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):
Grupo de prácticas:
Fecha de entrega:
Fecha evaluación en clase:

---

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

CAPTURAS:

1. PC:

```
[Sergio Aguilera Ramirez sergioaguilera@ei143091:~] 2018-02-27 martes
$lscpu
Arquitectura:          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:1
Núcleo(s) por «socket»:4
Socket(s):             1
Modo(s) NUMA:          1
ID de fabricante:      GenuineIntel
Familia de CPU:        6
Modelo:                42
Model name:            Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz
Revisión:             7
CPU MHz:               1640.252
CPU max MHz:           3400,0000
CPU min MHz:           1600,0000
BogoMIPS:              6220.07
Virtualización:       VT-x
Caché L1d:            32K
Caché L1i:            32K
Caché L2:             256K
Caché L3:             6144K
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 rdtscp
 lm constant_tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc aperf
mperf eagerfpu pni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 cx16 xt
pr pdcm pcid sse4_1 sse4_2 x2apic popcnt tsc_deadline_timer aes xsave avx lahf_l
m epb tpr_shadow vnmi flexpriority ept vpid xsaveopt dtherm ida arat pln pts
```

2. atcgrid:

```
[Sergio Aguilera Ramirez sergioaguilera@ei143091:~] 2018-02-27 martes
$ls
Descargas  Documentos  Escritorio  Imágenes  Música  Plantillas  Público  STDIN.e61953  STDIN.o61953  Vídeos
[Sergio Aguilera Ramirez sergioaguilera@ei143091:~] 2018-02-27 martes
$cat STDIN.o61953
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:             1602.695
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
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 arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf pni dtes64 monitor ds_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm pcid dca sse4_1 s
se4_2 popcnt lahf_lm epb pti retpoline tpr_shadow vnmi flexpriority ept vpid dtherm ida arat
```

3. Conteste a las siguientes preguntas:

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

   RESPUESTA:

   núcleos lógicos: 4

   núcleos físicos: 4

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

   RESPUESTA:

   núcleos lógicos: 24

   núcleos físicos: 12

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 funcion clock_gettime() permite obtener el valor de un momento determinado, este tiempos se guardan en las variables cgt1(principio) y cgt2 (final) , si restamos cgt2 menos cgt1 obtenemos el tiempo que a tardado en hacer un ciclo, esta resta se almacena en ncgt.

Las varibles cgt1 y cgt2 estan declradas como timespec (es una estructura que contiene dos variables tc_sec y tv_nsec).

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++
Reserva de memoria	Se utiliza la función malloc() para reservar memoria	Se utiliza la declaración new double[] para reservar memoria en una variable
Imprimir	Utilización de la función printf() para la salida de datos	Los datos se imprimen por el operando de insercción de flujo
Liberación de memoria	Usamos la función free() para liberar la memoria	Usamos la función delete[] para liberar la memoria

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:

```
[Sergio Aguilera Ramirez sergioaguilera@ei143091:~/Escritorio] 2018-02-27 martes
$gcc -O2 SumaVectoresC.c -o SumaVectores -lrt
[Sergio Aguilera Ramirez sergioaguilera@ei143091:~/Escritorio] 2018-02-27 martes
$./SumaVectores  2000
Tiempo(seg.):0.000008858        / Tamaño Vectores:2000 / V1[0]+V2[0]=V3[0](200.
000000+200.000000=400.000000) / / V1[1999]+V2[1999]=V3[1999](399.900000+0.100000
=400.000000) /
[Sergio Aguilera Ramirez sergioaguilera@ei143091:~/Escritorio] 2018-02-27 martes
$./SumaVectores  8888
Tiempo(seg.):0.000038602        / Tamaño Vectores:8888 / V1[0]+V2[0]=V3[0](888.
800000+888.800000=1777.600000) / / V1[8887]+V2[8887]=V3[8887](1777.500000+0.1000
00=1777.600000) /
[Sergio Aguilera Ramirez sergioaguilera@ei143091:~/Escritorio] 2018-02-27 martes
$./SumaVectores  9999
Tiempo(seg.):0.000046673        / Tamaño Vectores:9999 / V1[0]+V2[0]=V3[0](999.
900000+999.900000=1999.800000) / / V1[9998]+V2[9998]=V3[9998](1999.700000+0.1000
00=1999.800000) /
```

