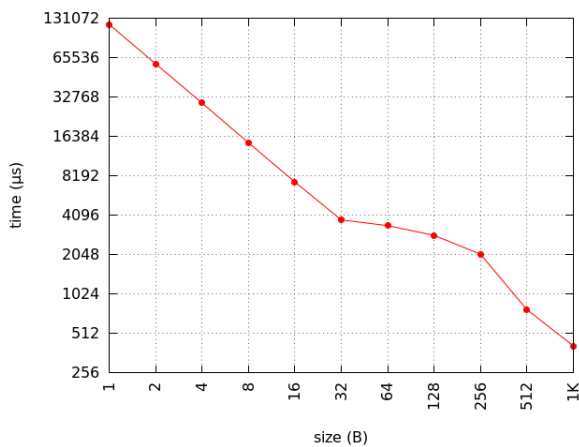


PRÁTICA 5B: LINE.CC + SIZE.CC EC

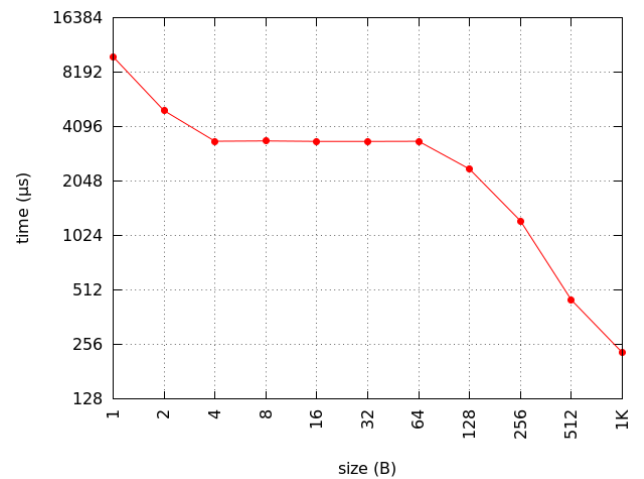
Jose Antonio Padial Molina
Curso 2016/2017

Esta práctica trata de averiguar el tamaño de línea de la caché de nuestro ordenador. Para ello, creamos un programa que realiza un cambio en el último bit de un array 32M veces y esto lo repite para cada posible tamaño de línea. Con este programa obtenemos unos datos con los que comprobaremos la variación de tiempos en realizar estas operaciones y a partir de eso, averiguar el tamaño de línea. Estas son las gráficas obtenidas para el programa line.cc con optimizaciones O0, O1, O2 y Ofast:

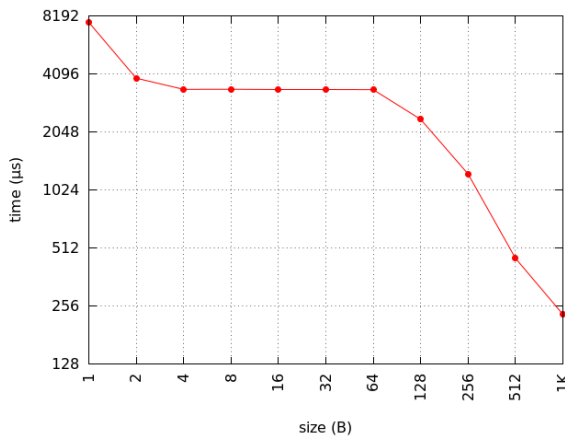
Optimización O0



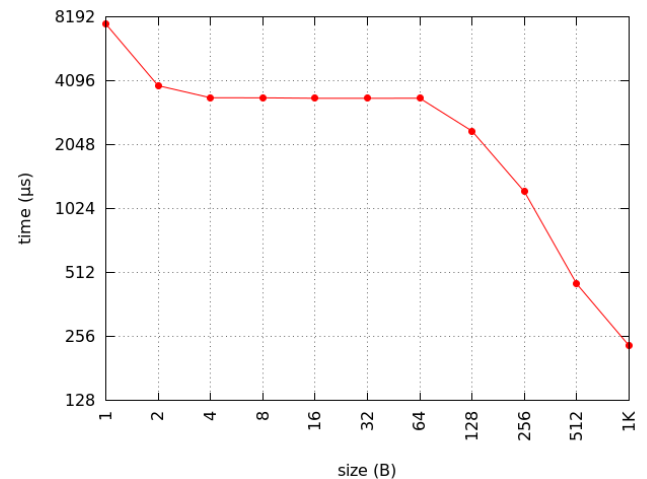
Optimización O1



Optimización O2



Optimización Ofast



Se puede observar que la optimización O0 es pésima para averiguar el tamaño de línea, sin embargo el resto de gráficas son prácticamente iguales y se puede apreciar que hay un estancamiento antes de llegar a 64B y a partir de ahí la gráfica desciende rápidamente. Con esto deducimos que el tamaño de línea es 64B. Si miramos las capturas de información de la cpu:

