

BP0.pdf



postdata9



Arquitectura de Computadores



2º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación
Universidad de Granada



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.





**KEEP
CALM
AND
ESTUDIA
UN POQUITO**

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: A2

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:

```
A2estudiante24@atcgrid:~  
CPU(s) del nodo NUMA 0: 0  
Indicadores: fpu vme de pse tsc msr pae mce cx8 apic sep  
mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 syscall nx mmxext fxsr_op  
t lm 3dnowext 3dnow rep_good nopl cpuid extd_apicid dni lahf_lm 3dnowprefetch re  
tpoline rsb_ctxsw vmmcall  
[A2estudiante24@atcgrid hello]$ cd ../  
[A2estudiante24@atcgrid ~]$ lscpu  
Arquitectura: x86_64  
modo(s) de operación de las CPUs: 32-bit, 64-bit  
Orden de los bytes: Little Endian  
CPU(s): 1  
Lista de la(s) CPU(s) en línea: 0  
Hilo(s) de procesamiento por núcleo: 1  
Núcleo(s) por «socket»: 1  
«Socket(s)»: 1  
Modo(s) NUMA: 1  
ID de fabricante: AuthenticAMD  
Familia de CPU: 15  
Modelo: 47  
Nombre del modelo: AMD Athlon(tm) 64 Processor 3000+  
Revisión: 2  
CPU MHz: 1802.252  
BogoMIPS: 3604.50  
Caché L1d: 64K  
Caché L1i: 64K  
Caché L2: 512K  
CPU(s) del nodo NUMA 0: 0  
Indicadores: fpu vme de pse tsc msr pae mce cx8 apic sep  
mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 syscall nx mmxext fxsr_op  
t lm 3dnowext 3dnow rep_good nopl cpuid extd_apicid dni lahf_lm 3dnowprefetch re  
tpoline rsb_ctxsw vmmcall  
[A2estudiante24@atcgrid ~]$
```

```
LXTerminal  
Archivo Editar Pestañas Ayuda  
[PaulaSantosOrtega postdata9@ei141082:~/Escritorio] 2018-02-20 martes  
$  
[PaulaSantosOrtega postdata9@ei141082:~/Escritorio] 2018-02-20 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: 1599.951  
CPU max MHz: 3400.0000  
CPU min MHz: 1600.0000  
BogoMIPS: 6220.41  
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 xtprocology nonstop_tsc aperf  
mpref eagerfpu dni pclmuldq 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_lm  
epb tpr_shadow vnmi flexpriority ept vpid xsaveopt dtherm ida arat pln pts  
[PaulaSantosOrtega postdata9@ei141082:~/Escritorio] 2018-02-20 martes  
$
```

Conteste a las siguientes preguntas:

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

RESPUESTA: El atcgrid de prácticas tiene 1 core lógico (podemos verlo usando la orden lscpu, en la columna de CPU) y 1 core físico (tenemos 1 socket y 1 núcleo por socket, y cada núcleo tiene 1 hilo de procesamiento).

Mi PC tiene 4 cores lógico y 4 cores físicos.

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

RESPUESTA: Un nodo de atcgrid tiene 1 core lógico (lo vemos en la fila de CPU(s)) y tiene 1 core físico.

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++:

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

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:

La variable `ncgt` contiene el tiempo que ha tardado el programa en calcular la suma de vectores en segundos más el tiempo en nanosegundos, dividido entre $1.e+9$.

La función `int clock_gettime(clockid_t clk_id, struct timespec *tp);` devuelve el tiempo especificado por el `clk_id`, que en nuestro código es `clock_realtime`, que representa el tiempo en segundos y nanosegundos transcurridos desde el 00:00:00 01-01-1975 hasta el momento actual.