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:

atcgrid:

```
Tiempo(seg.):0.000428126        / 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.):0.000840532        / Tamaño Vectores:131072      / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / / V1[131071]+V2[131071]=V3[131071](26214.30
0000+0.100000=26214.400000) /
Tiempo(seg.):0.001702508        / Tamaño Vectores:262144      / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]+V2[262143]=V3[262143](52428.70
0000+0.100000=52428.800000) /
Tiempo(seg.):0.002680909        / 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.):0.006309340        / Tamaño Vectores:1048576     / V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / / V1[1048575]+V2[1048575]=V3[1048575](20
9715.100000+0.100000=209715.200000) /
Tiempo(seg.):0.011991302        / Tamaño Vectores:2097152     / V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / / V1[2097151]+V2[2097151]=V3[2097151](41
9430.300000+0.100000=419430.400000) /
Tiempo(seg.):0.024207139        / Tamaño Vectores:4194304     / V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / / V1[4194303]+V2[4194303]=V3[4194303](83
8860.700000+0.100000=838860.800000) /
Tiempo(seg.):0.045145511        / Tamaño Vectores:8388608     / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / / V1[8388607]+V2[8388607]=V3[8388607](1
677721.500000+0.100000=1677721.600000) /
Tiempo(seg.):0.093513048        / Tamaño Vectores:16777216    / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / / V1[16777215]+V2[16777215]=V3[167772
15](3355443.100000+0.100000=3355443.200000) /
Tiempo(seg.):0.186619807        / Tamaño Vectores:33554432    / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[335544
31](6710886.300000+0.100000=6710886.400000) /
Tiempo(seg.):0.181766114        / Tamaño Vectores:33554432    / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[335544
31](6710886.300000+0.100000=6710886.400000) /
[Sergio Aguilera Ramirez sergioaguilera@ei143091:~/Escritorio] 2018-02-27 martes
```

1. **PC:**

```
[Sergio Aguilera Ramirez sergioaguilera@ei143091:~/Escritorio] 2018-02-27 martes
$./SumaVectores.sh
Tiempo(seg.):0.000377894         / Tamaño Vectores:65536        / V1[0]+V2[0]=V3
[0](6553.600000+6553.600000=13107.200000) / / V1[65535]+V2[65535]=V3[65535](1310
7.100000+0.100000=13107.200000) /
Tiempo(seg.):0.000456947         / 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.):0.001365350         / 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.):0.002414517         / 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.):0.005195055         / Tamaño Vectores:1048576      / V1[0]+V2[0]=V3
[0](104857.600000+104857.600000=209715.200000) / / V1[1048575]+V2[1048575]=V3[10
48575](209715.100000+0.100000=209715.200000) /
Tiempo(seg.):0.007772743         / Tamaño Vectores:2097152      / V1[0]+V2[0]=V3
[0](209715.200000+209715.200000=419430.400000) / / V1[2097151]+V2[2097151]=V3[20
97151](419430.300000+0.100000=419430.400000) /
Tiempo(seg.):0.014771415         / Tamaño Vectores:4194304      / V1[0]+V2[0]=V3
[0](419430.400000+419430.400000=838860.800000) / / V1[4194303]+V2[4194303]=V3[41
94303](838860.700000+0.100000=838860.800000) /
Tiempo(seg.):0.027711543         / Tamaño Vectores:8388608      / V1[0]+V2[0]=V3
[0](838860.800000+838860.800000=1677721.600000) / / V1[8388607]+V2[8388607]=V3[8
388607](1677721.500000+0.100000=1677721.600000) /
Tiempo(seg.):0.053150375         / 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.):0.105775105         / 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.):0.105251879         / 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) /
[Sergio Aguilera Ramirez sergioaguilera@ei143091:~/Escritorio] 2018-02-27 martes
```

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:

atcgrid:

