Práctica 6 - Estructura de Computadores

▼ **Captura de la ejecución de la orden lscpu y make info**

```
joshoc7@joshoc7-Aspire-A315-56:~$ lscpu
Arquitectura:                        x86_64
modo(s) de operación de las CPUs:    32-bit, 64-bit
Orden de los bytes:                  Little Endian
Address sizes:                       39 bits physical, 48 bits virtual
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:                              126
Nombre del modelo:                   Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz
Revisión:                            5
CPU MHz:                             1200.000
CPU MHz máx.:                        3600,0000
CPU MHz mín.:                        400,0000
BogoMIPS:                            2380.80
Virtualización:                      VT-x
Caché L1d:                           192 KiB
Caché L1i:                           128 KiB
Caché L2:                            2 MiB
Caché L3:                            6 MiB
CPU(s) del nodo NUMA 0:              0-7
Vulnerability Itlb multihit:         KVM: Mitigation: VMX disabled
Vulnerability L1tf:                  Not affected
Vulnerability Mds:                   Not affected
Vulnerability Meltdown:              Not affected
Vulnerability Spec store bypass:     Mitigation; Speculative Store Bypass disabl
                                     ed via prctl and seccomp
Vulnerability Spectre v1:            Mitigation; usercopy/swapgs barriers and __
                                     user pointer sanitization
Vulnerability Spectre v2:            Mitigation; Enhanced IBRS, IBPB conditional
                                     , RSB filling
Vulnerability Srbds:                 Not affected
Vulnerability Tsx async abort:       Not affected
Indicadores:                         fpu vme de pse tsc msr pae mce cx8 apic sep
                                      mtrr pge mca cmov pat pse36 clflush dts ac
                                     pi mmx fxsr sse sse2 ss ht tm pbe syscall n
                                     x pdpe1gb rdtscp lm constant_tsc art arch_p
                                     erfmon pebs bts rep_good nopl xtopology non
                                     stop_tsc cpuid aperfmperf tsc_known_freq pn
                                     i pclmulqdq dtes64 monitor ds_cpl vmx est t
                                     m2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_
                                     1 sse4_2 x2apic movbe popcnt tsc_deadline_t
                                     imer aes xsave avx f16c rdrand lahf_lm abm
                                     3dnowprefetch cpuid_fault epb invpcid_singl
                                     e ssbd ibrs ibpb stibp ibrs_enhanced tpr_sh
                                     adow vnmi flexpriority ept vpid ept_ad fsgs
                                     base tsc_adjust bmi1 avx2 smep bmi2 erms in
                                     vpcid avx512f avx512dq rdseed adx smap avx5
                                     12ifma clflushopt intel_pt avx512cd sha_ni
                                     avx512bw avx512vl xsaveopt xsavec xgetbv1 x
                                     saves split_lock_detect dtherm ida arat pln
                                      pts hwp hwp_notify hwp_act_window hwp_epp
                                     hwp_pkg_req avx512vbmi umip pku ospke avx51
                                     2_vbmi2 gfni vaes vpclmulqdq avx512_vnni av
                                     x512_bitalg avx512_vpopcntdq rdpid fsrm md_
                                     clear flush_l1d arch_capabilities
```

```
joshoc7@joshoc7-Aspire-A315-56:~/Escritorio/Practica 6$ make info
line size = 64B
cache size = 48K/32K/512K/6144K/
cache level = 1/1/2/3/
cache type = Data/Instruction/Unified/Unified/
```

Con la orden make info podemos observar que nuestro tamaño de línea es de 64 bytes y con la orden lscpu podemos ver que nuestra caché tiene tres niveles, siendo sus respectivos tamaños 320 KB, 2 MB y 6 MB.