La función `clock_gettime` devuelve la información en la estructura `timespec`:

```
struct timespec {  
    time_t tv_sec;      // segundos  
    long   tv_nsec;     // nanosegundos  
};
```

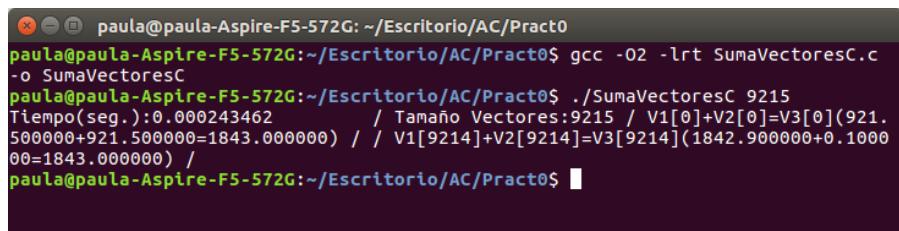
- b. Escribir en el cuaderno de prácticas las diferencias que hay entre el código fuente C y el código fuente C++ para la suma de vectores.

RESPUESTA:

Descripción diferencia	En C	En C++
Distinta forma de imprimir.	Usa la función printf()	Utiliza el objeto “cout” de la clase ostream
Diferentes formas de asignar memoria dinámica.	Malloc()	New
Diferentes formas de liberar memoria dinámica.	Free()	Delete[]
El uso del namespace std		

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:



```
paula@paula-Aspire-F5-572G:~/Escritorio/AC/Pract0
paula@paula-Aspire-F5-572G:~/Escritorio/AC/Pract0$ gcc -O2 -lrt SumaVectoresC.c
-o SumaVectoresC
paula@paula-Aspire-F5-572G:~/Escritorio/AC/Pract0$ ./SumaVectoresC 9215
Tiempo(seg.):0.000243462 / Tamaño Vectores:9215 / V1[0]+V2[0]=V3[0](921.
500000+921.500000=1843.000000) / / V1[9214]+V2[9214]=V3[9214](1842.900000+0.1000
00=1843.000000) /
paula@paula-Aspire-F5-572G:~/Escritorio/AC/Pract0$
```

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)

1.

RESPUESTA:

Para PC:

```
paula@paula-Aspire-F5-572G:~/Escritorio/AC/Pract0/VectoresLoc$ ./SumaVectoresLoc.sh
Tiempo(seg.):0.001933735          / Tamaño Vectores:65536          / V1[0]+V2[0]=V3[0](6553.600000+6553.60
0000=13107.200000) / V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
Tiempo(seg.):0.003086795          / 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.):0.004141464          / 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) /
./SumaVectoresLoc.sh: línea 23: 17759 Violación de segmento ('core' generado) ./SumaVectoresLoc $N
./SumaVectoresLoc.sh: línea 23: 17761 Violación de segmento ('core' generado) ./SumaVectoresLoc $N
./SumaVectoresLoc.sh: línea 23: 17763 Violación de segmento ('core' generado) ./SumaVectoresLoc $N
./SumaVectoresLoc.sh: línea 23: 17765 Violación de segmento ('core' generado) ./SumaVectoresLoc $N
./SumaVectoresLoc.sh: línea 23: 17767 Violación de segmento ('core' generado) ./SumaVectoresLoc $N
./SumaVectoresLoc.sh: línea 23: 17769 Violación de segmento ('core' generado) ./SumaVectoresLoc $N
./SumaVectoresLoc.sh: línea 23: 17771 Violación de segmento ('core' generado) ./SumaVectoresLoc $N
./SumaVectoresLoc.sh: línea 23: 17773 Violación de segmento ('core' generado) ./SumaVectoresLoc $N
paula@paula-Aspire-F5-572G:~/Escritorio/AC/Pract0/VectoresLoc$
```

En el PC, sale error para tamaño de vector mayor que 262144, de *Violación de segmento*. Esto se debe a que el programa en ejecución ha intentado acceder a una zona de la memoria a la que no tiene permiso, es decir, como estamos trabajando con variables locales, tenemos unos límites dentro de la sección que el sistema operativo nos ha asignado, y como vemos en la captura, a partir de 262144, las variables sobresalen del límite.