Globales:

```
[A2estudiante1@atcgrid ~]$ qsub SumaVectoresGlobales.sh
62055.atcgrid
[A2estudiante1@atcgrid ~]$ qstat
Job ID                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
62055.atcgrid              ...resC_vlocales A2estudiante1   00:00:00 C ac
[A2estudiante1@atcgrid ~]$ ls -lag
total 112
drwx------    5 A2estudiante1  4096 feb 27 11:09 .
drwxr-xr-x. 504 root          20480 feb 19 12:43 ..
-rw-------    1 A2estudiante1  5178 feb 26 16:47 .bash_history
-rw-r--r--    1 A2estudiante1    18 ene 16  2015 .bash_logout
-rw-r--r--    1 A2estudiante1   193 ene 16  2015 .bash_profile
-rw-r--r--    1 A2estudiante1   231 ene 16  2015 .bashrc
drwxr-xr-x    3 A2estudiante1  4096 feb 26  2016 .local
drwxr-xr-x    4 A2estudiante1  4096 ene 30  2015 .mozilla
drwxr-xr-x    2 A2estudiante1  4096 feb  5  2015 .ssh
-rwxr-xr-x    1 A2estudiante1  8888 feb 27 10:34 SumaVectores
-rw-------    1 A2estudiante1     0 feb 27 11:09 SumaVectoresC_vlocales.e62055
-rw-------    1 A2estudiante1  2632 feb 27 11:09 SumaVectoresC_vlocales.o62055
-rwxr-xr-x    1 A2estudiante1  8992 feb 27 10:17 SumaVectoresDinamicos
-rw-r--r--    1 A2estudiante1   751 feb 27 10:19 SumaVectoresDinamicos.sh
-rwxr-xr-x    1 A2estudiante1  8968 feb 27 10:16 SumaVectoresGlobales
-rw-r--r--    1 A2estudiante1   750 feb 27 10:19 SumaVectoresGlobales.sh
-rwxr-xr-x    1 A2estudiante1   750 feb 27 10:35 SumaVectores.sh
-rw-------    1 A2estudiante1    60 feb 26  2016 .Xauthority
```

```
Tiempo(seg.):0.000441425        / 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.):0.000859731        / Tamaño Vectores:131072      / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / / V1[131071]+V2[131071]=V3[131071](26214.30
0000+0.100000=26214.400000) /
Tiempo(seg.):0.001714935        / Tamaño Vectores:262144      / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]+V2[262143]=V3[262143](52428.70
0000+0.100000=52428.800000) /
Tiempo(seg.):0.002366087        / 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.):0.006161330        / Tamaño Vectores:1048576     / V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / / V1[1048575]+V2[1048575]=V3[1048575](20
9715.100000+0.100000=209715.200000) /
Tiempo(seg.):0.011882130        / Tamaño Vectores:2097152     / V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / / V1[2097151]+V2[2097151]=V3[2097151](41
9430.300000+0.100000=419430.400000) /
Tiempo(seg.):0.023614501        / Tamaño Vectores:4194304     / V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / / V1[4194303]+V2[4194303]=V3[4194303](83
8860.700000+0.100000=838860.800000) /
Tiempo(seg.):0.046713575        / Tamaño Vectores:8388608     / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / / V1[8388607]+V2[8388607]=V3[8388607](1
677721.500000+0.100000=1677721.600000) /
Tiempo(seg.):0.091814574        / Tamaño Vectores:16777216    / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / / V1[16777215]+V2[16777215]=V3[167772
15](3355443.100000+0.100000=3355443.200000) /
Tiempo(seg.):0.184998031        / Tamaño Vectores:33554432    / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[335544
31](6710886.300000+0.100000=6710886.400000) /
Tiempo(seg.):0.185475703        / Tamaño Vectores:33554432    / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[335544
31](6710886.300000+0.100000=6710886.400000) /
[Sergio Aguilera Ramirez sergioaguilera@ei143091:~/Escritorio] 2018-02-27 martes
```