▼ **Líneas de código añadidas**

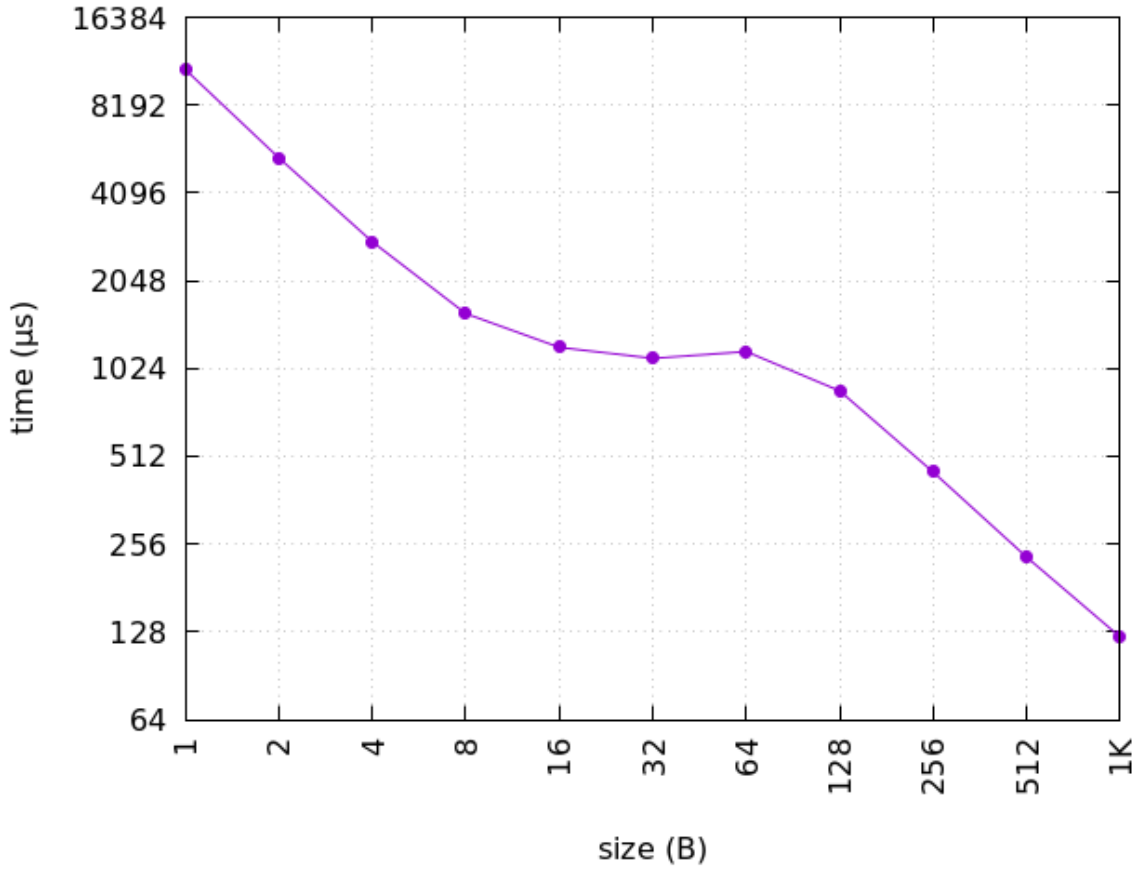
```
bytes[i] ^= 1;
```

Como en el programa line.cc sólo nos interesa el tiempo de acceso a cada elemento del vector, por lo que optamos por realizar una operación rapída, en este caso XOR, que es la propuesta por el guión de prácticas.

