# CARLOS MORALES AGUILERA
# PRÁCTICA 5. - CACHÉ.

## 1. Código empleado.

### - line.cc

```
// Carlos Morales Aguilera- 75925767-F
// EC - 2º B
// line.cc

#include <array>        // array
#include <chrono>       // high_resolution_clock
#include <iomanip>      // setw
#include <iostream>     // cout

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
const unsigned STEPS = 32 * 1024 * 1024; // 32M steps

std::array<char, 1 << 24> bytes;        // 16MB

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

        bytes.fill(0);

        for (unsigned line = 1; line <= MAXLINE; line <<= 1) // line in bytes
        {
                std::array<duration<float, std::milli>, REP> score;

                for (auto &s: score)
                {
                        auto start = high_resolution_clock::now();

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

                        auto stop = high_resolution_clock::now();

                        s = stop - start;
                }

                std::cout << std::setw(GAP) << line
                        << std::setw(GAP) << std::fixed << std::setprecision(1)
                        << std::setw(GAP) << std::min(score.begin(), score.end())->count()
                        << std::endl;
        }

        return bytes.front() + bytes.back();
}
```

**- size.cc**

```
// Carlos Morales Aguilera- 75925767-F
// EC - 2º B
// size.cc

#include <array>        // array
#include <chrono>       // high_resolution_clock
#include <iomanip>      // setw
#include <iostream>     // cout

using namespace std::chrono;

const unsigned LINE = 16;                   // line size
const unsigned MAXSIZE = 1024;              // maximun cache size to test
const unsigned GAP = 12;                    // gap for cout columns
const unsigned REP = 10;                    // number of repetitions of every test
const unsigned STEPS = 32 * 1024 * 1024; // 32M steps

std::array<char, 1 << 24> bytes;        // 16MB

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

        bytes.fill(0);

        for (unsigned size = MAXSIZE; size <= STEPS; size <<= 1)
        {
                std::array<duration<float, std::milli>, REP> score;

                for (auto &s: score)
                {
                        auto start = high_resolution_clock::now();

                        for (unsigned i = 0; i < STEPS; ++i)
                                bytes[(i<<6)&(size-1)]^=1;
                        auto stop = high_resolution_clock::now();

                        s = stop - start;
                }

                std::cout << std::setw(GAP) << size
                        << std::setw(GAP) << std::fixed << std::setprecision(1)
                        << std::setw(GAP) << std::min(score.begin(), score.end())->count()
                        << std::endl;
        }

        return bytes.front() + bytes.back();
}
```

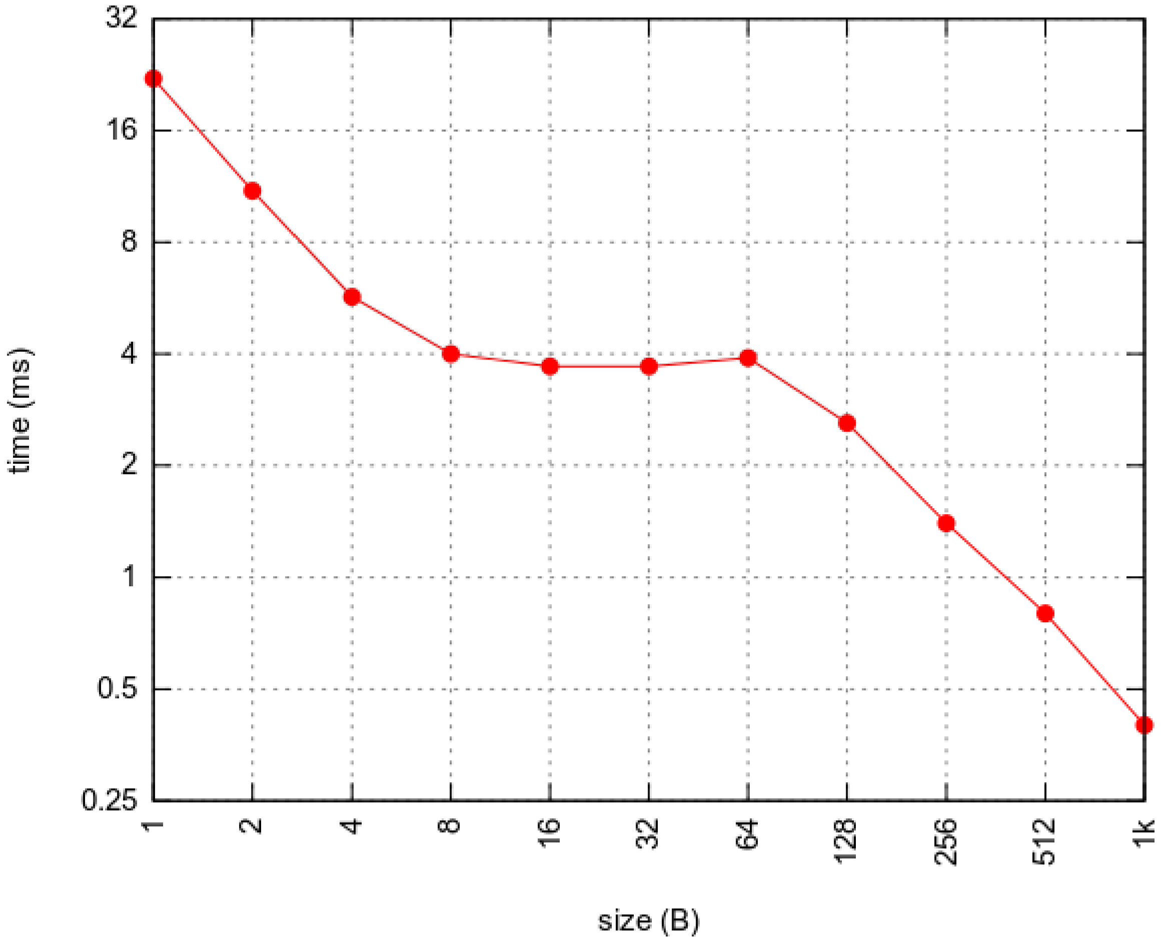
**2. Gráficas.**

**- line**

Resultados obtenidos:

