

Practica 5

Realizado por: Juan Sánchez Rodríguez

Apartado A

Lo primero que hemos hecho es consultar el modelo usando lscpu:

```
srjuan@eil40163:~/Escritorio/Home/EC/P5$ lscpu
Arquitectura:          x86_64
modo(s) de operación de las CPUs:32-bit, 64-bit
Orden de bytes:        Little Endian
CPU(s):                4
On-line CPU(s) list:   0-3
Hilo(s) de procesamiento por núcleo:1
Núcleo(s) por «socket»:4
Socket(s):             1
Modo(s) NUMA:          1
ID de fabricante:      GenuineIntel
Familia de CPU:         6
Modelo:                 60
Model name:             Intel(R) Core(TM) i5-4460 CPU @ 3.20GHz
Revisión:               3
CPU MHz:                926.367
CPU max MHz:            3400,0000
CPU min MHz:            800,0000
BogoMIPS:               6385.25
Virtualización:         VT-x
Caché L1d:              32K
Caché L1i:              32K
Caché L2:               256K
Caché L3:               6144K
NUMA node0 CPU(s):     0-3
Flags:                  fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1g
b rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_t
sc aperfmperf eagerfpu pni pclmulqdq dtes64 monitor ds_cpl vmx est tm2 ssse3 sdb
g fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer a
es xsave avx f16c rdrand lahf_lm abm epb tpr_shadow vnmi flexpriority ept vpid f
sgsbase tsc_adjust bml avx2 smep bmi2 erms invpcid xsaveopt dtherm ida arat pln
pts
```

Después usamos make info para saber el tamaño de línea:

```
srjuan@eil40163:~/Escritorio/Home/EC/P5$ make info
line size = 64B
cache size = 32K/32K/256K/6144K/
cache level = 1/1/2/3/
cache type = Data/Instruction/Unified/Unified/
srjuan@eil40163:~/Escritorio/Home/EC/P5$
```

Ahora que sabemos gracias a lscpu nuestro modelo lo buscamos en cpu world para ver los detalles de la caché:

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	8-way set associative	12-way set associative
Line size:	64 bytes	64 bytes	64 bytes	64 bytes
Comments:	Direct-mapped	Direct-mapped	Non-inclusive Direct-mapped	Inclusive Shared between all cores

Después de ver esto hacemos la siguiente modificación en line.cc, añadiendo así el xor:

```
#include <algorithm>    // nth_element
#include <array>        // array
#include <chrono>       // high_resolution_clock
#include <iomanip>       // setw
#include <iostream>     // cout
#include <vector>       // vector

using namespace std::chrono;

const unsigned MAXLINE = 1024; // maximum line size to test
const unsigned GAP = 12;      // gap for cout columns
const unsigned REP = 100;     // number of repetitions of every test

int main()
{
    std::cout << "#"
               << std::setw(GAP - 1) << "line (B)"
               << std::setw(GAP + 1) << "time (µs)"
               << std::endl;

    for (unsigned line = 1; line <= MAXLINE; line <= 1) // line in bytes
    {
        std::vector<duration<double, std::micro>> score(REP);

        for (auto &s: score)
        {
            std::vector<char> bytes(1 << 24); // 16MB

            auto start = high_resolution_clock::now();

            for (unsigned i = 0; i < bytes.size(); i += line)
                bytes[i] ^= 1;

            auto stop = high_resolution_clock::now();

            s = stop - start;
        }

        std::nth_element(score.begin(),
                        score.begin() + score.size() / 2,
                        score.end());

        std::cout << std::setw(GAP) << line
                  << std::setw(GAP) << std::fixed << std::setprecision(1)
                  << std::setw(GAP) << score[score.size() / 2].count()
                  << std::endl;
    }
}
```