Dinamicos:

```
[A2estudiante1@atcgrid ~]$ qsub SumaVectoresDinamicos.sh
62057.atcgrid
[A2estudiante1@atcgrid ~]$ qstat
Job ID                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
62057.atcgrid              ...resC_vlocales A2estudiante1   00:00:00 C ac
[A2estudiante1@atcgrid ~]$ ls -lag
total 112
drwx------    5 A2estudiante1  4096 feb 27 11:10 .
drwxr-xr-x. 504 root          20480 feb 19 12:43 ..
-rw-------    1 A2estudiante1  5178 feb 26 16:47 .bash_history
-rw-r--r--    1 A2estudiante1    18 ene 16  2015 .bash_logout
-rw-r--r--    1 A2estudiante1   193 ene 16  2015 .bash_profile
-rw-r--r--    1 A2estudiante1   231 ene 16  2015 .bashrc
drwxr-xr-x    3 A2estudiante1  4096 feb 26  2016 .local
drwxr-xr-x    4 A2estudiante1  4096 ene 30  2015 .mozilla
drwxr-xr-x    2 A2estudiante1  4096 feb  5  2015 .ssh
-rwxr-xr-x    1 A2estudiante1  8888 feb 27 10:34 SumaVectores
-rw-------    1 A2estudiante1     0 feb 27 11:10 SumaVectoresC_vlocales.e62057
-rw-------    1 A2estudiante1  2635 feb 27 11:10 SumaVectoresC_vlocales.o62057
-rwxr-xr-x    1 A2estudiante1  8992 feb 27 10:17 SumaVectoresDinamicos
-rw-r--r--    1 A2estudiante1   751 feb 27 10:19 SumaVectoresDinamicos.sh
-rwxr-xr-x    1 A2estudiante1  8968 feb 27 10:16 SumaVectoresGlobales
-rw-r--r--    1 A2estudiante1   750 feb 27 10:19 SumaVectoresGlobales.sh
-rwxr-xr-x    1 A2estudiante1   750 feb 27 10:35 SumaVectores.sh
-rw-------    1 A2estudiante1    60 feb 26  2016 .Xauthority
```

```
Tiempo(seg.):0.000275072        / 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.):0.000846484        / Tamaño Vectores:131072      / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / / V1[131071]+V2[131071]=V3[131071](26214.30
0000+0.100000=26214.400000) /
Tiempo(seg.):0.001415441        / Tamaño Vectores:262144      / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]+V2[262143]=V3[262143](52428.70
0000+0.100000=52428.800000) /
Tiempo(seg.):0.002834734        / 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.):0.005912146        / Tamaño Vectores:1048576     / V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / / V1[1048575]+V2[1048575]=V3[1048575](20
9715.100000+0.100000=209715.200000) /
Tiempo(seg.):0.011931191        / Tamaño Vectores:2097152     / V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / / V1[2097151]+V2[2097151]=V3[2097151](41
9430.300000+0.100000=419430.400000) /
Tiempo(seg.):0.023784229        / Tamaño Vectores:4194304     / V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / / V1[4194303]+V2[4194303]=V3[4194303](83
8860.700000+0.100000=838860.800000) /
Tiempo(seg.):0.045313921        / Tamaño Vectores:8388608     / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / / V1[8388607]+V2[8388607]=V3[8388607](1
677721.500000+0.100000=1677721.600000) /
Tiempo(seg.):0.093766697        / Tamaño Vectores:16777216    / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / / V1[16777215]+V2[16777215]=V3[167772
15](3355443.100000+0.100000=3355443.200000) /
Tiempo(seg.):0.186880643        / Tamaño Vectores:33554432    / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[335544
31](6710886.300000+0.100000=6710886.400000) /
Tiempo(seg.):0.369387061        / Tamaño Vectores:67108864    / V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) / / V1[67108863]+V2[67108863]=V3[67108
863](13421772.700000+0.100000=13421772.800000) /
[Sergio Aguilera Ramirez sergioaguilera@ei143091:~/Escritorio] 2018-02-27 martes
```

PC:

Globales:

```
[Sergio Aguilera Ramirez sergioaguilera@ei143091:~/Escritorio] 2018-02-27 martes
$./SumaVectoresGlobales.sh
Tiempo(seg.):0.000349109         / 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.):0.000468692         / Tamaño Vectores:131072      / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / / V1[131071]+V2[131071]=V3[131071](26214.30
0000+0.100000=26214.400000) /
Tiempo(seg.):0.001418716         / Tamaño Vectores:262144      / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]+V2[262143]=V3[262143](52428.70
0000+0.100000=52428.800000) /
Tiempo(seg.):0.002389042         / 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.):0.004381661         / Tamaño Vectores:1048576     / V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / / V1[1048575]+V2[1048575]=V3[1048575](20
9715.100000+0.100000=209715.200000) /
Tiempo(seg.):0.007921966         / Tamaño Vectores:2097152     / V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / / V1[2097151]+V2[2097151]=V3[2097151](41
9430.300000+0.100000=419430.400000) /
Tiempo(seg.):0.015502669         / Tamaño Vectores:4194304     / V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / / V1[4194303]+V2[4194303]=V3[4194303](83
8860.700000+0.100000=838860.800000) /
Tiempo(seg.):0.028309953         / Tamaño Vectores:8388608     / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / / V1[8388607]+V2[8388607]=V3[8388607](1
677721.500000+0.100000=1677721.600000) /
Tiempo(seg.):0.053468242         / Tamaño Vectores:16777216    / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / / V1[16777215]+V2[16777215]=V3[167772
15](3355443.100000+0.100000=3355443.200000) /
Tiempo(seg.):0.105949856         / Tamaño Vectores:33554432    / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[33554
31](6710886.300000+0.100000=6710886.400000) /
Tiempo(seg.):0.106100201         / Tamaño Vectores:33554432    / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[33554
31](6710886.300000+0.100000=6710886.400000) /
```

Dinamicos:

