

2º curso / 2º cuatr.  
Grado Ing. Inform.  
Doble Grado Ing.  
Inform. y Mat.

# Arquitectura de Computadores (AC)

## Cuaderno de prácticas.

### Bloque Práctico 0. Entorno de programación

Estudiante (nombre y apellidos): David Carrasco Chicharro

Grupo de prácticas: B1

Fecha de entrega: 04/03/2018

Fecha evaluación en clase: 05/03/2018

1. Incorpore volcados de pantalla que muestren lo que devuelve `lscpu` en `atcgird` y en su PC.

#### CAPTURAS:

```
David Carrasco Chicharro Biestudiante4@atcgird:~:2018-03-04 domingo
$ cat STDIN.o64188
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                24
On-line CPU(s) list:   0-23
Thread(s) per core:    2
Core(s) per socket:    6
Socket(s):             2
NUMA node(s):          2
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 44
Model name:            Intel(R) Xeon(R) CPU           E5645   @ 2.40GHz
Stepping:              2
CPU MHz:               1673.226
CPU max MHz:           2401.0000
CPU min MHz:           1600.0000
BogoMIPS:              4800.14
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              12288K
NUMA node0 CPU(s):     0-5,12-17
NUMA node1 CPU(s):     6-11,18-23
```

Imagen 1: `lscpu` en `atcgird`

```
David Carrasco Chicharro david@david:~/Uni/AC/Practicas/5.Realizadas/BP0:2018-02-22 jueves
$ lscpu
Architecture:          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: 2
Núcleo(s) por «socket»: 2
Socket(s):             1
Modo(s) NUMA:          1
ID de fabricante:      GenuineIntel
Familia de CPU:        6
Modelo:                61
Model name:            Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz
Revisión:              4
CPU MHz:               2195.077
CPU max MHz:           2700.0000
CPU min MHz:           500.0000
BogoMIPS:              4390.15
Virtualización:        VT-x
Caché L1d:             32K
Caché L1i:             32K
Caché L2:              256K
Caché L3:              3072K
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 pdpe1gb rdtscp lm constant_tsc arc
                        h_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 x2apic movbe popcnt
                        tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch cpuid_fault epb invpcid
                        d_single pti intel_pt tpr_shadow vnmi flexpriority ept vpid fsgsbase tsc_adjust bmi1 avx2 smep
                        bmi2 erms invpcid rdseed adx smap xsaveopt dtherm ida arat pln pts
```

Imagen 2: `lscpu` PC local

Conteste a las siguientes preguntas:

- a. ¿Cuántos cores físicos y cuántos cores lógicos tiene atcgrid de prácticas o su PC?

**RESPUESTA:** Mi PC tiene 4 cores físicos y 2 cores lógicos.

- b. ¿Cuántos cores físicos y cuántos cores lógicos tiene un nodo de atcgrid?

**RESPUESTA:** Un nodo de atcgrid tiene 12 cores físicos y 24 cores lógicos

2. En el Listado 1 se puede ver un código fuente C que calcula la suma de dos vectores y en el Listado 2 una versión con C++:

```
v3 = v1 + v2; v3(i) = v1(i) + v2(i), i=0,...N-1
```

Los códigos utilizan directivas del compilador para fijar el tipo de variable de los vectores (v1, v2 y v3). En los comentarios que hay al principio de los códigos se indica cómo hay que compilarlos. Los vectores pueden ser:

- Variables locales: descomentando en el código `#define VECTOR_LOCAL` y comentando `#define VECTOR_GLOBAL` y `#define VECTOR_DYNAMIC`
- Variables globales: descomentando `#define VECTOR_GLOBAL` y comentando `#define VECTOR_LOCAL` y `#define VECTOR_DYNAMIC`
- Variables dinámicas: descomentando `#define VECTOR_DYNAMIC` y comentando `#define VECTOR_LOCAL` y `#define VECTOR_GLOBAL`. Si se usan los códigos tal y como están en Listado 1 y Listado 2, sin hacer ningún cambio, los vectores (v1, v2 y v3) serán variables dinámicas.

Por tanto, se debe definir sólo una de las siguientes constantes: `VECTOR_LOCAL`, `VECTOR_GLOBAL` o `VECTOR_DYNAMIC`.

- a. En los dos códigos (Listado 1 y Listado 2) se utiliza la función `clock_gettime()` para obtener el tiempo de ejecución del trozo de código que calcula la suma de vectores. En el código se imprime la variable `ncgt`, ¿qué contiene esta variable? ¿qué información devuelve exactamente la función `clock_gettime()`? ¿en qué estructura de datos devuelve `clock_gettime()` la información (indicar el tipo de estructura de datos y describir la estructura de datos)?