| # line (B) | time (ms) |
|---|---|
| 1 | 22.1 |
| 2 | 11.0 |
| 4 | 5.7 |
| 8 | 4.0 |
| 16 | 3.7 |
| 32 | 3.7 |
| 64 | 3.9 |
| 128 | 2.6 |
| 256 | 1.4 |
| 512 | 0.8 |
| 1024 | 0.4 |

Gráfica obtenida:

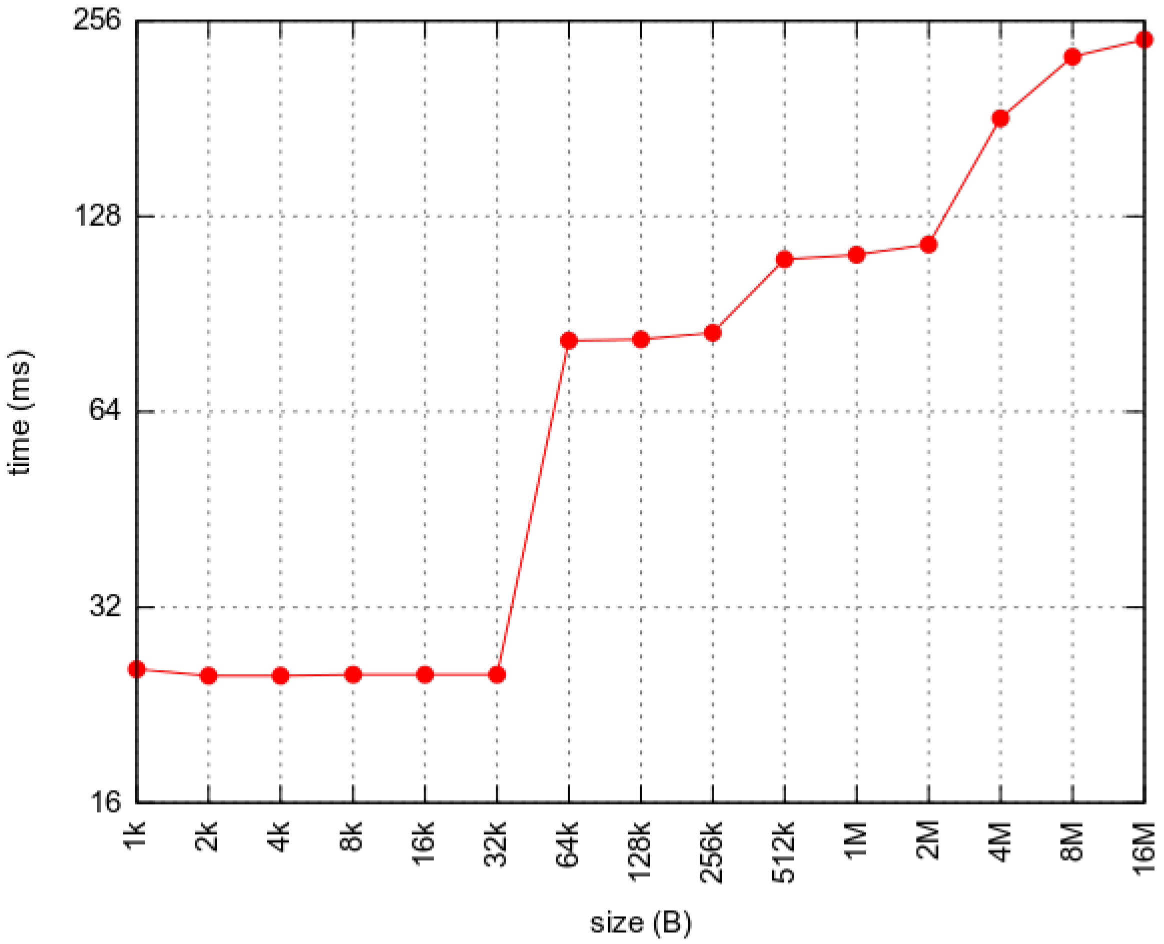


**- size**

Resultados obtenidos:

| # size (B) | time (ms) |
|---|---|
| 1024 | 25.7 |
| 2048 | 25.1 |
| 4096 | 25.1 |
| 8192 | 25.2 |
| 16384 | 25.2 |
| 32768 | 25.2 |
| 65536 | 82.4 |
| 131072 | 82.7 |
| 262144 | 84.7 |
| 524288 | 109.9 |
| 1048576 | 111.7 |
| 2097152 | 115.8 |
| 4194304 | 181.3 |
| 8388608 | 225.4 |
| 16777216 | 239.7 |

Gráfica obtenida:



### 3. Modelo de procesador.

**Procesador:** Intel Core i7-4510U Mobile processor – CL8064701477301

**Cachés:**

- Level 1 cache size ? 2 x 32 KB 8-way set associative instruction caches
  2 x 32 KB 8-way set associative data caches
- Level 2 cache size ? 2 x 256 KB 8-way set associative caches
- Level 3 cache size 4 MB 16-way set associative shared cache