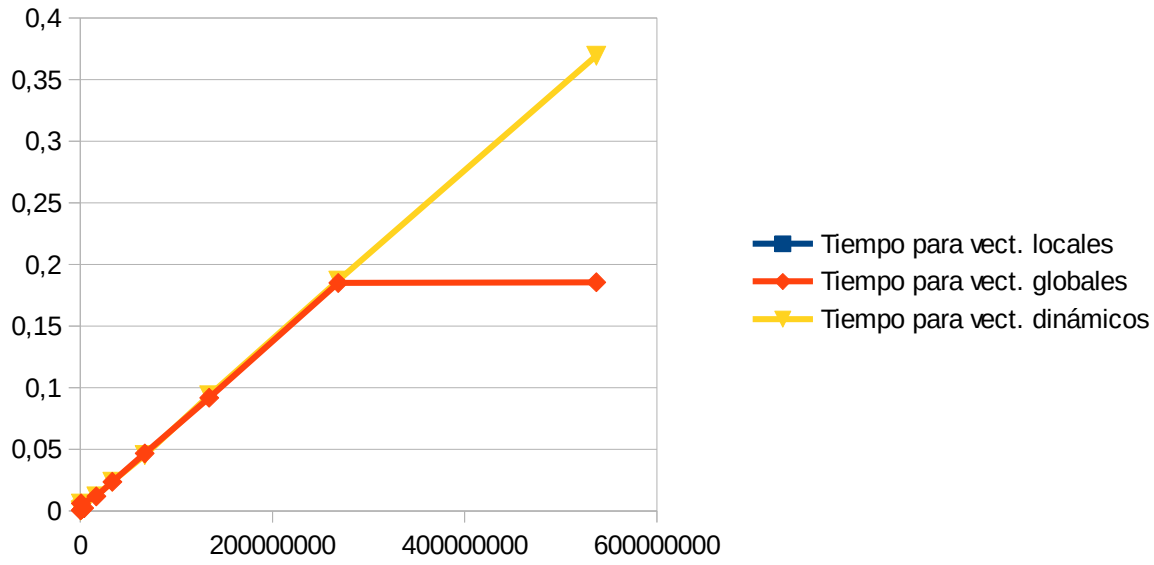
```
[Sergio Aguilera Ramirez sergioaguilera@ei143091:~/Escritorio] 2018-02-27 martes
$./SumaVectoresDinamicos.sh
Tiempo(seg.):0.000361905         / Tamaño Vectores:65536       / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / / V1[65535]+V2[65535]=V3[65535](13107.100000+
0.100000=13107.200000) /
Tiempo(seg.):0.000641744         / Tamaño Vectores:131072      / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / / V1[131071]+V2[131071]=V3[131071](26214.30
0000+0.100000=26214.400000) /
Tiempo(seg.):0.001355970         / Tamaño Vectores:262144      / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]+V2[262143]=V3[262143](52428.70
0000+0.100000=52428.800000) /
Tiempo(seg.):0.002461690         / 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.):0.004211733         / Tamaño Vectores:1048576     / V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / / V1[1048575]+V2[1048575]=V3[1048575](20
9715.100000+0.100000=209715.200000) /
Tiempo(seg.):0.007773941         / Tamaño Vectores:2097152     / V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / / V1[2097151]+V2[2097151]=V3[2097151](41
9430.300000+0.100000=419430.400000) /
Tiempo(seg.):0.014385780         / Tamaño Vectores:4194304     / V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / / V1[4194303]+V2[4194303]=V3[4194303](83
8860.700000+0.100000=838860.800000) /
Tiempo(seg.):0.026935520         / Tamaño Vectores:8388608     / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / / V1[8388607]+V2[8388607]=V3[8388607](1
677721.500000+0.100000=1677721.600000) /
Tiempo(seg.):0.053510314         / Tamaño Vectores:16777216    / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / / V1[16777215]+V2[16777215]=V3[167772
15](3355443.100000+0.100000=3355443.200000) /
Tiempo(seg.):0.101723833         / Tamaño Vectores:33554432    / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[335544
31](6710886.300000+0.100000=6710886.400000) /
Tiempo(seg.):0.210136764         / Tamaño Vectores:67108864    / V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) / / V1[67108863]+V2[67108863]=V3[67108
863](13421772.700000+0.100000=13421772.800000) /
```

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?