**RESPUESTA:**

La variable `ncgt` contiene el tiempo que tardan en sumarse v1 y v2.

La información que devuelve la función `clock_gettime()` es el tiempo real del sistema, contado desde el llamado “*Unix Epoch*”, cuya cuenta comenzó el 01/01/1970 – 00:00:00 UTC.

La estructura de datos en la que devuelve `clock_gettime()` la información es `timespec`, que contiene dos variables:

`tv_sec` de tipo `time_t`, que almacena los segundos

`tv_nsec` de tipo `long`, que almacena los nanosegundos

- b. Escribir en el cuaderno de prácticas las diferencias que hay entre el código fuente C y el código fuente C++ para la suma de vectores.

**RESPUESTA:**

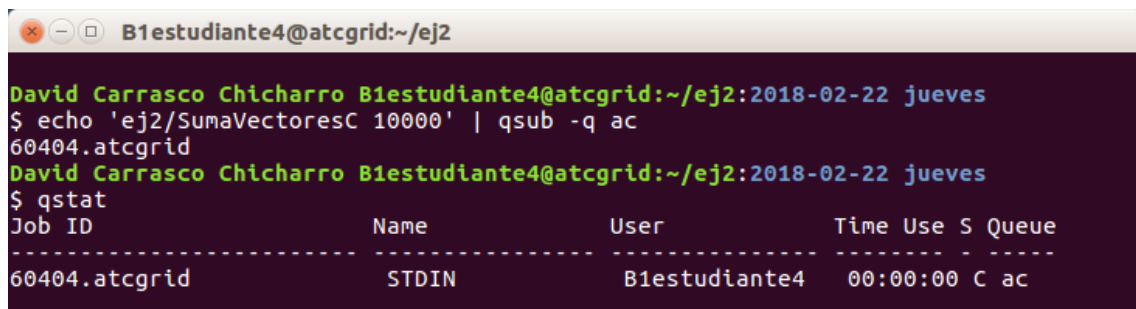
Descripción diferencia	En C	En C++
Declaración de constantes	Empiezan con #define	Empiezan con const
Salida estándar por pantalla	printf( );	cout << "... ";
Declaración de variables dinámicas	malloc	new
Liberar variables dinámicas	free	Delete[]
Cronometrar tiempo	clock_gettime(CLOCK_REALTIME, &var)	auto start = std::chrono::high_resolution_clock::now();

3. Generar el ejecutable del código fuente C del Listado 1 para vectores locales (para ello antes de compilar debe descomentar la definición de VECTOR\_LOCAL y comentar las definiciones de VECTOR\_GLOBAL y VECTOR\_DYNAMIC). Incorporar volcados de pantalla que demuestren la ejecución correcta en atcgrid o en su PC.

**RESPUESTA:**

Orden de compilación: \$ gcc -O2 SumaVectoresC.c -o SumaVectoresC -lrt

Se exporta el fichero ejecutable SumaVectoresC a atcgrid mediante sftp y se ejecuta:



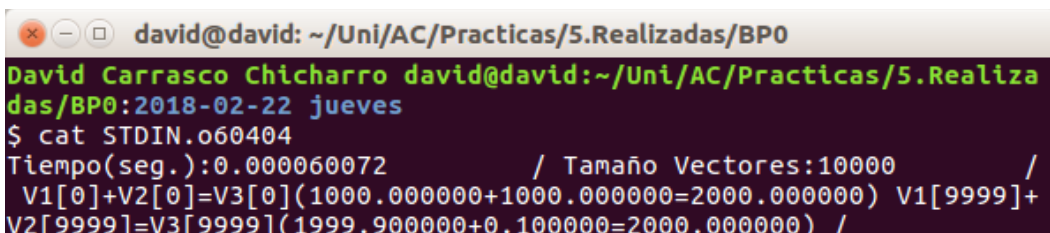
```

B1estudiante4@atcgrid:~/ej2
David Carrasco Chicharro B1estudiante4@atcgrid:~/ej2:2018-02-22 jueves
$ echo 'ej2/SumaVectoresC 10000' | qsub -q ac
60404.atcgrid
David Carrasco Chicharro B1estudiante4@atcgrid:~/ej2:2018-02-22 jueves
$ qstat
Job ID              Name              User              Time Use S Queue
-----
60404.atcgrid      STDIN              B1estudiante4     00:00:00 C ac

