

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): Jesús Manuel García Palma

Grupo de prácticas: D2

Fecha de entrega: 08/03/18

Fecha evaluación en clase: 09/03/08

[RECORDATORIO, quitar todo este texto en rojo del cuaderno definitivo–

1. COMENTARIOS

- 1) Este cuaderno de prácticas se utilizará para asignarle una puntuación durante la evaluación continua de prácticas y también lo utilizará como material de estudio y repaso para preparar el examen de prácticas escrito. Luego redáctelo con cuidado, y sea ordenado y claro.
- 2) No use máquinas virtuales. Se piden obtener resultados en atcgrid y en su PC (PC del aula o PC personal).
- 3) Debe modificar el prompt en los computadores que utilice en prácticas para que aparezca su nombre y apellidos, su usuario (\u), el computador (\h), el directorio de trabajo del bloque práctico (\w), la fecha (\D) completa (%F) y el día (%A) . Para modificar el prompt utilice lo siguiente (si es necesario, use export delante):

donde NombreApellidos es su nombre seguido de sus apellidos, por ejemplo: Juan Ortuño Vilariño

2. NORMAS SOBRE EL USO DE LA PLANTILLA

- 1) Usar **interlineado SENCILLO**.
- 2) Respetar los tipos de letra y tamaños indicados:
 - Calibri-11 o Liberation Serif-11 para el texto
 - **Courier New-10 o Liberation Mono-10 para nombres de fichero, comandos, variables de entorno, etc., cuando se usan en el texto.**
- 3) Insertar las capturas de pantalla donde se pidan y donde se considere oportuno. En particular, los listados de código se deben insertar como capturas de pantalla. En todas las capturas de pantalla, incluidas las de los listados de código, **debe aparecer el directorio y usuario**. El tamaño de letra en las capturas debe ser similar al tamaño que se está usando en el texto.
Recuerde que debe adjuntar al zip de entrega, el pdf de este fichero, todos los ficheros con código fuente implementados/utilizados y el resto de ficheros que haya implementado/utilizado (scripts, hojas de cálculo, etc.)]

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

CAPTURAS:

```

Jesus@jesus-Lenovo-G585: ~/Escritorio/AC/Practicas_grupo reducido_/2. Seminarios/Seminario0/scriptHelloOMP
Archivo Editar Ver Buscar Terminal Ayuda
[Jesus Manuel García Palma jesus@jesus-Lenovo-G585:~/Escritorio/AC/Practicas_grupo reducido_/2. Seminarios/Seminario0/EjerciciosCuaderno1/1]
2018-02-26 Lunes
$ls-cpu
Arquitectura: x86_64
modo(s) de operación de las CPUs: 32-bit, 64-bit
Orden de bytes: Little Endian
CPU(s): 2
On-line CPU(s) list: 0,1
Hilo(s) de procesamiento por núcleo: 1
Núcleo(s) por «socket»: 2
Socket(s): 1
Modo(s) NUMA: 1
ID de fabricante: AuthenticAMD
Familia de CPU: 20
Modelo: 2
Model name: AMD E1-1200 APU with Radeon(tm) HD Graphics
Revisión: 0
CPU MHz: 1166.000
CPU max MHz: 1400.0000
CPU min MHz: 777.0000
BogoMIPS: 2794.98
Virtualización: AMD-V
Caché L1d: 32K
Caché L1i: 32K
Caché L2: 512K
NUMA node0 CPU(s): 0,1
Flags: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht sysc
all nx mmxext fxsr_opt pdpe1gb rdtscp lm constant_tsc rep_good noopl nonstop_tsc cpuid extd_apicid aperfmperf pni monitor ssse3 cx16 popcnt lah
f_lm cmp_legacy svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch ibs skinit wdt hw_pstate retpoline retpoline_amd rsb_ctxsw vmcall
arat npt lbrv svm_lock nrip_save pausefilter
[Jesus Manuel García Palma jesus@jesus-Lenovo-G585:~/Escritorio/AC/Practicas_grupo reducido_/2. Seminarios/Seminario0/EjerciciosCuaderno1/1]
$