RESPUESTA:

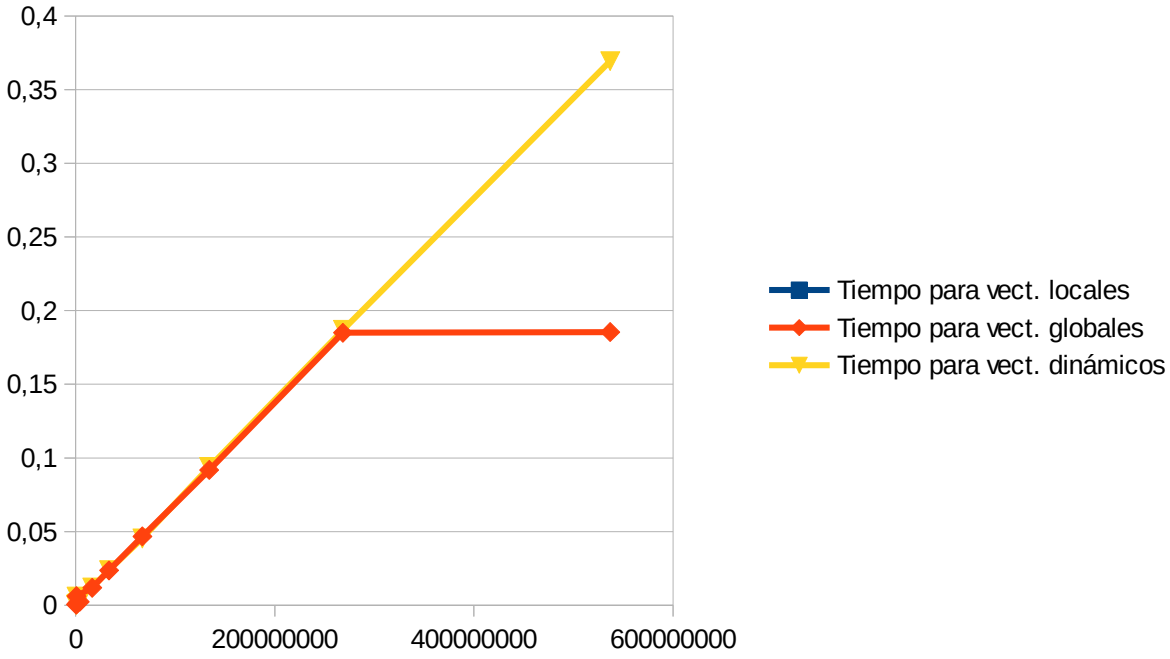
Atcgrid:

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0.000395877	0,000441425	0,000275072
131072	1048576	0.000686455	0,000859731	0,000846484
262144	2097152	0.001361396	0,001714935	0,001415441
524288	4194304		0,002366087	0,002834734
104857	838856		0,006161330	0,005912146
2097152	16777216		0,011882130	0,011931191
4194304	33554432		0,023614501	0,023784229
8388608	67108864		0,046713575	0,045313921
16777216	134217728		0,091814574	0,093766697
33554432	268435456		0,184998031	0,186880643
67108864	536870912		0,185475703	0,369387061



PC:

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0.000249858	0,000349109	0,000361905
131072	1048576	0.000503466	0,000468692	0,000641744
262144	2097152	0.001054192	0,001418716	0,001355970
524288	4194304		0,002389042	0,002461690
1048576	8388608		0,004381661	0.004211733
2097152	16777216		0,007921966	0,007773941
4194304	33554432		0,015502669	0,014385780
8388608	67108864		0,028309953	0,026935520
16777216	134217728		0,053468242	0,053510314
33554432	268435456		0,105949856	0,101723833
67108864	536870912		0,106100201	0,210136764



Solución: Si existen diferencias en los tiempos de ejecución, y que en la ejecución de mi ordenador local es un poco mas rápido que la ejecución de los mismos códigos en el nodo front-end, esto se debe a que en el nodo existen menos núcleos por lo que esto hace que valla mas lentos.

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 (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.

RESPUESTA:
cuando ejecutamos el comando de compilación nos da un error, esto se debe ha que hay un desbordamiento, ya que 2^32-1 es el numero máximo que podemos representar en un entero de 32 bits por lo cual el tamaño definido es demasiado grande. (debemos cambiar en el codigo C #define MAX 4294967295 // 2^32-1).

Tabla 1 .

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536				
131072				
262144				
524288				
1048576				
2097152				
4194304				
8388608				
16777216				
33554432				
67108864				