```

Imagen 3: Ejecución en atcgrid

Se visualiza en el terminal local el resultado de la ejecución en atcgrid



```

david@david: ~/Uni/AC/Practicas/5.Realizadas/BP0
David Carrasco Chicharro david@david:~/Uni/AC/Practicas/5.Realiza
das/BP0:2018-02-22 jueves
$ cat STDIN.o60404
Tiempo(seg.):0.000060072          / Tamaño Vectores:10000          /
V1[0]+V2[0]=V3[0](1000.000000+1000.000000=2000.000000) V1[9999]+
V2[9999]=V3[9999](1999.900000+0.100000=2000.000000) /

```

Imagen 4: Resultado de la ejecución

Mediante la orden cat se puede observar que, tal y como se esperaba, el resultado es correcto.

4. Ejecutar en atcgrid el código generado en el apartado anterior usando el script del Listado 3. Generar el ejecutable usando la opción de optimización `-O2` tal y como se indica en el comentario que hay al principio del programa. Ejecutar el código también en su PC para los mismos tamaños. ¿Se obtiene error para alguno de los tamaños? En caso afirmativo, ¿a qué se debe este error? (Incorporar volcados de pantalla)

**RESPUESTA:**

Da error tanto en atcgrid como en mi PC para tamaños del vector iguales o superiores a 524288 elementos, tal y como se puede ver en las capturas adjuntas. Esto se debe a que los vectores son locales y la reserva de memoria se realiza en la pila, cuyo tamaño no es muy elevado; al superarlo no caben más valores y se produce un error. Si fueran globales o dinámicos no daría el error.

```
David Carrasco Chicharro Blestudiente4@atcgrid:~/ej4:2018-02-26 lunes
$ cat SumaVectoresC_vlocales.e61461
/var/lib/torque/mom_priv/jobs/61461.atcgrid.SC: line 20: 6431 Segmentation fault
core dumped) $PBS_0 WORKDIR/SumaVectoresC $N
/var/lib/torque/mom_priv/jobs/61461.atcgrid.SC: line 20: 6434 Segmentation fault
core dumped) $PBS_0 WORKDIR/SumaVectoresC $N
/var/lib/torque/mom_priv/jobs/61461.atcgrid.SC: line 20: 6437 Segmentation fault
core dumped) $PBS_0 WORKDIR/SumaVectoresC $N
/var/lib/torque/mom_priv/jobs/61461.atcgrid.SC: line 20: 6446 Segmentation fault
core dumped) $PBS_0 WORKDIR/SumaVectoresC $N
/var/lib/torque/mom_priv/jobs/61461.atcgrid.SC: line 20: 6449 Segmentation fault
core dumped) $PBS_0 WORKDIR/SumaVectoresC $N
/var/lib/torque/mom_priv/jobs/61461.atcgrid.SC: line 20: 6452 Segmentation fault
core dumped) $PBS_0 WORKDIR/SumaVectoresC $N
/var/lib/torque/mom_priv/jobs/61461.atcgrid.SC: line 20: 6455 Segmentation fault
core dumped) $PBS_0 WORKDIR/SumaVectoresC $N
/var/lib/torque/mom_priv/jobs/61461.atcgrid.SC: line 20: 6458 Segmentation fault
core dumped) $PBS_0 WORKDIR/SumaVectoresC $N
```

Imagen 5: Error en atcgrid

```
David Carrasco Chicharro davidcch@eil41089:~/Escritorio/Home/Practicas/5.Realizadas/BP0/ej4:2018
-02-26 lunes
$ ./SumaVectores.sh
Tiempo(seg.):0.000305722 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](6553.600000+6553
.600000=13107.200000) V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
Tiempo(seg.):0.000636551 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.200000+131
07.200000=26214.400000) V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tiempo(seg.):0.001331782 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+262
14.400000=52428.800000) V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
./SumaVectores.sh: línea 21: 3771 Violación de segmento ('core' generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 21: 3773 Violación de segmento ('core' generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 21: 3775 Violación de segmento ('core' generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 21: 3777 Violación de segmento ('core' generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 21: 3779 Violación de segmento ('core' generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 21: 3781 Violación de segmento ('core' generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 21: 3783 Violación de segmento ('core' generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 21: 3785 Violación de segmento ('core' generado) ./SumaVectoresC $N
```