Para atcgrid:

```
[A2estudiante24@atcgrid:~]$ cat SumaVectoresC_vlocales.o63627
[Tiempo(seg.):0.000420133 / 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.):0.000839202 / 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.):0.001392156 / 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) /
[A2estudiante24@atcgrid ~]$ cat SumaVectoresC_vlocales.e63627
/var/lib/torque/nom_priv/jobs/63627.atcgrid.SC: line 23: 29811 Segmentation fault      (core dumped) ./SumaVectoresLoc $N
/var/lib/torque/nom_priv/jobs/63627.atcgrid.SC: line 23: 29814 Segmentation fault      (core dumped) ./SumaVectoresLoc $N
/var/lib/torque/nom_priv/jobs/63627.atcgrid.SC: line 23: 29817 Segmentation fault      (core dumped) ./SumaVectoresLoc $N
/var/lib/torque/nom_priv/jobs/63627.atcgrid.SC: line 23: 29824 Segmentation fault      (core dumped) ./SumaVectoresLoc $N
/var/lib/torque/nom_priv/jobs/63627.atcgrid.SC: line 23: 29827 Segmentation fault      (core dumped) ./SumaVectoresLoc $N
/var/lib/torque/nom_priv/jobs/63627.atcgrid.SC: line 23: 29830 Segmentation fault      (core dumped) ./SumaVectoresLoc $N
/var/lib/torque/nom_priv/jobs/63627.atcgrid.SC: line 23: 29834 Segmentation fault      (core dumped) ./SumaVectoresLoc $N
/var/lib/torque/nom_priv/jobs/63627.atcgrid.SC: line 23: 29837 Segmentation fault      (core dumped) ./SumaVectoresLoc $N
[A2estudiante24@atcgrid ~]$
```

Podemos observar que ocurre el mismo error que en el PC, *Segmentation fault*.



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.

Available on the
App Store

GET IT ON
Google Play



18

Ver mis op

Cuaderno de prácticas de Arquitectura de Computadores, Grado en Ingeniería Informática

Continúa d

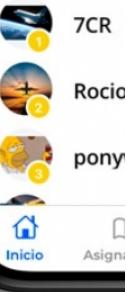


405416_arts_esce_ues2016juniy.pdf

2.

RESPUESTA:

Top de tu gr



EN ATCGRID:

Para variables globales:

```
A2estudiante24@atcgrid:~$ cat SumaVectoresC_vglobales.o63547
[A2estudiante24@atcgrid ~]$ cat SumaVectoresC_vglobales.o63547
Tiempo(seg.):0.00051029          / 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.):0.000792256         / Tamaño Vectores:131072     / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / / V1[131071]+V2[131
071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tiempo(seg.):0.01385945          / Tamaño Vectores:262144     / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]+V2[26
143]=V3[262143](52428.700000+0.100000=52428.800000) /
Tiempo(seg.):0.003096655         / Tamaño Vectores:524288     / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / / V1[524287]+V2[52
4287]=V3[524287](104857.500000+0.100000=104857.600000) /
Tiempo(seg.):0.006259664          / 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.):0.012137120          / 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.):0.025346216          / 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.):0.047970985          / Tamaño Vectores:8388608   / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / / V1[8388607]+V
2[8388607]=V3[8388607](1677721.590000+0.100000=1677721.600000) /
Tiempo(seg.):0.095998369          / 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.):0.18855907           / 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.):0.372082705          / Tamaño Vectores:67108864   / V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) / / V1[6710886
3]+V2[67108863]=V3[67108863](13421772.700000+0.100000=13421772.800000) /
[A2estudiante24@atcgrid ~]$
```