Listado 1 . Código C que suma dos vectores

```
/* SumaVectoresC.c
 Suma de dos vectores: v3 = v1 + v2

 Para compilar usar (-lrt: real time library):
        gcc  -O2 SumaVectores.c -o SumaVectores –lrt
 gcc  -O2 –S SumaVectores.c –lrt   //para generar el código ensamblador

 Para ejecutar use: SumaVectoresC longitud
*/

#include <stdlib.h> // biblioteca con funciones atoi(), malloc() y free()
#include <stdio.h> // biblioteca donde se encuentra la función printf()
#include <time.h>       // biblioteca donde se encuentra la función clock_gettime()

//#define PRINTF_ALL      // comentar para quitar el printf ...
                          // que imprime todos los componentes
//Sólo puede estar definida una de las tres constantes VECTOR_ (sólo uno de los ...
//tres defines siguientes puede estar descomentado):
//#define VECTOR_LOCAL    // descomentar para que los vectores sean variables ...
                          // locales (si se supera el tamaño de la pila se ...
                          // generará el error "Violación de Segmento")
//#define VECTOR_GLOBAL// descomentar para que los vectores sean variables ...
                          // globales (su longitud no estará limitada por el ...
                          // tamaño de la pila del programa)
#define VECTOR_DYNAMIC    // descomentar para que los vectores sean variables ...
                          // dinámicas (memoria reutilizable durante la ejecución)
#ifdef VECTOR_GLOBAL
#define MAX 33554432         //=2^25
double v1[MAX], v2[MAX], v3[MAX];
#endif

int main(int argc, char** argv){

   int i;
   struct timespec cgt1,cgt2; double ncgt; //para tiempo de ejecución

   //Leer argumento de entrada (nº de componentes del vector)
   if (argc<2){
      printf("Faltan nº componentes del vector\n");
      exit(-1);
   }

   unsigned int N = atoi(argv[1]); // Máximo N =2^32-1=4294967295 (sizeof(unsigned
int) = 4 B)
   #ifdef VECTOR_LOCAL
   double v1[N], v2[N], v3[N];   // Tamaño variable local en tiempo de ejecución ...
                                 // disponible en C a partir de actualización C99
   #endif
   #ifdef VECTOR_GLOBAL
   if (N>MAX) N=MAX;
   #endif
   #ifdef VECTOR_DYNAMIC
   double *v1, *v2, *v3;
   v1 = (double*) malloc(N*sizeof(double));// malloc necesita el tamaño en bytes
   v2 = (double*) malloc(N*sizeof(double)); //si no hay espacio suficiente malloc
```

```
devuelve NULL
   v3 = (double*) malloc(N*sizeof(double));
      if ( (v1==NULL) || (v2==NULL) || (v3==NULL) ){
      printf("Error en la reserva de espacio para los vectores\n");
      exit(-2);
   }
   #endif

   //Inicializar vectores
   for(i=0; i<N; i++){
      v1[i] = N*0.1+i*0.1; v2[i] = N*0.1-i*0.1; //los valores dependen de N
   }

   clock_gettime(CLOCK_REALTIME,&cgt1);
   //Calcular suma de vectores
   for(i=0; i<N; i++)
      v3[i] = v1[i] + v2[i];

   clock_gettime(CLOCK_REALTIME,&cgt2);
   ncgt=(double) (cgt2.tv_sec-cgt1.tv_sec)+
          (double) ((cgt2.tv_nsec-cgt1.tv_nsec)/(1.e+9));

   //Imprimir resultado de la suma y el tiempo de ejecución
   #ifdef PRINTF_ALL
   printf("Tiempo(seg.):%11.9f\t / Tamaño Vectores:%u\n",ncgt,N);
   for(i=0; i<N; i++)
      printf("/ V1[%d]+V2[%d]=V3[%d](%8.6f+%8.6f=%8.6f) /\n",
                i,i,i,v1[i],v2[i],v3[i]);

   #else
      printf("Tiempo(seg.):%11.9f\t / Tamaño Vectores:%u\t/ V1[0]+V2[0]=V3[0](%8.6f+
%8.6f=%8.6f) / /
                V1[%d]+V2[%d]=V3[%d](%8.6f+%8.6f=%8.6f) /\n",
                ncgt,N,v1[0],v2[0],v3[0],N-1,N-1,N-1,v1[N-1],v2[N-1],v3[N-1]);
   #endif

   #ifdef VECTOR_DYNAMIC
   free(v1); // libera el espacio reservado para v1
   free(v2); // libera el espacio reservado para v2
   free(v3); // libera el espacio reservado para v3
   #endif
   return 0;
}
```