Imagen 6: Error en mi PC

5. Generar los ejecutables del código fuente C para vectores globales y para dinámicos. Genere el ejecutable usando `-O2`. Ejecutar los dos códigos en atcgrid usando un script como el del Listado 3 (hay que poner en el script el nombre de los ficheros ejecutables generados en este ejercicio) para el mismo rango de tamaños utilizado en el ejercicio anterior. Ejecutar también los códigos en su PC. ¿Se obtiene error usando vectores globales o dinámicos? ¿A qué cree que es debido? (Incorporar volcados de pantalla)

**RESPUESTA:**

No da ningún error ni el programa con vectores globales ni tampoco con vectores dinámicos tanto en atcgrid como en mi PC, puesto que, a diferencia de los vectores locales, la memoria se almacena en el “*heap*” cuyo tamaño es mucho mayor que el de la pila.

```

David Carrasco Chicharro Blestudiente4@atcgrid:~/ej5:2018-02-26 lunes
$ cat SumaVectoresC_vglobales.o61486
Id. usuario del trabajo: Blestudiente4
Id. del trabajo: 61486.atcgrid
Nombre del trabajo especificado por usuario: SumaVectoresC_vglobales
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/Blestudiente4/ej5
Cola: ac
Nodos asignados al trabajo:
atcgrid1
atcgrid1
atcgrid1
atcgrid1
Tiempo(seg.):0.000327515 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0]
(6553.600000+6553.600000=13107.200000) V1[65535]+V2[65535]=V3[65535](13107.100000+0
.100000=13107.200000) /
Tiempo(seg.):0.000853576 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0]
(13107.200000+13107.200000=26214.400000) V1[131071]+V2[131071]=V3[131071](26214.300
000+0.100000=26214.400000) /
Tiempo(seg.):0.001678104 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0]
(26214.400000+26214.400000=52428.800000) V1[262143]+V2[262143]=V3[262143](52428.700
000+0.100000=52428.800000) /
Tiempo(seg.):0.002799768 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0]
(52428.800000+52428.800000=104857.600000) V1[524287]+V2[524287]=V3[524287](104857.5
00000+0.100000=104857.600000) /
Tiempo(seg.):0.005764091 / Tamaño Vectores:1048576 / V1[0]+V2[0]=V3[0]
(104857.600000+104857.600000=209715.200000) V1[1048575]+V2[1048575]=V3[1048575](209
715.100000+0.100000=209715.200000) /
Tiempo(seg.):0.011723774 / Tamaño Vectores:2097152 / V1[0]+V2[0]=V3[0]
(209715.200000+209715.200000=419430.400000) V1[2097151]+V2[2097151]=V3[2097151](419
430.300000+0.100000=419430.400000) /
Tiempo(seg.):0.023270369 / Tamaño Vectores:4194304 / V1[0]+V2[0]=V3[0]
(419430.400000+419430.400000=838860.800000) V1[4194303]+V2[4194303]=V3[4194303](838
860.700000+0.100000=838860.800000) /
Tiempo(seg.):0.046986802 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0]
(838860.800000+838860.800000=1677721.600000) V1[8388607]+V2[8388607]=V3[8388607](16
77721.500000+0.100000=1677721.600000) /
Tiempo(seg.):0.093752897 / Tamaño Vectores:16777216 / V1[0]+V2[0]=V3[0]
(1677721.600000+1677721.600000=3355443.200000) V1[16777215]+V2[16777215]=V3[1677721
5](3355443.100000+0.100000=3355443.200000) /
Tiempo(seg.):0.185047679 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0]
(3355443.200000+3355443.200000=6710886.400000) V1[33554431]+V2[33554431]=V3[3355443
1](6710886.300000+0.100000=6710886.400000) /
Tiempo(seg.):0.183930548 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0]
(3355443.200000+3355443.200000=6710886.400000) V1[33554431]+V2[33554431]=V3[3355443
1](6710886.300000+0.100000=6710886.400000) /

```

Imagen 7: Resultado ejecución vectores globales atcgrid

```

David Carrasco Chicharro Blestudiente4@atcgrid:~/ej5:2018-02-26 lunes
$ cat SumaVectoresC_vdinamicos.o61487
Id. usuario del trabajo: Blestudiente4
Id. del trabajo: 61487.atcgrid
Nombre del trabajo especificado por usuario: SumaVectoresC_vdinamicos
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/Blestudiente4/ej5
Cola: ac
Nodos asignados al trabajo:
atcgrid1
atcgrid1