Para variables dinámicas:

```
A2estudiante24@atcgrid:~$ cat SumaVectoresC_vdinamicas.o63584
[A2estudiante24@atcgrid ~]$ cat SumaVectoresC_vdinamicas.o63584
Tiempo(seg.):0.000415983          / 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.):0.000841472          / Tamaño Vectores:131072     / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / / V1[131071]+V2[131
071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tiempo(seg.):0.001685759          / Tamaño Vectores:262144     / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]+V2[26
143]=V3[262143](52428.700000+0.100000=52428.800000) /
Tiempo(seg.):0.002581688          / Tamaño Vectores:524288     / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / / V1[524287]+V2[52
4287]=V3[524287](104857.500000+0.100000=104857.600000) /
Tiempo(seg.):0.005996094          / 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.):0.012017986          / 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.):0.023357061          / 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.):0.046919889          / Tamaño Vectores:8388608   / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / / V1[8388607]+V
2[8388607]=V3[8388607](1677721.590000+0.100000=1677721.600000) /
Tiempo(seg.):0.094643716          / 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.):0.190939352          / 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.):0.371185766          / Tamaño Vectores:67108864   / V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) / / V1[6710886
3]+V2[67108863]=V3[67108863](13421772.700000+0.100000=13421772.800000) /
[A2estudiante24@atcgrid ~]$
```

EN PC:

Para variables globales:

```
paula@paula-Aspire-F5-572G:~/Escritorio/AC/Pract0/VectoresGlob$ ./SumaVectoresGlob.sh
Tiempo(seg.):0.000447999 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000)
00) / / V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
Tiempo(seg.):0.001095447 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000)
0000) / / V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tiempo(seg.):0.002506892 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000)
00000) / / V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
Tiempo(seg.):0.004028354 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.6
00000) / / V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
Tiempo(seg.):0.004485927 / 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.):0.009639123 / 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.):0.018935331 / 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.):0.037184718 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=167772
1.600000) / / V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000) /
Tiempo(seg.):0.073899107 / Tamaño Vectores:16777216 / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355
443.200000) / / V1[16777215]+V2[16777215]=V3[16777215](3355443.100000+0.100000=3355443.200000) /
Tiempo(seg.):0.146159651 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710
886.400000) / / V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
Tiempo(seg.):0.145503630 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710
886.400000) / / V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
paula@paula-Aspire-F5-572G:~/Escritorio/AC/Pract0/VectoresGlob$
```

Para variables dinámicas:

```
paula@paula-Aspire-F5-572G:~/Escritorio/AC/Pract0/VectoresDin$ ./SumaVectoresDin.sh
paula@paula-Aspire-F5-572G:~/Escritorio/AC/Pract0/VectoresDin$ ./SumaVectoresDin.sh
Tiempo(seg.):0.000309518 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.2000
00) / / V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
Tiempo(seg.):0.000728598 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.40
0000) / / V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tiempo(seg.):0.001667216 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.80
0000) / / V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
Tiempo(seg.):0.003601452 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.6
00000) / / V1[524287]+V2[524287]=V3[104857](104857.500000+0.100000=104857.600000) /
Tiempo(seg.):0.004775221 / 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.):0.009613387 / 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.):0.018927052 / 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.):0.037027785 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=167772
1.600000) / / V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000) /
Tiempo(seg.):0.073517534 / Tamaño Vectores:16777216 / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355
443.200000) / / V1[16777215]+V2[16777215]=V3[16777215](3355443.100000+0.100000=3355443.200000) /
Tiempo(seg.):0.145398009 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710
886.400000) / / V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
Tiempo(seg.):0.321520569 / Tamaño Vectores:67108864 / V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=1342
1772.800000) / / V1[67108863]+V2[67108863]=V3[67108863](13421772.700000+0.100000=13421772.800000) /
paula@paula-Aspire-F5-572G:~/Escritorio/AC/Pract0/VectoresDin$
```

Para variables dinámicas, no se obtiene error tanto en PC como en atcgrid. Sin embargo, para variables globales vemos que en PC no calcula para tamaño de vector 67108864.