```

```

Jesus@jesus-Lenovo-G585: ~/Escritorio/AC/Practicas_grupo reducido_/2. Seminarios/Seminario0/scriptHelloOMP
Archivo Editar Ver Buscar Terminal Ayuda
[Jesus Manuel García Palma jesus@jesus-Lenovo-G585:~/Escritorio/AC/Practicas_grupo reducido_/2. Seminarios/Seminario0/EjerciciosCuaderno1/1]
2018-02-26 Lunes
$cat STDIN.o61558
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: 1598.814
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 s
yscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good noopl xtopology nonstop_tsc cpuid aperfmperf pni dtes64 monitor ds_cpl
vmx smx est tm2 ssse3 cx16 xtpr pdcm pcid dca sse4_1 sse4_2 popcnt lahf_lm epb pti retpoline tpr_shadow vnmi flexpriority ept vpid dtherm ida
arat

```

1- Mi PC

2- ATCGRID

1. Conteste a las siguientes preguntas:

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

RESPUESTA: Mi PC → 2 cores físicos / 2 cores lógicos

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

RESPUESTA: nodo ATCGRID → 12 cores físicos / 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++:

$v3 = v1 + v2$; $v3(i) = v1(i) + v2(i)$, $i=0, \dots, N-1$

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: `ncgt` contiene la diferencia de la variable `cgt2` y `cgt1` cuyos tiempos son los calculados antes y después del bucle. Respecto a `clock_gettime()`, devuelve un valor en un instante de tiempo, a partir de una estructura que guarda los segundos y nanosegundos de ese instante, la cual podemos obtener de la cabecera `time.h`

- 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++
Tamaño vector	<code>malloc(N*sizeof(double))</code>	<code>v1 = new double [N];</code>
Liberar espacio	<code>free(v1);</code>	<code>delete [] v1;</code>
Salida de datos	<code>Printf(...)</code>	<code>Cout<<"..."</code>
Espacio de nombres	No tiene	Sí tiene
Cabeceras	<code>.h</code>	Normal

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: Una vez compilado se lanza la orden `echo './SumaVectoresC' | qsub -q ac`, dando como resultado la siguiente salida:

```
sftp> put Local/SumaVectoresC
Uploading Local/SumaVectoresC to /home/D2estudiante5/Practica0/ej2/Local/SumaVectoresC
Local/SumaVectoresC 100% 8536 231.1KB/s 00:00
sftp> put Local/SumaVectoresCpp
Uploading Local/SumaVectoresCpp to /home/D2estudiante5/Practica0/ej2/Local/SumaVectoresCpp
Local/SumaVectoresCpp 100% 13KB 184.4KB/s 00:00
sftp> cd ..
sftp> cd Global/
sftp> put
Dinamic/ Global/ Local/
sftp> put Global/
SumaVectoresC SumaVectoresC.c
SumaVectoresCpp SumaVectoresCpp.cpp

sftp> put Global/SumaVectoresC
Uploading Global/SumaVectoresC to /home/D2estudiante5/Practica0/ej2/Global/SumaVectoresC
Global/SumaVectoresC 100% 8624 225.5KB/s 00:00
sftp> put Global/SumaVectoresCpp
Uploading Global/SumaVectoresCpp to /home/D2estudiante5/Practica0/ej2/Global/SumaVectoresCpp
Global/SumaVectoresCp 100% 13KB 189.9KB/s 00:00
sftp>
```

```
[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej2/Local] 2018-03-04
domingo
$echo 'Practica0/ej2/Local/SumaVectoresC 50' | qsub -q ac
64221.atcgrid
[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej2/Local] 2018-03-04
domingo
$echo 'Practica0/ej2/Local/SumaVectoresCpp 50' | qsub -q ac
64222.atcgrid
[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej2/Local] 2018-03-04
domingo
$qstat
Job ID Name User Time Use S Queue
-----
64222.atcgrid STDIN D2estudiante5 00:00:00 C ac
Sublime Text
[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej2/Local] 2018-03-04
domingo
$
```

```
sftp> get Dynamic/STDIN.o*
Fetching /home/D2estudiante5/Practica0/ej2/Dynamic/STDIN.o64218 to STDIN.o64218
/home/D2estudiante5/Pra 100% 149 2.1KB/s 00:00
Fetching /home/D2estudiante5/Practica0/ej2/Dynamic/STDIN.o64220 to STDIN.o64220
/home/D2estudiante5/Pra 100% 1846 26.5KB/s 00:00
sftp> get Local/S
STDIN.e64221 STDIN.e64222 STDIN.o64221
STDIN.o64222 SumaVectoresC SumaVectoresCpp
sftp> get Local/ST
STDIN.e64221 STDIN.e64222 STDIN.o64221 STDIN.o64222
sftp> get Local/STDIN.o*
Fetching /home/D2estudiante5/Practica0/ej2/Local/STDIN.o64221 to STDIN.o64221
/home/D2estudiante5/Pra 100% 149 2.2KB/s 00:00
Fetching /home/D2estudiante5/Practica0/ej2/Local/STDIN.o64222 to STDIN.o64222
/home/D2estudiante5/Pra 100% 1846 26.4KB/s 00:00
sftp> get Global/ST
STDIN.e64223 STDIN.e64224 STDIN.o64223 STDIN.o64224
sftp> get Global/STDIN.o*
Fetching /home/D2estudiante5/Practica0/ej2/Global/STDIN.o64223 to STDIN.o64223
/home/D2estudiante5/Pra 100% 149 2.2KB/s 00:00
Fetching /home/D2estudiante5/Practica0/ej2/Global/STDIN.o64224 to STDIN.o64224
/home/D2estudiante5/Pra 100% 1846 27.1KB/s 00:00
sftp>
```

```
[jesusManuelGarciaPalma jesus@jesus-Lenovo-G585:~/Escritorio/AC/Practicas_grupo reducido_/2. Seminarios/Seminario0/EjerciciosCuaderno1/2/Loca
l] 2018-03-04 domingo
$ls
STDIN.064221 STDIN.064222 SumaVectoresC SumaVectoresC.c SumaVectoresCpp SumaVectoresCpp.cpp
[jesusManuelGarciaPalma jesus@jesus-Lenovo-G585:~/Escritorio/AC/Practicas_grupo reducido_/2. Seminarios/Seminario0/EjerciciosCuaderno1/2/Loca
l] 2018-03-04 domingo
$cat STDIN.064221
Tiempo(seg.):0.000000255 / Tamaño Vectores:50 / V1[0]+V2[0]=V3[0](5.000000+5.000000=10.000000) V1[49]+V2[49]=V3[49](9.900000+0.10000
0=10.000000) /
[jesusManuelGarciaPalma jesus@jesus-Lenovo-G585:~/Escritorio/AC/Practicas_grupo reducido_/2. Seminarios/Seminario0/EjerciciosCuaderno1/2/Loca
l] 2018-03-04 domingo
$
```