```



```

atcgrid1
atcgrid1
atcgrid1
Tiempo(seg.):0.000387231 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0]
(6553.600000+6553.600000=13107.200000) V1[65535]+V2[65535]=V3[65535] (13107.100000+0.100000=13107.200000) /
Tiempo(seg.):0.000852687 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0]
(13107.200000+13107.200000=26214.400000) V1[131071]+V2[131071]=V3[131071] (26214.300000+0.100000=26214.400000) /
Tiempo(seg.):0.001685993 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0]
(26214.400000+26214.400000=52428.800000) V1[262143]+V2[262143]=V3[262143] (52428.700000+0.100000=52428.800000) /
Tiempo(seg.):0.003493714 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0]
(52428.800000+52428.800000=104857.600000) V1[524287]+V2[524287]=V3[524287] (104857.500000+0.100000=104857.600000) /
Tiempo(seg.):0.005991254 / Tamaño Vectores:1048576 / V1[0]+V2[0]=V3[0]
(104857.600000+104857.600000=209715.200000) V1[1048575]+V2[1048575]=V3[1048575] (209715.100000+0.100000=209715.200000) /
Tiempo(seg.):0.012013229 / Tamaño Vectores:2097152 / V1[0]+V2[0]=V3[0]
(209715.200000+209715.200000=419430.400000) V1[2097151]+V2[2097151]=V3[2097151] (419430.300000+0.100000=419430.400000) /
Tiempo(seg.):0.024176034 / Tamaño Vectores:4194304 / V1[0]+V2[0]=V3[0]
(419430.400000+419430.400000=838860.800000) V1[4194303]+V2[4194303]=V3[4194303] (838860.700000+0.100000=838860.800000) /
Tiempo(seg.):0.048470443 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0]
(838860.800000+838860.800000=1677721.600000) V1[8388607]+V2[8388607]=V3[8388607] (1677721.500000+0.100000=1677721.600000) /
Tiempo(seg.):0.096046612 / Tamaño Vectores:16777216 / V1[0]+V2[0]=V3[0]
(1677721.600000+1677721.600000=3355443.200000) V1[16777215]+V2[16777215]=V3[16777215] (3355443.100000+0.100000=3355443.200000) /
Tiempo(seg.):0.187249955 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0]
(3355443.200000+3355443.200000=6710886.400000) V1[33554431]+V2[33554431]=V3[33554431] (6710886.300000+0.100000=6710886.400000) /
Tiempo(seg.):0.355963147 / Tamaño Vectores:67108864 / V1[0]+V2[0]=V3[0]
(6710886.400000+6710886.400000=13421772.800000) V1[67108863]+V2[67108863]=V3[67108863] (13421772.700000+0.100000=13421772.800000) /

```

Imagen 8: Resultado ejecución vectores dinámicos atcgrid

```

David Carrasco Chicharro davidcch@ei141089:~/Escritorio/Home/Practicas/5.Realizadas/BP0/ej5:
26 lunes
$ ./SumaVectoresGlobales.sh
Tiempo(seg.):0.000378481 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0] (6553.600000+6553.600000=13107.200000) V1[65535]+V2[65535]=V3[65535] (13107.100000+0.100000=13107.200000) /
Tiempo(seg.):0.000480653 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0] (13107.200000+13107.200000=26214.400000) V1[131071]+V2[131071]=V3[131071] (26214.300000+0.100000=26214.400000) /
Tiempo(seg.):0.001518595 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0] (26214.400000+26214.400000=52428.800000) V1[262143]+V2[262143]=V3[262143] (52428.700000+0.100000=52428.800000) /
Tiempo(seg.):0.002381146 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0] (52428.800000+52428.800000=104857.600000) V1[524287]+V2[524287]=V3[524287] (104857.500000+0.100000=104857.600000) /
Tiempo(seg.):0.004269943 / Tamaño Vectores:1048576 / V1[0]+V2[0]=V3[0] (104857.600000+104857.600000=209715.200000) V1[1048575]+V2[1048575]=V3[1048575] (209715.100000+0.100000=209715.200000) /
Tiempo(seg.):0.007828335 / Tamaño Vectores:2097152 / V1[0]+V2[0]=V3[0] (209715.200000+209715.200000=419430.400000) V1[2097151]+V2[2097151]=V3[2097151] (419430.300000+0.100000=419430.400000) /
Tiempo(seg.):0.015118158 / Tamaño Vectores:4194304 / V1[0]+V2[0]=V3[0] (419430.400000+419430.400000=838860.800000) V1[4194303]+V2[4194303]=V3[4194303] (838860.700000+0.100000=838860.800000) /
Tiempo(seg.):0.028184535 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0] (838860.800000+838860.800000=1677721.600000) V1[8388607]+V2[8388607]=V3[8388607] (1677721.500000+0.100000=1677721.600000) /
Tiempo(seg.):0.053460635 / Tamaño Vectores:16777216 / V1[0]+V2[0]=V3[0] (1677721.600000+1677721.600000=3355443.200000) V1[16777215]+V2[16777215]=V3[16777215] (3355443.100000+0.100000=3355443.200000) /
Tiempo(seg.):0.105896071 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0] (3355443.200000+3355443.200000=6710886.400000) V1[33554431]+V2[33554431]=V3[33554431] (6710886.300000+0.100000=6710886.400000) /
Tiempo(seg.):0.105731644 / Tamaño Vectores:67108864 / V1[0]+V2[0]=V3[0] (6710886.400000+6710886.400000=13421772.800000) V1[67108863]+V2[67108863]=V3[67108863] (13421772.700000+0.100000=13421772.800000) /

