectura:          x86_64
modo(s) de operación de las CPUs:32-bit, 64-bit
Orden de bytes:       Little Endian
CPU(s):               8
On-line CPU(s) list:  0-7
Hilo(s) de procesamiento por núcleo:2
Núcleo(s) por «socket»:4
Socket(s):            1
Modo(s) NUMA:         1
ID de fabricante:     GenuineIntel
Familia de CPU:       6
Modelo:               158
Model name:           Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz
Revisión:             9
CPU MHz:              899.951
CPU max MHz:          3800,0000
CPU min MHz:          800,0000
BogoMIPS:             5616.00
Virtualización:       VT-x
Caché L1d:            32K
Caché L1i:            32K
Caché L2:             256K
Caché L3:             6144K
NUMA node0 CPU(s):    0-7
Flags:                fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
_tsc art arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc aper
_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdr
ep bmi2 erms invpcid mpx rdseed adx smap clflushopt xsaveopt xsavec xge
```

## make info

```
cp bmi2 erms invpcid mpx rdseed adx smap clflushopt x
antonio-Lenovo 35 ~/Documentos/git/PRACTICAS-EC/Pract
4420 ± make info
line size = 64B
cache size = 32K/32K/256K/6144K/
cache level = 1/1/2/3/
cache type = Data/Instruction/Unified/Unified/
```

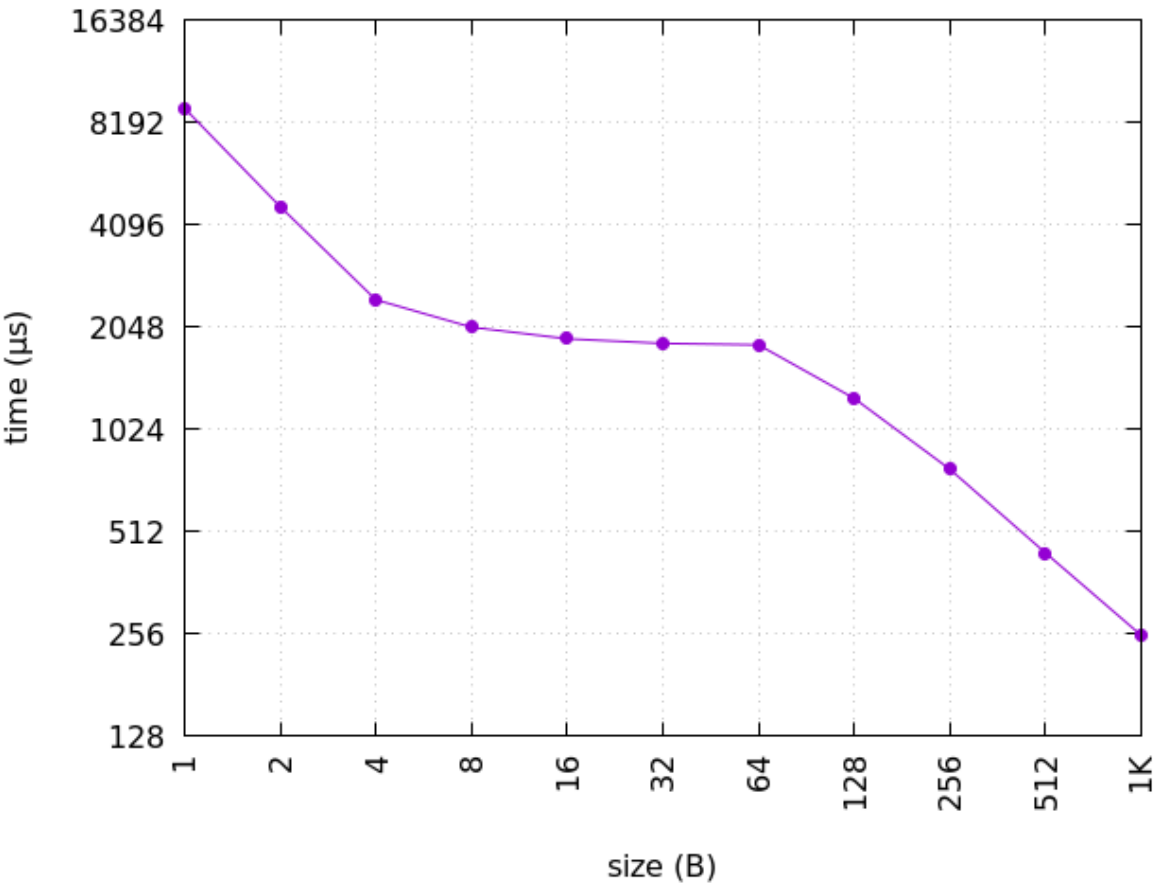
# CPU-World

## General information

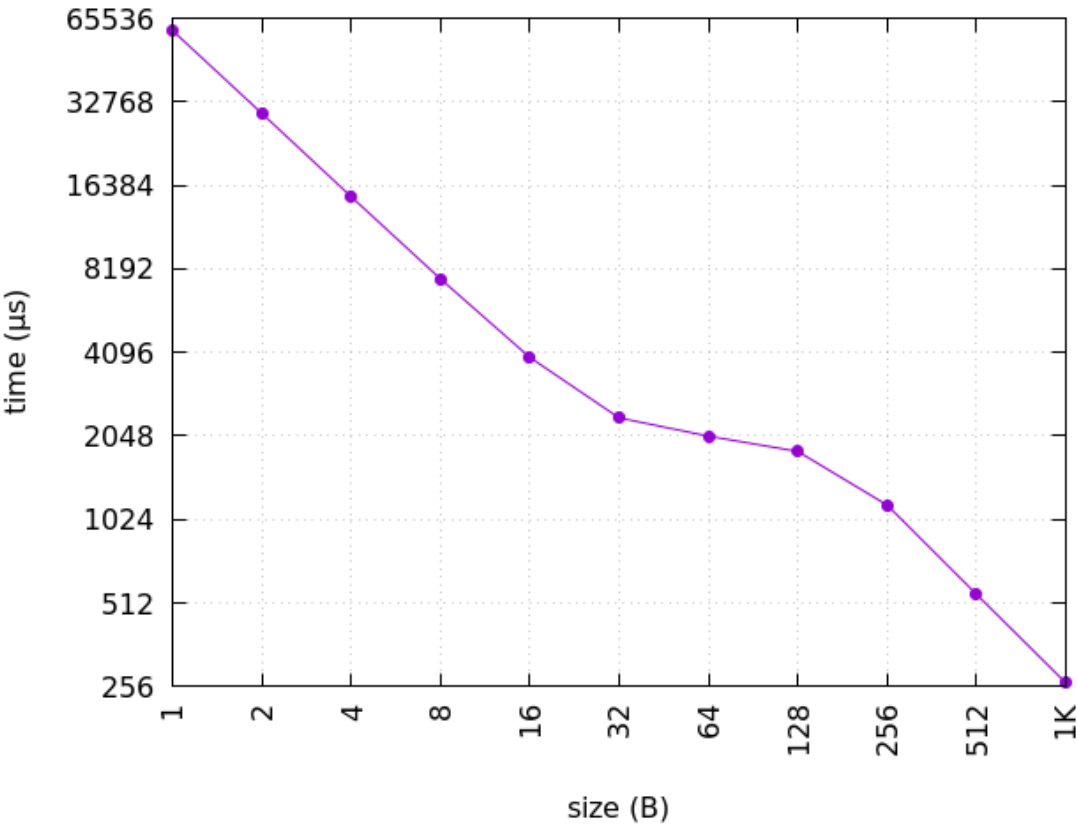
|  |  |
|--|--|
| Type                                   | <a href="#">CPU / Microprocessor</a>                               |
| Market segment                         | Mobile   |
| Family                                 | <a href="#">Intel Core i7 Mobile</a> ▼                             |
| Model number <span>?</span>            | <a href="#">i7-7700HQ</a> ▼  |
| CPU part number                        | CL8067702870109 is an OEM/tray microprocessor ▼                    |
| <b>Frequency</b> <span>?</span>        | <b>2800 MHz</b>  |
| Maximum turbo frequency                | 3800 MHz (1 core)<br>3600 MHz (2 cores)<br>3400 MHz (3 or 4 cores) |
| <b>Bus speed</b> <span>?</span>        | <b>8 GT/s DMI</b>  |
| <b>Clock multiplier</b> <span>?</span> | <b>28</b>  |
| Package                                | 1440-ball micro-FCBGA  |
| <b>Socket</b>                          | <b>BGA1440</b>   |