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: Después de ejecutar la orden `qsub SumaVectores.sh -q ac` se obtiene la siguiente salida, donde se aprecia que se realiza la suma 3 veces y para debido a un error de violación de segmento. Esto último se puede comprobar mirando el `.e` obtenido. Respecto a mi PC, no imprime ninguna variable y es necesario hacer CTRL+C para poder desbloquear el terminal.

[illegible]

```

Tiempo(seg.):0.000424675 / 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.):0.000843769 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) V1[131071]+V2[1310
71]=V3[131071](26214.300000+0.100000=26214.400000) /
Tiempo(seg.):0.001695278 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) V1[262143]+V2[2621
43]=V3[262143](52428.700000+0.100000=52428.800000) /
[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej4] 2018-03-05 lunes
$

```

```

[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej4] 2018-03-05 lunes
$cat SumaVectoresC_vlocales.e65095
/var/lib/torque/mom_priv/jobs/65095.atcgrid.SC: line 20: 12987 Segmentation fault (core dumped) $PBS_O_WORKDIR/SumaVectores $N
/var/lib/torque/mom_priv/jobs/65095.atcgrid.SC: line 20: 12990 Segmentation fault (core dumped) $PBS_O_WORKDIR/SumaVectores $N
/var/lib/torque/mom_priv/jobs/65095.atcgrid.SC: line 20: 12993 Segmentation fault (core dumped) $PBS_O_WORKDIR/SumaVectores $N
/var/lib/torque/mom_priv/jobs/65095.atcgrid.SC: line 20: 12996 Segmentation fault (core dumped) $PBS_O_WORKDIR/SumaVectores $N
/var/lib/torque/mom_priv/jobs/65095.atcgrid.SC: line 20: 12999 Segmentation fault (core dumped) $PBS_O_WORKDIR/SumaVectores $N
/var/lib/torque/mom_priv/jobs/65095.atcgrid.SC: line 20: 13002 Segmentation fault (core dumped) $PBS_O_WORKDIR/SumaVectores $N
/var/lib/torque/mom_priv/jobs/65095.atcgrid.SC: line 20: 13005 Segmentation fault (core dumped) $PBS_O_WORKDIR/SumaVectores $N
/var/lib/torque/mom_priv/jobs/65095.atcgrid.SC: line 20: 13008 Segmentation fault (core dumped) $PBS_O_WORKDIR/SumaVectores $N
[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej4] 2018-03-05 lunes
$