```

Imagen 9: Resultado ejecución vectores globales en mi PC

```

David Carrasco Chicharro davidcch@ei141089:~/Escritorio/Home/Practicas/5.Realizadas/BP0/ej5:
2018-02-26 Lunes
$ ./SumaVectoresDinamicos.sh
Tiempo(seg.):0.000294124 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](6553.600
000+6553.600000=13107.200000) V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.2000
00) /
Tiempo(seg.):0.000609622 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.20
0000+13107.200000=26214.400000) V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214
.400000) /
Tiempo(seg.):0.001391929 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.40
0000+26214.400000=52428.800000) V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428
.800000) /
Tiempo(seg.):0.002439600 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](52428.80
0000+52428.800000=104857.600000) V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104
857.600000) /
Tiempo(seg.):0.005149050 / Tamaño Vectores:1048576 / V1[0]+V2[0]=V3[0](104857.6
00000+104857.600000=209715.200000) V1[1048575]+V2[1048575]=V3[1048575](209715.100000+0.10000
0=209715.200000) /
Tiempo(seg.):0.007520642 / Tamaño Vectores:2097152 / V1[0]+V2[0]=V3[0](209715.2
00000+209715.200000=419430.400000) V1[2097151]+V2[2097151]=V3[2097151](419430.300000+0.10000
0=419430.400000) /
Tiempo(seg.):0.014036458 / Tamaño Vectores:4194304 / V1[0]+V2[0]=V3[0](419430.4
00000+419430.400000=838860.800000) V1[4194303]+V2[4194303]=V3[4194303](838860.700000+0.10000
0=838860.800000) /
Tiempo(seg.):0.026249617 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](838860.8
00000+838860.800000=1677721.600000) V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.100
000=1677721.600000) /
Tiempo(seg.):0.050532020 / Tamaño Vectores:16777216 / V1[0]+V2[0]=V3[0](1677721.
600000+1677721.600000=3355443.200000) V1[16777215]+V2[16777215]=V3[16777215](3355443.100000+
0.100000=3355443.200000) /
Tiempo(seg.):0.100695733 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](3355443.
200000+3355443.200000=6710886.400000) V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+
0.100000=6710886.400000) /
Tiempo(seg.):0.201130539 / Tamaño Vectores:67108864 / V1[0]+V2[0]=V3[0](6710886.
400000+6710886.400000=13421772.800000) V1[67108863]+V2[67108863]=V3[67108863](13421772.70000
0+0.100000=13421772.800000) /

```

Imagen 10: Resultado ejecución vectores dinámicos en mi PC

6. Rellenar una tabla como la Tabla 1 para atcgrid y otra para su PC con los tiempos de ejecución obtenidos en los ejercicios anteriores para el trozo de código que realiza la suma de vectores. En la columna “Bytes de un vector” hay que poner el total de bytes reservado para un vector. Ayudándose de una hoja de cálculo represente en una misma gráfica los tiempos de ejecución obtenidos en atcgrid y en su PC para vectores locales, globales y dinámicos (eje y) en función del tamaño en bytes de un vector (los valores de la segunda columna de la tabla, que están en escala logarítmica, deben estar en el eje x). Utilice escala logarítmica en el eje de ordenadas (eje y). ¿Hay diferencias en los tiempos de ejecución?