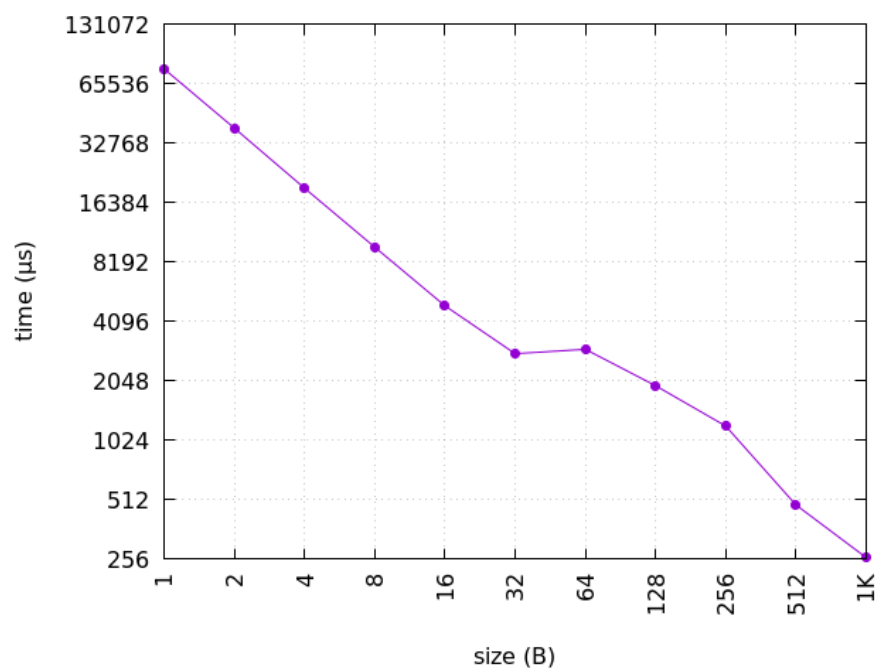
Una vez añadido lo compilamos con -O0, -O1, -O2, -Ofast para ver cual es mejor:

-O0			-O1		
#	line (B)	time (µs)	#	line (B)	time (µs)
	1	77777.9		1	7468.2
	2	38998.8		2	3801.8
	4	19362.5		4	2798.5
	8	9741.9		8	3002.5
	16	4911.9		16	3098.2
	32	2810.7		32	3152.7
	64	2958.3		64	3089.0
	128	1934.6		128	1753.3
	256	1208.7		256	850.6
	512	481.7		512	314.2
	1024	260.9		1024	163.1

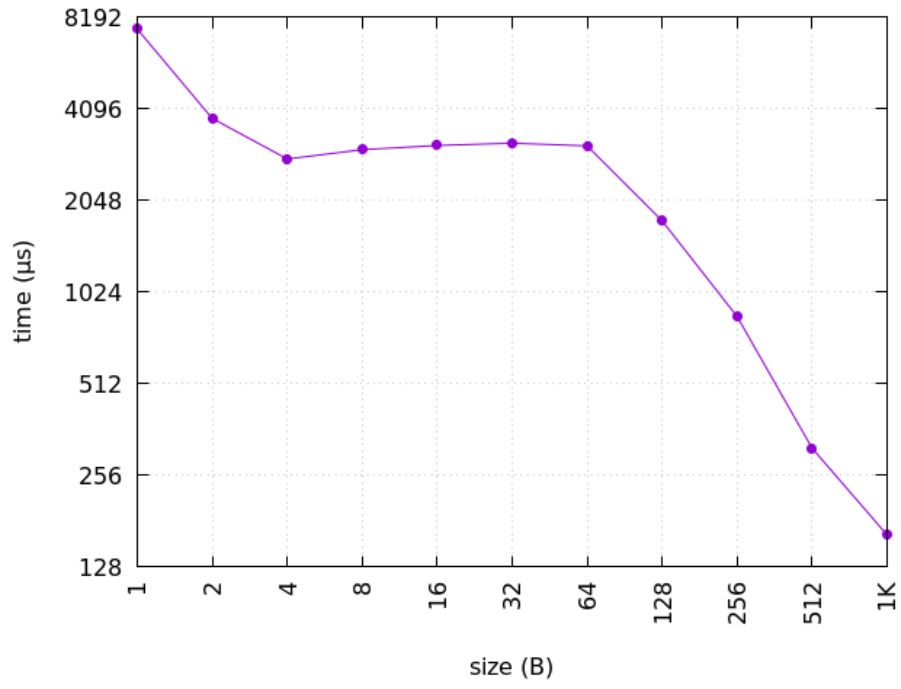
-O2			-Ofast		
#	line (B)	time (µs)	#	line (B)	time (µs)
	1	7470.8		1	7469.4
	2	3807.0		2	3802.7
	4	2835.5		4	2801.2
	8	3007.2		8	3010.8
	16	3093.8		16	3101.4
	32	3154.2		32	3140.6
	64	3107.5		64	3101.4
	128	1746.6		128	1746.9
	256	848.4		256	852.9
	512	317.8		512	315.3
	1024	164.6		1024	163.2