| Size                                   | 1.65" x 1.1" / 4.2cm x 2.8cm                                       |
| Introduction date                      | <a href="#">January 3, 2017</a>                                    |
| Price at introduction                  | \$378  |

| Cache details  |                       |                       |                                |                                       |
|----------------|-----------------------|-----------------------|--------------------------------|---------------------------------------|
| Cache:         | L1 data               | L1 instruction        | L2                             | L3                                    |
| Size:          | 4 x 32 KB             | 4 x 32 KB             | 4 x 256 KB                     | 6 MB                                  |
| Associativity: | 8-way set associative | 8-way set associative | 4-way set associative          | 12-way set associative                |
| Line size:     | 64 bytes              | 64 bytes              | 64 bytes                       | 64 bytes                              |
| Comments:      | Direct-mapped         | Direct-mapped         | Non-inclusive<br>Direct-mapped | Inclusive<br>Shared between all cores |

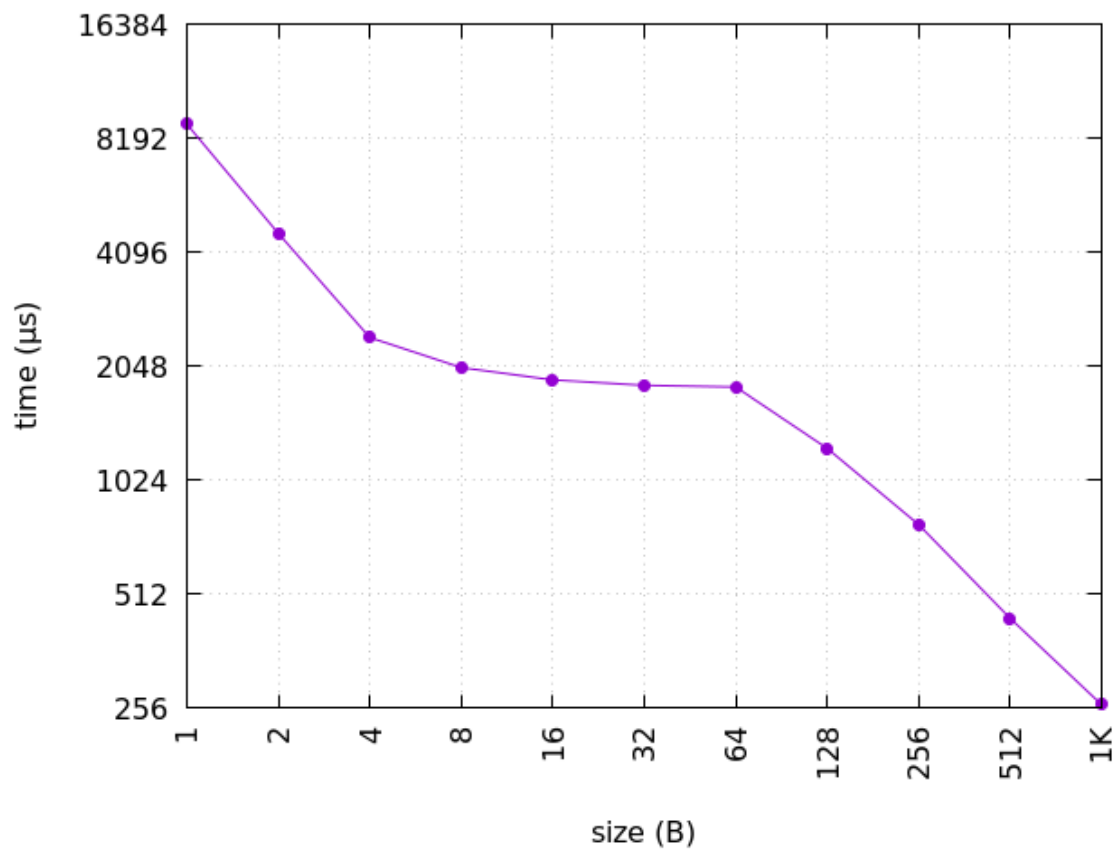
line.Ofast.png



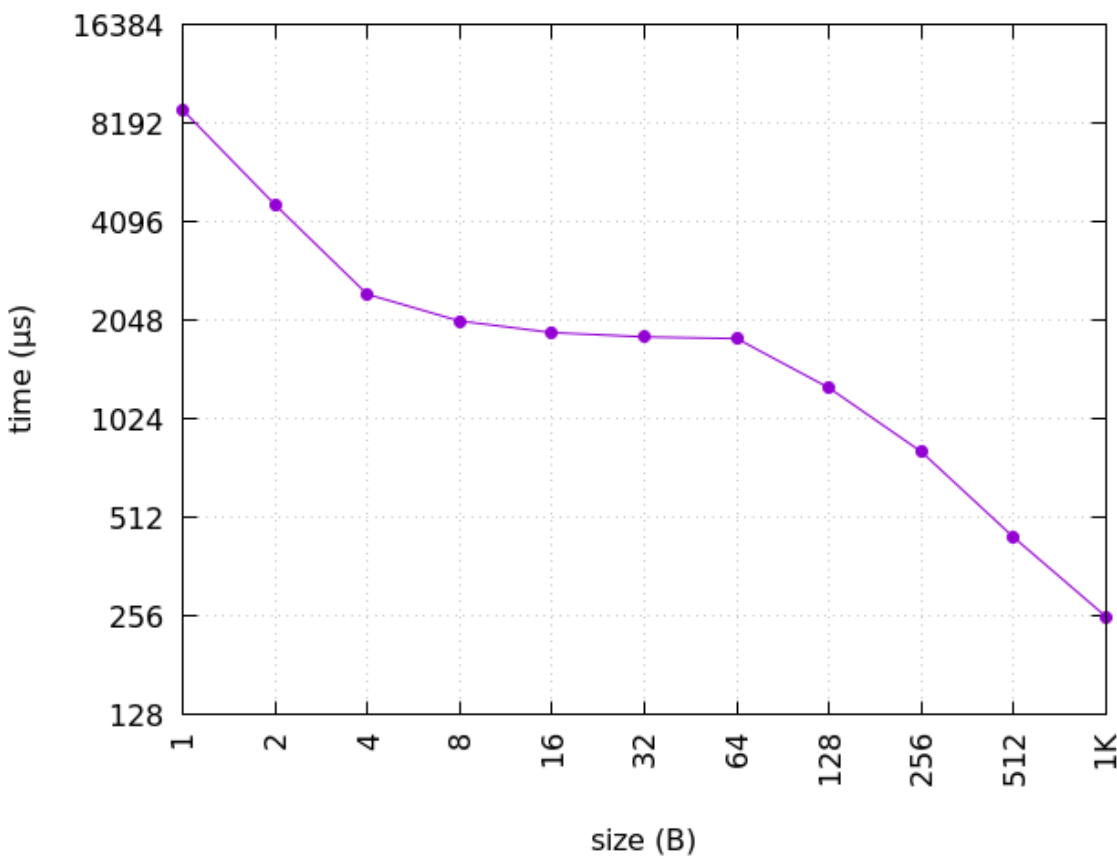
line.O0.png



line.O1.png



line.O2.png



# Memoria

El objetivo de la practica ha sido ver la importancia que tiene la memoria cache en los procesadores y además poder calcular la capacidad que tiene la memoria cache de nuestro procesador.

Para ello hemos cogido diferentes tamaños de bloques a leer que tienen de tamaño  $2^n B$ .

y luego con cada tamaño de bloque realizamos una cantidad elevada de operaciones, pero esas operaciones van a tener muy poco peso. Es decir son rápidas de realizar. Con ello podremos ver en como influye el tamaño del bloque. Para cada tamaño de bloque tomamos el tiempo antes y después de realizar las operaciones y así podemos luego realizar la gráfica.

Yo diría que la optimización de nivel 0 es la que mejor representa la cache, ya que muestra un dato inequívocamente de cual es el tamaño de la cache, ya que las otras pueden dar lugar a confusiones.