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:

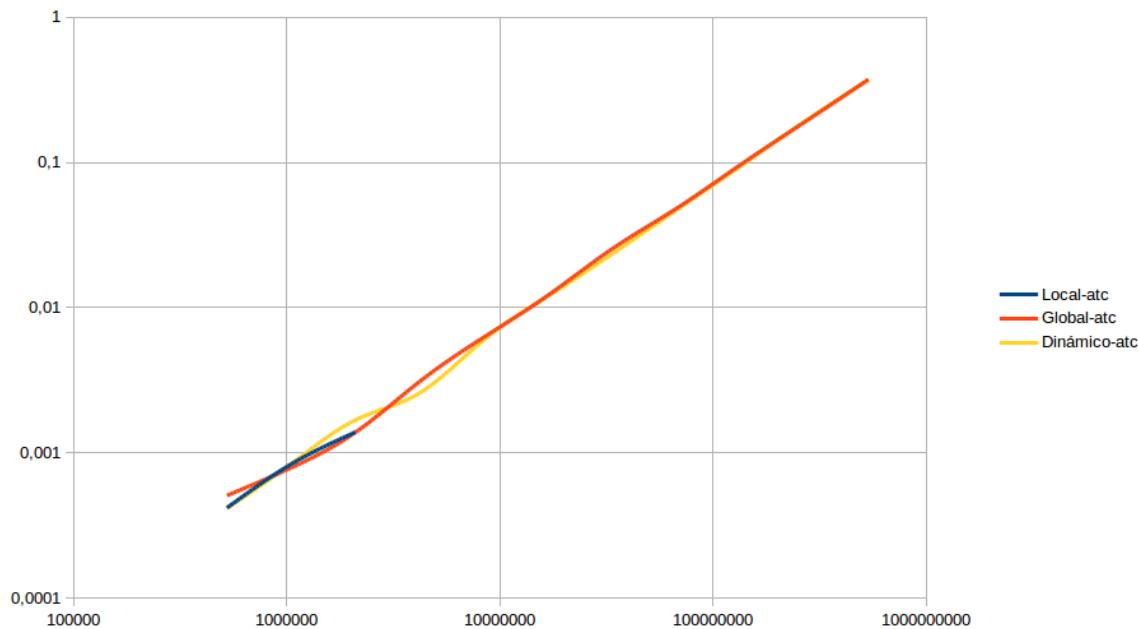
Tabla 1 . Para atcgrid

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0,000420133	0,000510229	0,000415983
131072	1048576	0,000839202	0,000792256	0,000841472
262144	2097152	0,001392156	0,001385945	0,001685759
524288	4194304		0,003096655	0,002581680
1048576	8388608		0,006250664	0,005996004
2097152	16777216		0,012137120	0,012017980
4194304	33554432		0,025346216	0,023357001
8388608	67108864		0,047970985	0,046919809
16777216	134217728		0,095998309	0,094043710
33554432	268435456		0,188855907	0,190939352
67108864	536870912		0,372082705	0,371185706