**Tabla 1:** Tiempos de ejecución en atcgrid

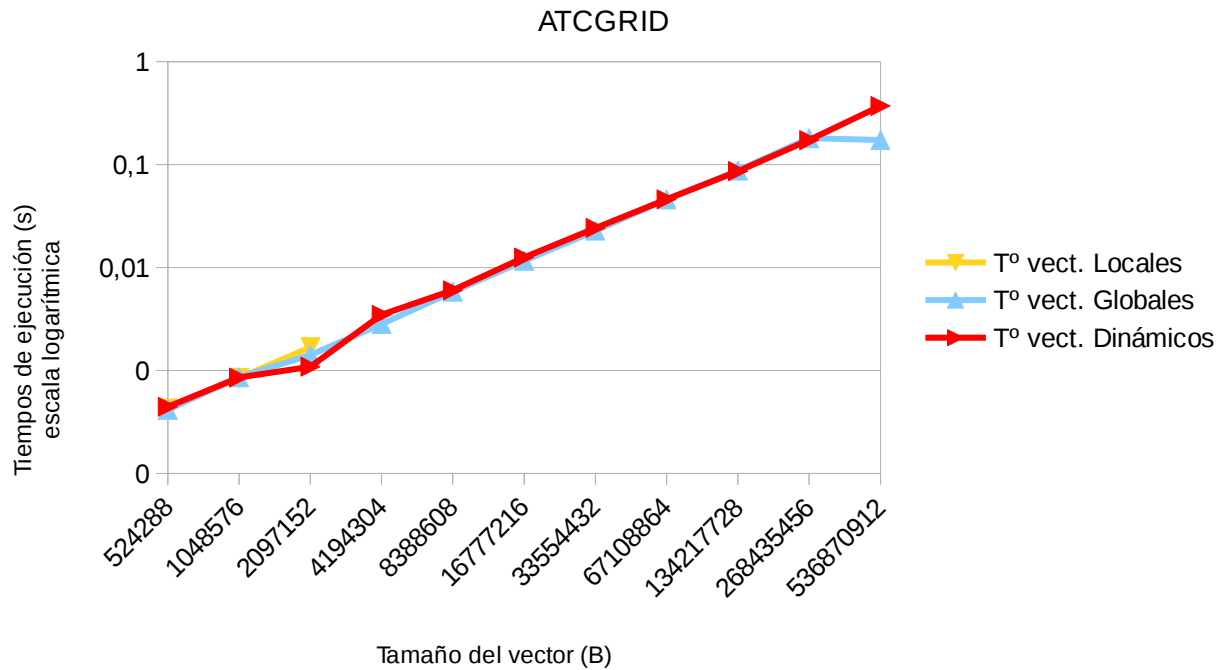
Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0.000426293	0.000415631	0.000444032
131072	1048576	0.000846751	0.000859191	0.000854546
262144	2097152	0.001683029	0.001415588	0.001088729
524288	4194304		0.002830192	0.003478085
1048576	8388608		0.005816500	0.006012478
2097152	16777216		0.011639127	0.012525106
4194304	33554432		0.022792980	0.024239064
8388608	67108864		0.045918141	0.046080817
16777216	134217728		0.087544028	0.086363888
33554432	268435456		0.180384394	0.172661483
67108864	536870912		0.172776820	0.371013431

**Tabla 2:** Tiempos de ejecución en mi PC

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0.000704913	0.000731570	0.000397149
131072	1048576	0.000917651	0.000736883	0.000842992
262144	2097152	0.001475570	0.001423187	0.001507079
524288	4194304		0.003078495	0.002832654
1048576	8388608		0.004855659	0.005444448
2097152	16777216		0.009823883	0.009661454
4194304	33554432		0.018814281	0.018872486
8388608	67108864		0.038489129	0.037797072
16777216	134217728		0.072112088	0.076308093
33554432	268435456		0.147177517	0.152789751
67108864	536870912		0.148135747	0.294691852