A continuación presentaré las gráficas:

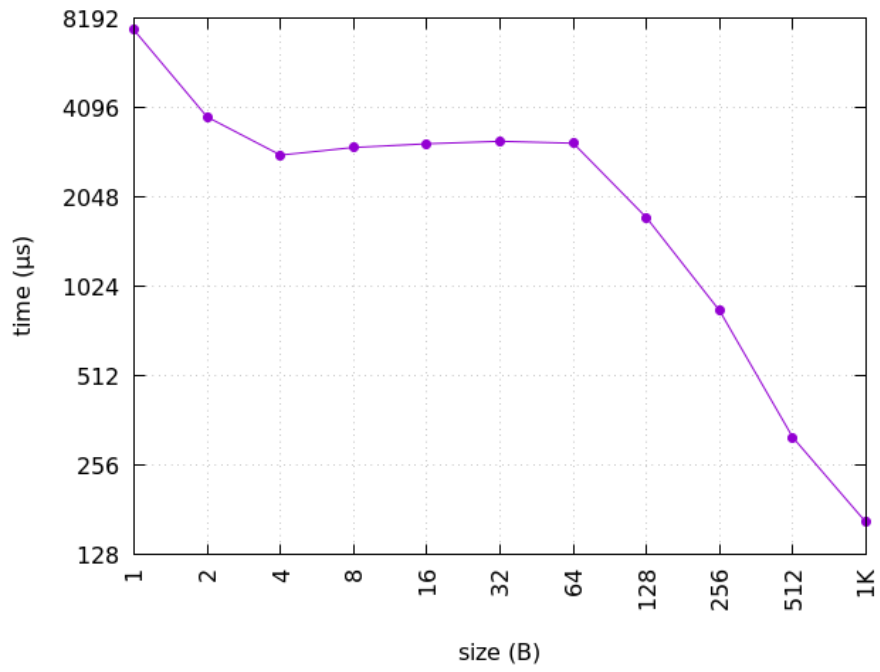
-O0



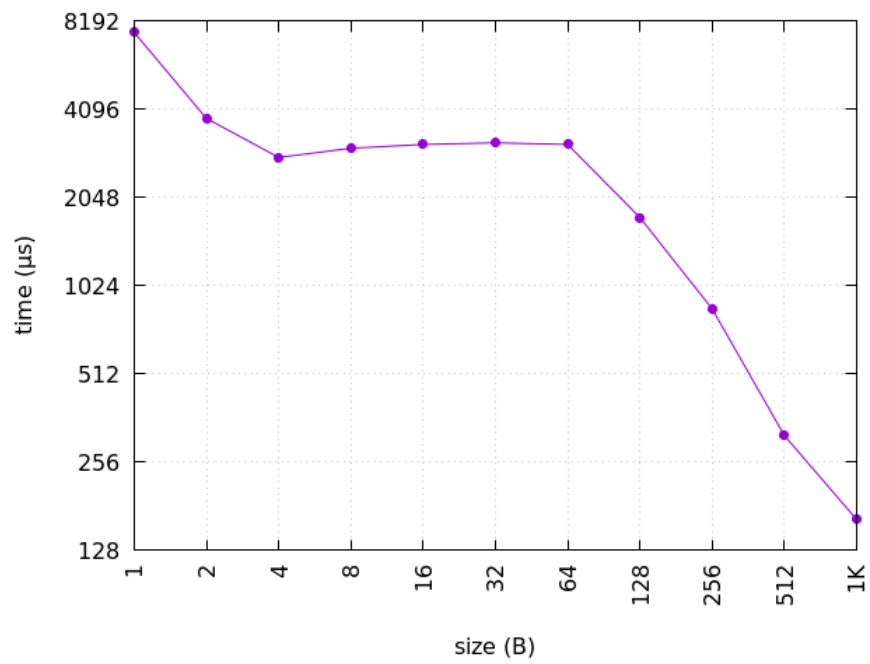
-O1



-O2



-Ofast



Cómo podemos apreciar menos -O0, las otras tres son mucho mejores y muy parecidas entre sí.