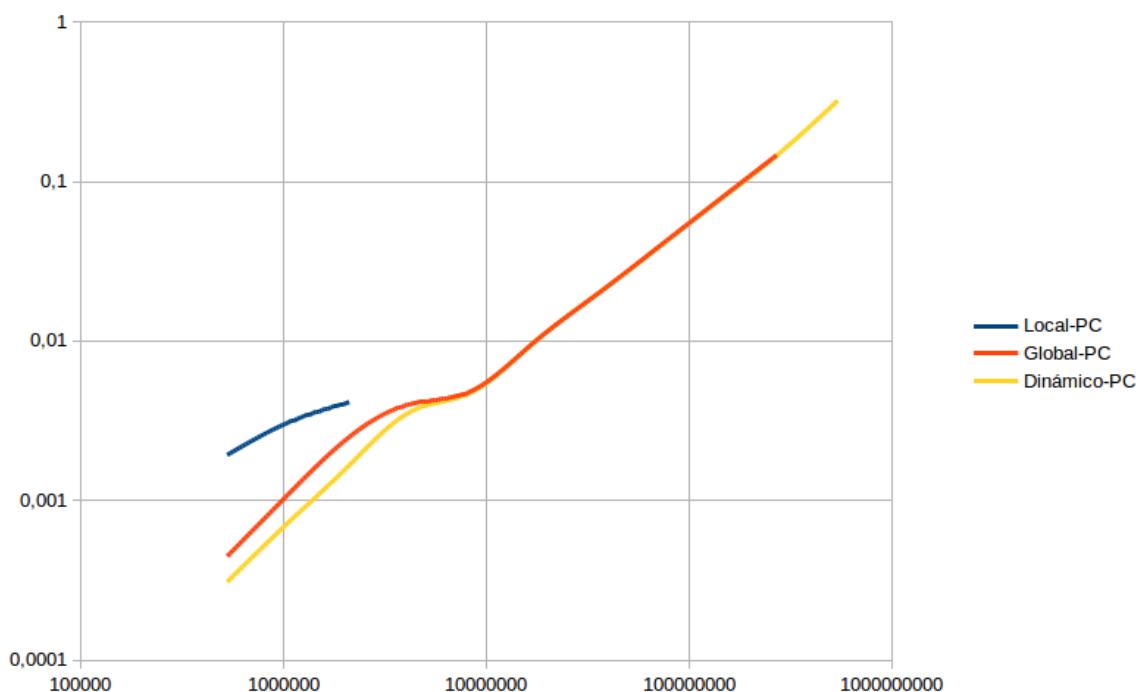
Tabla 2 . Para mi PC

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0,001933735	0,000447999	0,000309518
131072	1048576	0,003086795	0,001095447	0,000728598
262144	2097152	0,004141404	0,002506892	0,001667216
524288	4194304		0,004028354	0,003601452
1048576	8388608		0,004859927	0,004775221
2097152	16777216		0,009639123	0,009613387
4194304	33554432		0,018935331	0,018927052
8388608	67108864		0,037184718	0,037027785
16777216	134217728		0,073899107	0,073517534
33554432	268435456		0,146159651	0,145398009
67108864	536870912			0,321520569

Gráfica en función del tamaño y tiempo de ejecución en atcgrid



Gráfica en función del tamaño y tiempo de ejecución en PC



Sí hay diferencia en los tiempos de ejecución.



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

Available on the App Store GET IT ON Google Play

Cuaderno de prácticas de Arquitectura de Computadores, Grado en Ingeniería Informática

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 $2^{32}-1$.

RESPUESTA:

```
paula@paula-Aspire-F5-572G:~/Escritorio/AC/Pract0/VectoresGlob$ gcc -O2 -lrt SumaVectoresGlob.c -o SumaVectoresGlob  
/tmp/ccNUX0hm.o: En la función _main':  
SumaVectoresGlob.c:(.text.startup+0x9): reubicación truncada para ajustar: R_X86_64_325 contra el símbolo `v2' definido en la sección COMMON en /tmp/ccNUX0hm.o  
SumaVectoresGlob.c:(.text.startup+0xc0): reubicación truncada para ajustar: R_X86_64_325 contra el símbolo `v2' definido en la sección COMMON en /tmp/ccNUX0hm.o  
SumaVectoresGlob.c:(.text.startup+0x8): reubicación truncada para ajustar: R_X86_64_325 contra el símbolo `v3' definido en la sección COMMON en /tmp/ccNUX0hm.o  
SumaVectoresGlob.c:(.text.startup+0xfc): reubicación truncada para ajustar: R_X86_64_325 contra el símbolo `v3' definido en la sección COMMON en /tmp/ccNUX0hm.o  
SumaVectoresGlob.c:(.text.startup+0x15): reubicación truncada para ajustar: R_X86_64_325 contra el símbolo `v2' definido en la sección COMMON en /tmp/ccNUX0hm.o  
SumaVectoresGlob.c:(.text.startup+0x12b): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo `v3' definido en la sección COMMON en /tmp/ccNUX0hm.o  
SumaVectoresGlob.c:(.text.startup+0x135): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo `v2' definido en la sección COMMON en /tmp/ccNUX0hm.o  
collect2: error: ld returned 1 exit status  
paula@paula-Aspire-F5-572G:~/Escritorio/AC/Pract0/VectoresGlob$
```

Ocurre un error al intentar compilar el archivo .c con “reubicación truncada para ajustar”. Esto se debe a que el segmento no es lo suficientemente grande para almacenar las variables.