Listado 2 . Código C++ que suma dos vectores

```
/* SumaVectoresCpp.cpp
 Suma de dos vectores: v3 = v1 + v2

 Para compilar usar (-lrt: real time library):
                g++  -O2 SumaVectoresCpp.cpp -o SumaVectoresCpp -lrt

 Para ejecutar use: SumaVectoresCpp longitud
*/

#include <cstdlib>   // biblioteca con atoi()
#include <iostream> // biblioteca donde se encuentra la función cout
using namespace std;
#include <time.h>       // biblioteca donde se encuentra la función clock_gettime()

//#define COUT_ALL // comentar para quitar el cout ...
                   // que imprime todos los componentes
//Sólo puede estar definida una de las tres constantes VECTOR_ (sólo uno de los ...
//tres defines siguientes puede estar descomentado):
//#define VECTOR_LOCAL    // descomentar para que los vectores sean variables ...
                          // locales (si se supera el tamaño de la pila se ...
                          // generará el error "Violación de Segmento")
//#define VECTOR_GLOBAL// descomentar para que los vectores sean variables ...
                          // globales (su longitud no estará limitada por el ...
                          // tamaño de la pila del programa)
#define VECTOR_DYNAMIC    // descomentar para que los vectores sean variables ...
                          // dinámicas (memoria reutilizable durante la ejecución)
#ifdef VECTOR_GLOBAL
#define MAX 33554432       //=2^25
double v1[MAX], v2[MAX], v3[MAX];
#endif

int main(int argc, char** argv){

   struct timespec cgt1,cgt2;   //para tiempo de ejecución

   //Leer argumento de entrada (nº de componentes del vector)
   if (argc<2){
      cout << "Faltan nº componentes del vector\n" << endl ;
      exit(-1);
   }

   unsigned int N = atoi(argv[1]);
   #ifdef VECTOR_LOCAL
   double v1[N], v2[N], v3[N];
   #endif
   #ifdef VECTOR_GLOBAL
   if (N>MAX) N=MAX;
   #endif
   #ifdef VECTOR_DYNAMIC
   double *v1, *v2, *v3;
   v1 = new double [N];    //si no hay espacio suficiente new genera una excepción
   v2 = new double [N];
   v3 = new double [N];
   #endif

   //Inicializar vectores
   for(int i=0; i<N; i++){
```

```
      v1[i] = N*0.1+i*0.1; v2[i] = N*0.1-i*0.1; //los valores dependen de N
   }
    clock_gettime(CLOCK_REALTIME,&cgt1);
   //Calcular suma de vectores
   for(int i=0; i<N; i++)
      v3[i] = v1[i] + v2[i];
    clock_gettime(CLOCK_REALTIME,&cgt2);
   double ncgt=(double) (cgt2.tv_sec-cgt1.tv_sec)+
         (double) ((cgt2.tv_nsec-cgt1.tv_nsec)/(1.e+9));

   //Imprimir resultado de la suma y el tiempo de ejecución
   #ifdef COUT_ALL
   cout << "Tiempo(seg.):" << ncgt << "\t/ Tamaño Vectores:" << N << endl;
   for(int i=0; i<N; i++)
         cout << "/ V1[" << i << "]+V2[" << i << "]=V3" << i << "](" << v1[i] << "+"
<< v2[i] << "="
         << v3[i] << ") /\t" << endl;
   cout <<"\n"<< endl;
   #else
         cout << "Tiempo(seg.):" << ncgt << "\t/ Tamaño Vectores:" << N << "\t/
V1[0]+V2[0]=V3[0]("
         << v1[0] << "+" << v2[0] << "=" << v3[0] << ") / / V1[" << N-1 << "]+V2["
<< N-1 << "]=V3["
         << N-1 << "](" << v1[N-1] << "+" << v2[N-1] << "=" << v3[N-1] << ")/\n" <<
endl;
   #endif

   #ifdef VECTOR_DYNAMIC
   delete [] v1; // libera el espacio reservado para v1
   delete [] v2; // libera el espacio reservado para v2
   delete [] v3; // libera el espacio reservado para v3
   #endif
   return 0;
}
```

Listado 3 . Script para la suma de vectores (SumaVectores.sh). Se supone en el script que el fichero a ejecutar se llama SumaVectorC y que se encuentra en el directorio en el que se ha ejecutado qsub.

```
#!/bin/bash
#Se asigna al trabajo el nombre SumaVectoresC_vlocales
#PBS -N SumaVectoresC_vlocales
#Se asigna al trabajo la cola ac
#PBS -q ac
#Se imprime información del trabajo usando variables de entorno de PBS
echo "Id. usuario del trabajo: $PBS_O_LOGNAME"
echo "Id. del trabajo: $PBS_JOBID"
echo "Nombre  del trabajo especificado por usuario: $PBS_JOBNAME"
echo "Nodo que ejecuta qsub: $PBS_O_HOST"
echo "Directorio en el que se ha ejecutado qsub: $PBS_O_WORKDIR"
echo "Cola: $PBS_QUEUE"
echo "Nodos asignados al trabajo:"
cat $PBS_NODEFILE
#Se ejecuta SumaVectorC, que está en el directorio en el que se ha ejecutado qsub,
#para N potencia de 2 desde 2^16 a 2^26
for ((N=65536;N<67108865;N=N*2))
```

```
do
    $PBS_O_WORKDIR/SumaVectoresC $N
done
```