make info

```
josepadial@eil40160:~/Escritorio/Home/Cris$ make info
line size = 64B
cache size = 32K/32K/256K/6144K/
cache level = 1/1/2/3/
cache type = Data/Instruction/Unified/Unified/
josepadial@eil40160:~/Escritorio/Home/Cris$
```

lscpu

```
josepadial@eil40160:~/Escritorio/Home/Cris$ lscpu
Arquitectura:          i686
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
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:                1000.000
CPU max MHz:            3400,0000
CPU min MHz:            800,0000
BogoMIPS:               6385.86
Virtualización:         VT-x
Caché L1d:              32K
Caché L1i:              32K
Caché L2:               256K
Caché L3:               6144K
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 nx pdpe1gb rdtscp
lm constant_tsc arch_perfmon pebs bts xtopology nonstop_tsc aperfmperf eagerfpu
pni pclmulqdq dtes64 monitor ds_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm p
cid sse4_1 sse4_2 x2apic movbe popcnt tsc deadline_timer aes xsave avx f16c rdra
nd lahf_lm abm epb tpr shadow vnmi flexpriority epb vpid fsgsbase tsc_adjust bmi
1 avx2 smep bmi2 erms invpcid xsaveopt dtherm ida arat pln pts
```

CPU-WORLD

Intel Core i5-4460 - CM8064601560722 / BX80646I54460 / BXC80646I54460 - Mozilla Firefox

Intel Core i5-4460 - C... x +

one drive

■ CPU ID information for the Core i5-4460

Detailed characteristics of processor's internals, including x86 instruction set extensions and individual instructions, high- and low-level technologies, are listed below. This list was acquired from an actual Intel Core i5 i5-4460 processor with the help of the x86 CPUID instruction. Any discrepancies between CPUID features and official specifications are likely due to some features being disabled in BIOS, or due to a bug in our CPUID decoding algorithm. Different steppings of Intel processors may also have slightly different features.

Our CPUID database contains 3 records for this microprocessor. [See all submitted records.](#)

Use our [CPU identification tool](#) to check features of your processor.

Submission details			
Manufacturer:	Intel	Measured frequency:	3601 MHz
CPU Family:	Core i5	Comment:	
Processor Number:	i5-4460		
Part number (supplied):	CM8064601560722	Submitted by:	CPU-World
Part number (guessed):	CM8064601560722	Submitted on:	
S-Spec Number:	SR1QK	CWID version:	0.5

General information	
Vendor:	GenuineIntel
Processor name (BIOS):	Intel(R) Core(TM) i5-4460 CPU @ 3.20GHz
Cores:	4
Logical processors:	4
Processor type:	Original OEM Processor
CPUID signature:	306C3
Family:	6 (06h)
Model:	60 (03Ch)
Stepping:	3 (03h)
TLB/Cache details:	64-byte Prefetching Data TLB: 1-GB pages, 4-way set associative, 4 entries Data TLB: 4-KB Pages, 4-way set associative, 64 entries Instruction TLB: 4-KByte pages, 8-way set associative, 128 entries L2 TLB: 1-MB, 4-way set associative, 64-byte line size Shared 2nd-Level TLB: 4-KByte / 2-MB pages, 8-way associative, 1024 entries

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

Supported instructions	
Instruction set extensions	Additional Instructions
MMX	Advanced Bit manipulation

podemos mejorar su experiencia.

-44...

practica-05.pdf...

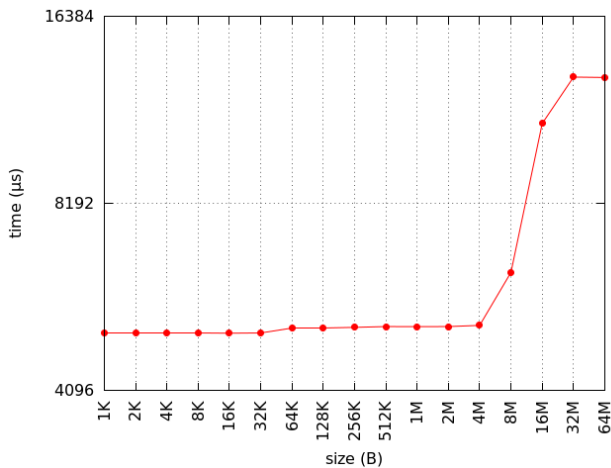
[LXTerminal]

line.cc (~/Escri...