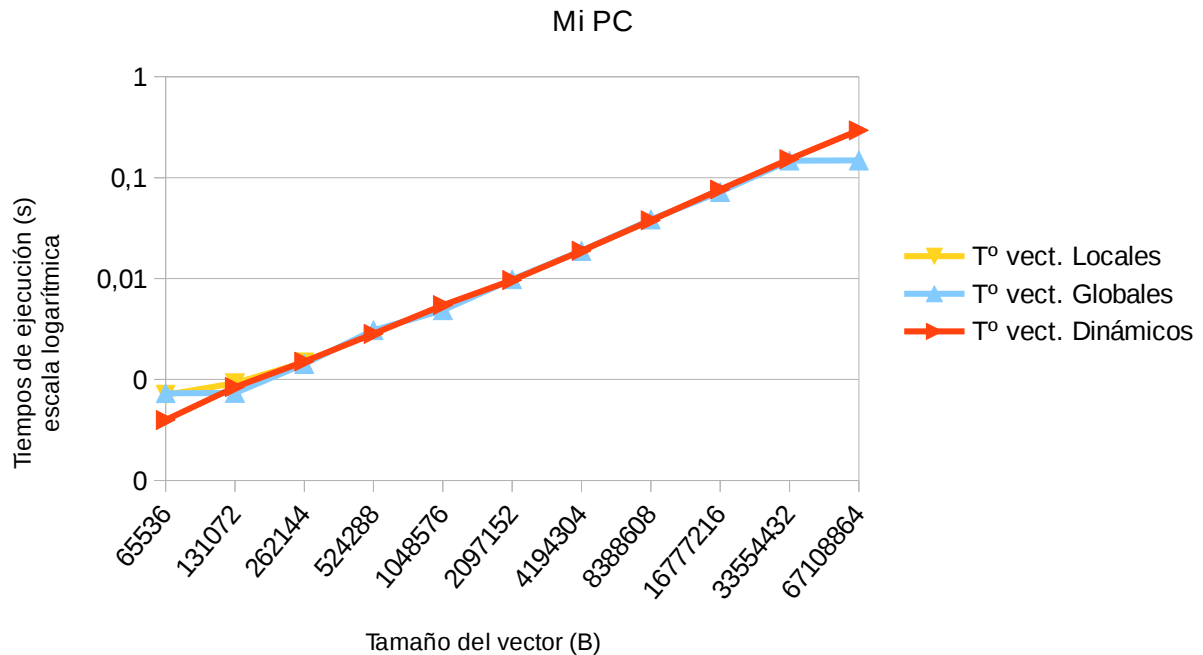
**RESPUESTA:**

Apenas se aprecia ninguna diferencia en los tiempos de ejecución de la suma de vectores (ya sean locales, globales o dinámicos), ya sea ejecutado el programa en atcgrid o en mi PC.

**Tiempos de ejecución para la suma de vectores**



## Tiempos de ejecución para la suma de vectores



7. Modificar el código fuente C para que el límite de los vectores cuando se declaran como variables globales sea igual al máximo número que se puede almacenar en la variable N ( $\text{MAX}=2^{32}-1$ ). Generar el ejecutable usando variables globales. ¿Qué ocurre? ¿A qué es debido? Razone además por qué el máximo número que se puede almacenar en N es  $2^{32}-1$ .

```
David Carrasco Chicharro david@david:~/Uni/AC/Practicas/5.Realizadas/BP0/ej7:2018-02-26 lunes
$ gcc -O2 SumaVectoresC.c -o SumaVectoresC -lrt
/tmp/cc85okED.o: En la función 'main':
SumaVectoresC.c:(.text.startup+0x79): reubicación truncada para ajustar: R_X86_64_32S
contra el símbolo 'v2' definido en la sección COMMON en /tmp/cc85okED.o
SumaVectoresC.c:(.text.startup+0xc0): reubicación truncada para ajustar: R_X86_64_32S
contra el símbolo 'v2' definido en la sección COMMON en /tmp/cc85okED.o
SumaVectoresC.c:(.text.startup+0xc8): reubicación truncada para ajustar: R_X86_64_32S
contra el símbolo 'v3' definido en la sección COMMON en /tmp/cc85okED.o
SumaVectoresC.c:(.text.startup+0xfc): reubicación truncada para ajustar: R_X86_64_32S
contra el símbolo 'v3' definido en la sección COMMON en /tmp/cc85okED.o
SumaVectoresC.c:(.text.startup+0x115): reubicación truncada para ajustar: R_X86_64_32
S contra el símbolo 'v2' definido en la sección COMMON en /tmp/cc85okED.o
SumaVectoresC.c:(.text.startup+0x12b): reubicación truncada para ajustar: R_X86_64_PC
32 contra el símbolo 'v3' definido en la sección COMMON en /tmp/cc85okED.o
SumaVectoresC.c:(.text.startup+0x135): reubicación truncada para ajustar: R_X86_64_PC
32 contra el símbolo 'v2' definido en la sección COMMON en /tmp/cc85okED.o
collect2: error: ld returned 1 exit status
```

### RESPUESTA:

Al compilar vectores estáticos se produce un error con el enlazador debido a que el objetivo del esquema de direccionamiento relativo es mayor de lo que puede admitirse con el desplazamiento de 32 bits del modo de direccionamiento relativo elegido.

El máximo número almacenable es  $2^{32}-1$  porque dicho valor es el mayor entero positivo con signo de 32 bits.