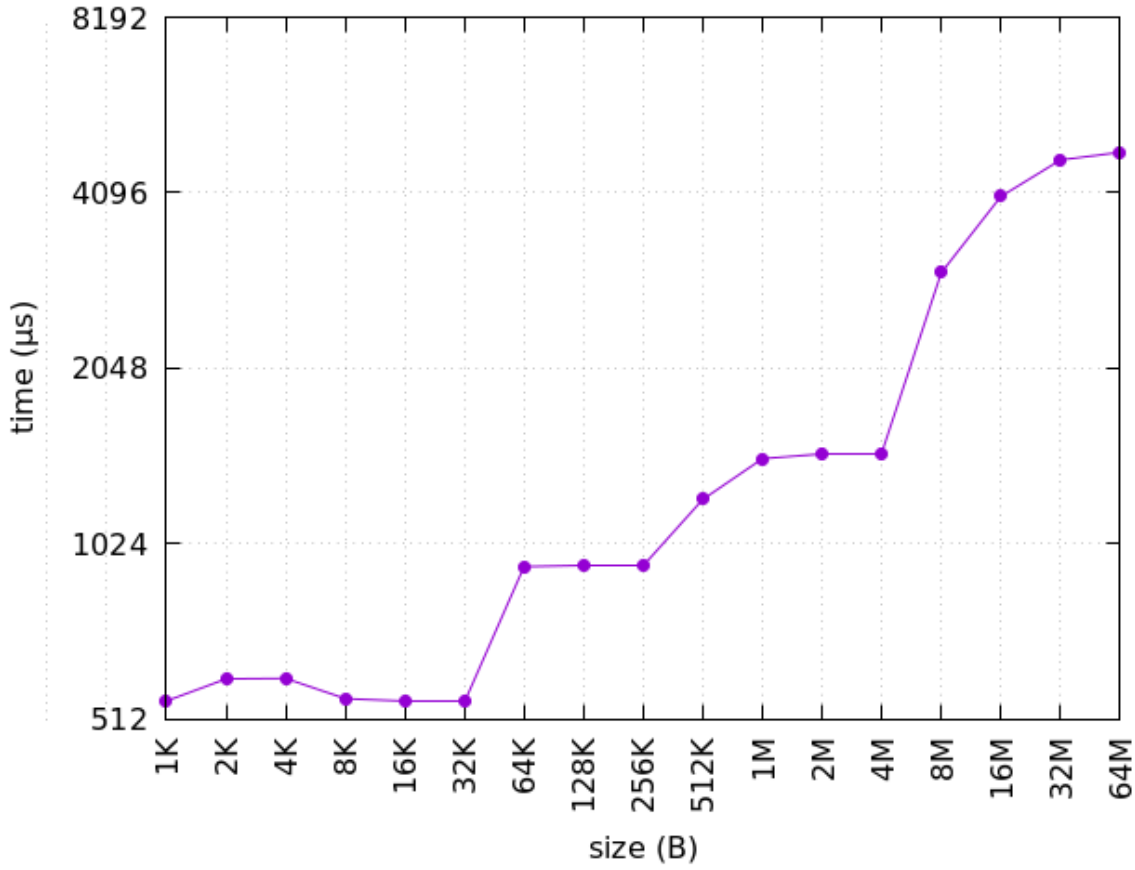
```
bytes[(i*64) & (size-1)]++ ;
```

En el programa size.cc queremos estudiar los distintos niveles de la caché, y como sabemos que el tamaño de línea es 64 bytes, con este código accedemos a posiciones del vector separadas entre sí por el tamaño de línea. Así, la caché debe cargar por cada solicitud de una entrada del vector.

▼ **Gráficas line.png y size.png con su interpretación**



Análisis primera gráfica (line.png): En la gráfica se produce un decremento notable a partir de los 64 bytes, que es nuestro tamaño de línea, lo cual quiere decir que el tiempo de acceso se reduce, habiendo menos fallos si size > 64. Esto se debe a que si es menor que 64, estamos cargando datos innecesarios. Cuando avanzamos con un valor mayor que 64, se dejan sin cargar datos del vector que no necesitamos, dando lugar a un decrecimiento del tiempo en la gráfica.



Análisis segunda gráfica (size.png): En esta gráfica podemos observar tres trozos costantes. El primero llega hasta los 32K, de forma que al acceder a una zona de memoria que no alcance ese límite, el tiempo de acceso es de 512 microsegundos. Al intentar acceder a más memoria, el tiempo de acceso pasa a ser de unos 1000 microsegundos hasta llegar a los 2 MB, deduciendo que el tamaño del nivel L2 de caché es de 2 MB. Finalmente, desde esos 2 MB, el tiempo de acceso vuelve a incrementar hasta establecerse a partir de los 8 MB, y como 8 MB - 2 MB = 6 MB, deducimos que el nivel L3 de caché es de 6 MB.

José Alberto Hoces Castro 2º Doble Grado en Ingeniería Informática y Matemáticas