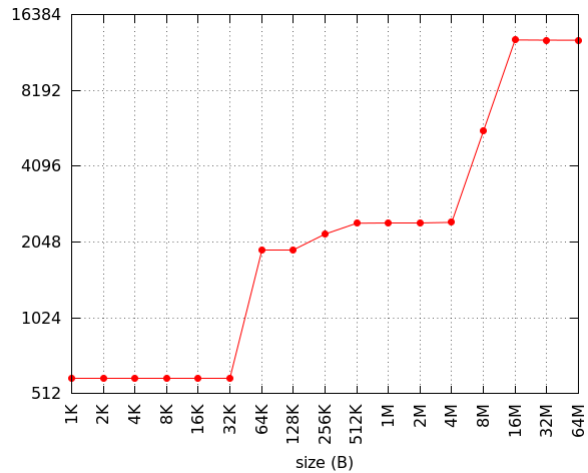
En este caso la captura que aporta la respuesta es la de make info, donde efectivamente aparece que el tamaño de cache es 64B.

Continuando la práctica anterior ahora vamos a averiguar el tamaño de la caché con otro programa. Esta vez para cada tamaño de caché crearemos un vector con ese tamaño y veremos cuando se llena la cache y necesita acceder a la siguiente de mayor tamaño. Esto son las gráficas obtenidas para el programa size.cc con optimizaciones O0, O1, O2 y Ofast:

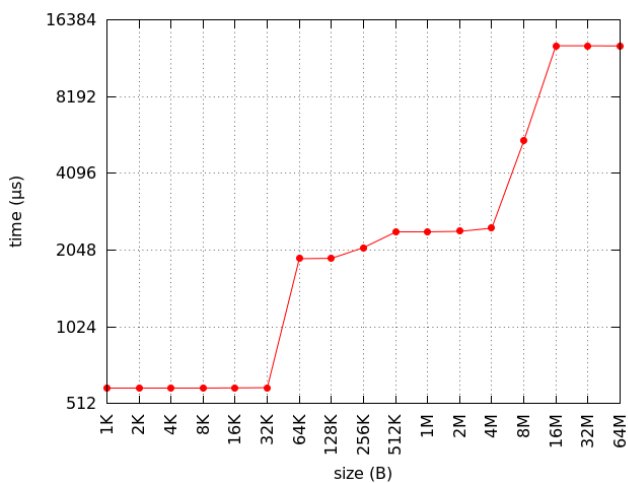
Optimización O0



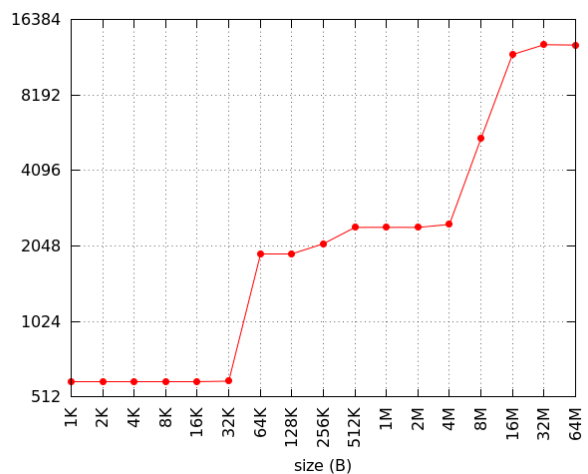
Optimización O1



Optimización O2



Optimización Ofast



Como en la práctica anterior la optimización O0 no sirve, y ahora la que mejor se aprecia los tamaños de cache es la O1. Se puede observar que a partir de 32 K hay un salto bastante notable de tiempo, eso quiere decir que tenemos una cache L1 de 32K que se ha llenado y se empieza a usar la siguiente. Al llegar a 256 ocurre lo mismo, por lo tanto tenemos una cache L2 de 256K. Finalmente cuando llegamos a 4M se aprecia otro salto mayor lo que nos dice que tenemos una cache L3 de 4M.