El máximo número que se puede almacenar en N es $2^{32}-1$ porque N es un unsigned int, es decir, ocupa 4 Bytes = 32 bits. El mayor número que puede representarse en binario con 32 bits es $2^{32}-1$, ya que la última potencia es 2^{31} . Por ejemplo, si tenemos 4 bits (1 Byte) el mayor número que podemos representar es 15, o lo que es lo mismo, 2^4-1 .

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 PRINTE_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)

#ifndef 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
#ifndef PRINTE_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) /\n"
           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

#ifndef 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()

#ifndef 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)

#ifndef VECTOR_GLOBAL
#define MAX 33554432 //=2^25
double v1[MAX], v2[MAX], v3[MAX];
#endif

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

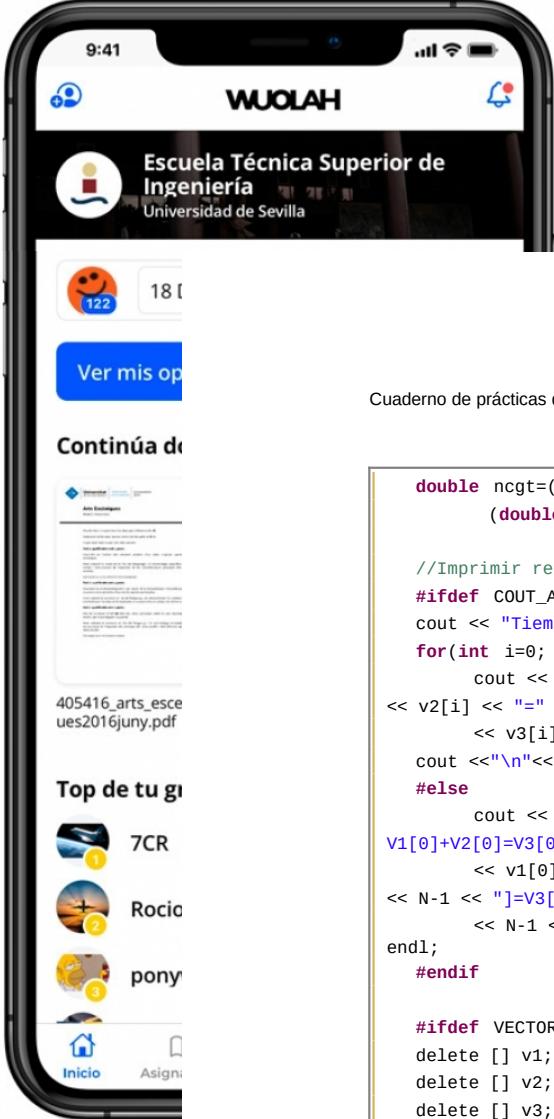
    struct timespec cgt1,cgt2; //para tiempo de ejecución

    //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]);
#ifndef VECTOR_LOCAL
    double v1[N], v2[N], v3[N];
#endif
#ifndef VECTOR_GLOBAL
    if (N>MAX) N=MAX;
#endif
#ifndef 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,&cg1);
    //Calcular suma de vectores
    for(int i=0; i<N; i++)
        v3[i] = v1[i] + v2[i];
    clock_gettime(CLOCK_REALTIME,&cg2);
}

```



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

Available on the
App Store

GET IT ON
Google Play

Cuaderno de prácticas de Arquitectura de Computadores, Grado en Ingeniería Informática

```
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 << "]=" << 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
```