```

Resultado en Mi pc:

```

[JesusManuelGarciaPalma jesus@jesus-Lenovo-G585:~/Escritorio/AC/Practicas_grupo
reducido_/2. Seminarios/Seminario0/EjerciciosCuaderno1/4] 2018-03-05 lunes
$./SumaVectores.sh
Id. usuario del trabajo:
Id. del trabajo:
Nombre del trabajo especificado por usuario:
Nodo que ejecuta qsub:
Directorio en el que se ha ejecutado qsub:
Cola:
Nodos asignados al trabajo:
^C
[JesusManuelGarciaPalma jesus@jesus-Lenovo-G585:~/Escritorio/AC/Practicas_grupo
reducido_/2. Seminarios/Seminario0/EjerciciosCuaderno1/4] 2018-03-05 lunes
$

```

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: Se realiza el comando qsub echo './SumaVectoresC.sh para el caso dinámico y global, dando como resultado las salidas que se aprecian en las imágenes. Esta vez no presentan ningún error en cuanto a la memoria, pues ya que el uso de memoria dinámica permite que no haya ningún error respecto al espacio. Los .e no contienen tampoco nada en este caso.

```

[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej5/Dynamic] 2
018-03-06 martes
$echo './SumaVectoresC 50' | qsub -q ac
65292.atcgrid
[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej5/Dynamic] 2
018-03-06 martes
$qstat
Job ID          Name          User          Time Use S Qu
eue
-----
65292.atcgrid   STDIN         D2estudiante5 00:00:00 C a
c
[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej5/Dynamic] 2
018-03-06 martes
$

```



```
[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej5/Dynamic] 2
018-03-06 martes
$cat STDIN.o65292
Tiempo(seg.):0.000000272 / Tamaño Vectores:50 / V1[0]+V2[0]=V3
[0](5.000000+5.000000=10.000000) V1[49]+V2[49]=V3[49](9.900000+0.100000=
10.000000) /
[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej5/Dynamic] 2
018-03-06 martes
$cat STDIN.e65292
[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej5/Dynamic] 2
018-03-06 martes
```

```
[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej5/Global] 20
18-03-06 martes
$echo './SumaVectoresC 50' | qsub -q ac
65293.atcgrid
[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej5/Global] 20
18-03-06 martes
$qstat
Job ID Name User Time Use S Qu
eue
-----
65293.atcgrid STDIN D2estudiante5 00:00:00 C a
c
[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej5/Global] 20
18-03-06 martes
$
```

```
[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej5/Global] 20
18-03-06 martes
$ls
STDIN.e65293 SumaVectoresC SumaVectores.sh
STDIN.o65293 SumaVectoresCpp
[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej5/Global] 20
18-03-06 martes
$cat STDIN.o65293
Tiempo(seg.):0.000000260 / Tamaño Vectores:50 / V1[0]+V2[0]=V3
[0](5.000000+5.000000=10.000000) V1[49]+V2[49]=V3[49](9.900000+0.100000=
10.000000) /
[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej5/Global] 20
18-03-06 martes
$cat STDIN.e65293
[JesusManuelGarciaPalma D2estudiante5@atcgrid:~/Practica0/ej5/Global] 20
18-03-06 martes
$
```

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:**MI PC**

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0.001051895	0.001254557	0.001163476
131072	1048576	0.002391433	0.002600275	0.002383693
262144	2097152	0.004952756	0.004710459	0.005384720
524288	4194304		0.011048798	0.008822877
1048576	8388608		0.037753236	0.018636698
2097152	16777216		0.040989826	0.033456460
4194304	33554432		0.074914294	0.065012121
8388608	67108864		0.137712540	0.129059583
16777216	134217728		0.292529855	0.275947051
33554432	268435456		0.534600883	0.522095246
67108864	536870912		0.534600883	1.038733842

ATCGRID

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0.000273670	0.000427001	0.000425493
131072	1048576	0.000895667	0.000912634	0.000840005
262144	2097152	0.001710022	0.001191749	0.001508164
524288	4194304		0.002573360	0.003485943
1048576	8388608		0.005849741	0.006447685
2097152	16777216		0.012524890	0.011932643
4194304	33554432		0.025169624	0.024016956
8388608	67108864		0.044866345	0.048298570
16777216	134217728		0.089924658	0.090421157
33554432	268435456		0.189724176	0.185446534
67108864	536870912		0.377675080	0.380676303

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: Se ha usado el valor 4294967295 ($2^{32}-1$) , dando lugar a un error de desbordamiento ya que dicho número es el máximo numero que es posible representar en un entero de 32 bits

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

```