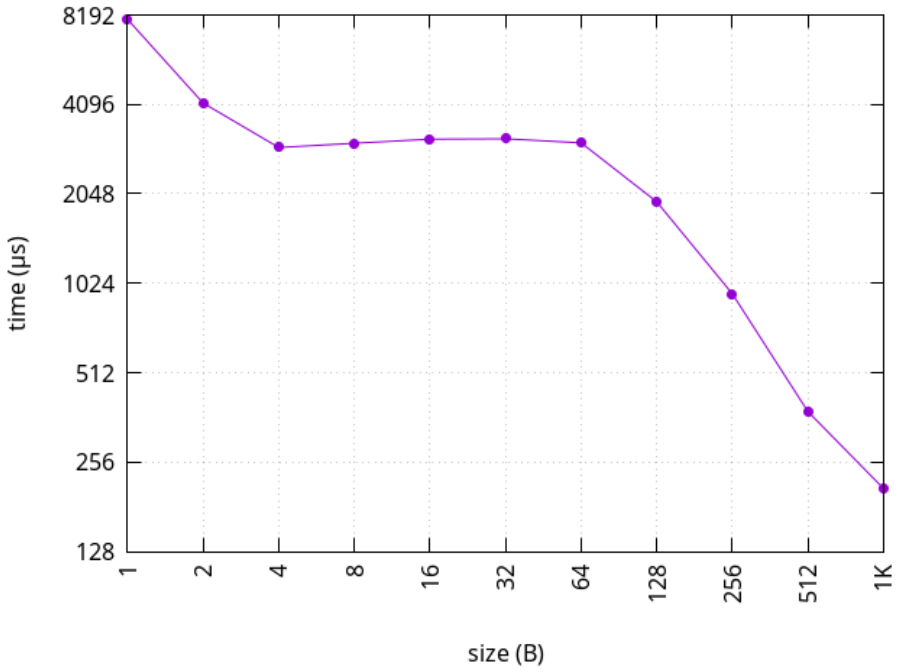
Práctica 6. Estructura de Computadores
Noelia Escalera Mejías. Grupo A3

Programa line.cc

```
#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; // maximun 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    ) << "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;
       }
}
```

Gráfica line.png



La gráfica anterior muestra cómo el tamaño de línea es de 64 bytes. Podemos observar que es ahí dónde acaba esa especie de línea constante. Eso significa que ha cargado un bloque de ese tamaño en memoria y a partir de ahí empieza a haber fallos y la gráfica empieza a decrecer.

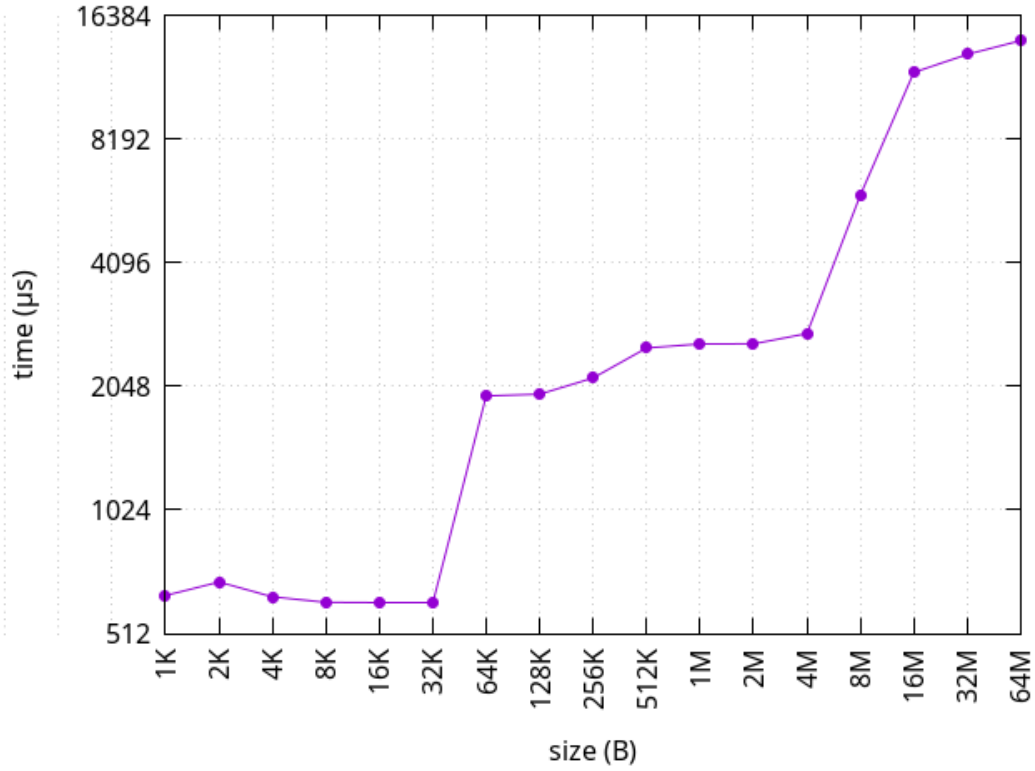
Noelia Escalera Mejías

Grupo A3

Programa size.cc

```
#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 MINSIZE = 1 << 10; // minimun line size to test:  1KB
const unsigned MAXSIZE = 1 << 26; // maximun line size to test: 32MB
const unsigned GAP = 12;          // gap for cout columns
const unsigned REP = 100;         // number of repetitions of every test
const unsigned STEPS = 1e6;       // steps

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

       for (unsigned size = MINSIZE; size <= MAXSIZE; size *= 2)
       {
       std::vector<duration<double, std::micro>> score(REP);

       for (auto &s: score)
       {
              std::vector<char> bytes(size);

              auto start = high_resolution_clock::now();

              unsigned mask = size -1 ;
              for (unsigned i = 0; i < STEPS * 64; i+=64)
                     bytes[i & mask]++;

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

Gráfica size.png



La gráfica anterior muestra que mi procesador tiene 3 niveles de caché para datos, a parte de un nivel para instrucciones. El primer nivel es de 32K, el segundo de 256K y el tercero de 6M.

Captura de lscpu:

```
noelia@noelia-HP-ENVY-17-Notebook-PC:~$ lscpu
Arquitectura:                    x86_64
modo(s) de operación de las CPUs:    32-bit, 64-bit
Orden de los bytes:              Little Endian
CPU(s):                          8
Lista de la(s) CPU(s) en línea:      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:                          60
Nombre del modelo:               Intel(R) Core(TM) i7-4700MQ CPU @ 2.40GHz
Revisión:                        3
CPU MHz:                         1700.974
CPU MHz máx.:                    3400,0000
CPU MHz mín.:                    800,0000
BogoMIPS:                        4788.87
Virtualización:                  VT-x
Caché L1d:                       32K
Caché L1i:                       32K
Caché L2:                        256K
Caché L3:                        6144K
CPU(s) del nodo NUMA 0:          0-7
Indicadores:                     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 h
t tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 m
onitor ds_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm cpu
id_fault epb invpcid_single pti ssbd ibrs ibpb stibp tpr_shadow vnmi flexpriority ept vpid fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid xsav
eopt dtherm ida arat pln pts flush_l1d
```

Noelia Escalera Mejías

Grupo A3

Captura de los detalles de la caché de CPU World

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

Noelia Escalera Mejías

Grupo A3