

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): Javier Galera Garrido

Grupo de prácticas: B3

Fecha de entrega:

Fecha evaluación en clase:

[-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):

```
PS1="[NombreApellidos \u@\h:\w] \D{%F %A}\n$"
```

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.)]

Ejercicios basados en los ejemplos del seminario práctico

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

CAPTURAS:

Conteste a las siguientes preguntas:

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

RESPUESTA: En mi uso de ordenador propio utilizando `lscpu`, cuento con un único socket como microprocesador a 2 núcleos por socket, por tanto 2 cores físicos, y 4 lógicos (observar en captura de pantalla apartado CPU).

```
jota@jota-Erazer-P6679-MD60359: ~/Escritorio/segundo/ac/practicas
Archivo Editar Ver Buscar Terminal Ayuda
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: 2
Núcleo(s) por «socket»: 2
Socket(s): 1
Modo(s) NUMA: 1
ID de fabricante: GenuineIntel
Familia de CPU: 6
Modelo: 142
Model name: Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz
Revisión: 9
CPU MHz: 2700.000
CPU max MHz: 3100,0000
CPU min MHz: 400,0000
BogoMIPS: 5424.00
Virtualización: VT-x
Caché L1d: 32K
Caché L1i: 32K
Caché L2: 256K
Caché L3: 3072K
NUMA node0 CPU(s): 0-3
```

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

RESPUESTA:

```
jota@jota-Erazer-P6679-MD60359: ~/Escritorio/segundo/ac/practicas
Archivo Editar Ver Buscar Terminal Ayuda
[javierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/prac
ticas] 2018-02-27 martes
Scat STDIN.o62242
Arquitectura: 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: 1600.021
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

jota@jota-Erazer-P6679-MD60359: ~
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
B3estudiante21@atcgrid:~/practica1 jota@jota-Erazer-P6679-MD60359: ~
Xed/ ZFEffirefox.xcf
apache-jmeter-3.3.tgz apache-jmeter-3.3/
avilli bomba_javier_galera_garrido
capertaFran/ codigo.zip
core estudio50/
examen50nod2.zip home/
nmap.pdf p4Code/
passmysql pruebasSQL/
resumenes FR/ segundo/
sqldeveloper.conf tercero/

sftp> lcd Escritorio/segundo/
ac/ ec/ so/
sftp> lcd Escritorio/segundo/ac/practicas/
pract1/ prompt.sh userandpass
sftp> lcd Escritorio/segundo/ac/practicas/
pract1/ prompt.sh userandpass
sftp> lcd Escritorio/segundo/ac/practicas/
STDIN.o62242 STDIN.o62242
sftp> get STDIN.o62242
STDIN.o62242 STDIN.o62242
Fetching /home/B3estudiante21/practica1/STDIN.o62242 to STDIN.o62242
/home/B3estudiante21/practica1/STDIN.o62242 100% 1159 15.4KB/s 00:00
sftp> |
```

Cada nodo cuenta con doble socket de núcleo, con 6 cores físicos cada uno, lo que hace una suma de 12 cores físicos y 24 lógicos puesto que son hyperthreading.

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++:

$v_3 = v_1 + v_2$; $v_3(i) = v_1(i) + v_2(i)$, $i=0, \dots, N-1$

Los códigos utilizan directivas del compilador para fijar el tipo de variable de los vectores (v_1 , v_2 y v_3). 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:

The *res* and *tp* arguments are **timespec** structs, as specified in `<time.h>`:

```
struct timespec {
    time_t    tv_sec;        /* seconds */
    long      tv_nsec;       /* nanoseconds */
};
```

Fuente de imagen: https://linux.die.net/man/3/clock_gettime

1. Información de la variable `ncgt`: Obtenido del mismo código:

```
73 ncgt=(double) (cgt2.tv_sec-cgt1.tv_sec)+
74      (double) ((cgt2.tv_nsec-cgt1.tv_nsec)/(1.e+9));
```

Por tanto, la variable `ncgt` contiene la diferencia de los tiempos entre `cgt2` y `cgt1` aplicando la suma de nanosegundos a los segundos.

2. Información retornada por la función `clock_gettime()`:

`clock_gettime()` devuelve exactamente un valor entero, 0, en caso de éxito, ó -1 en caso de error.

3. Estructura de datos devuelta: llamarlo estructura de datos devuelta no es exactamente la explicación, el caso es que al llamar a la función **`clock_gettime()`**:

```
clock_gettime(CLOCK_REALTIME,&cgt1);
```

`cgt1` o `cgt2` son de una estructura llamada: “**timespec**” que contiene dos valores (segundos y nanosegundos); Al invocar la función le pasaremos una de estas dos estructuras por referencia, tal que guardará en sus variables los segundos y milisegundos exactos en lo que se realiza la función.

- 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++
Entre las bibliotecas usadas.	stdlib.h, stdio.h	cstdlib (el stdlib de c), iostream

Uso de namespaces.	En C no.	Using namespace std.
Reserva de la memoria dinámica.	Utiliza la función “ malloc ”.	Utiliza “ new ”, además con respecto a “ malloc ”, éste si te advierte del error en la reserva de memoria dinámica mediante una excepción.
Flujo de salida.	Utiliza “ printf ”.	Utiliza “ cout ”.
Liberar memoria dinámica.	Free (v1);	Delete [] v1;

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:

1. Mostramos contenido(el inicio) del archivo) para mostrar el código:

```

jota@jota-Erazer-P6679-MD60359: ~/Escritorio/segundo/ac/practicas
Archivo Editar Ver Buscar Terminal Ayuda
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/prac
ticas/pract1] 2018-02-27 martes
$cat sumavectores.c
/* SumaVectores.c
*/

#include <stdlib.h>
#include <stdio.h>
#include <time.h>

// #define PRINTF_ALL

#define VECTOR_LOCAL

// #define VECTOR_GLOBAL

// #define VECTOR_DYNAMIC

#ifdef VECTOR_GLOBAL
#define MAX 33554432
double v1[MAX], v2[MAX], v3[MAX];
#endif

```

2. Compilación del programa:

```
jota@jota-Erazer-P6679-MD60359: ~/Escritorio/segundo/ac/practicas
Archivo Editar Ver Buscar Terminal Ayuda
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/prac
ticas/pract1] 2018-02-27 martes
$gcc -O2 sumavectores.c -o sumavectoresc -lrt
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/prac
ticas/pract1] 2018-02-27 martes
$
```

3. Ejecución del programa:

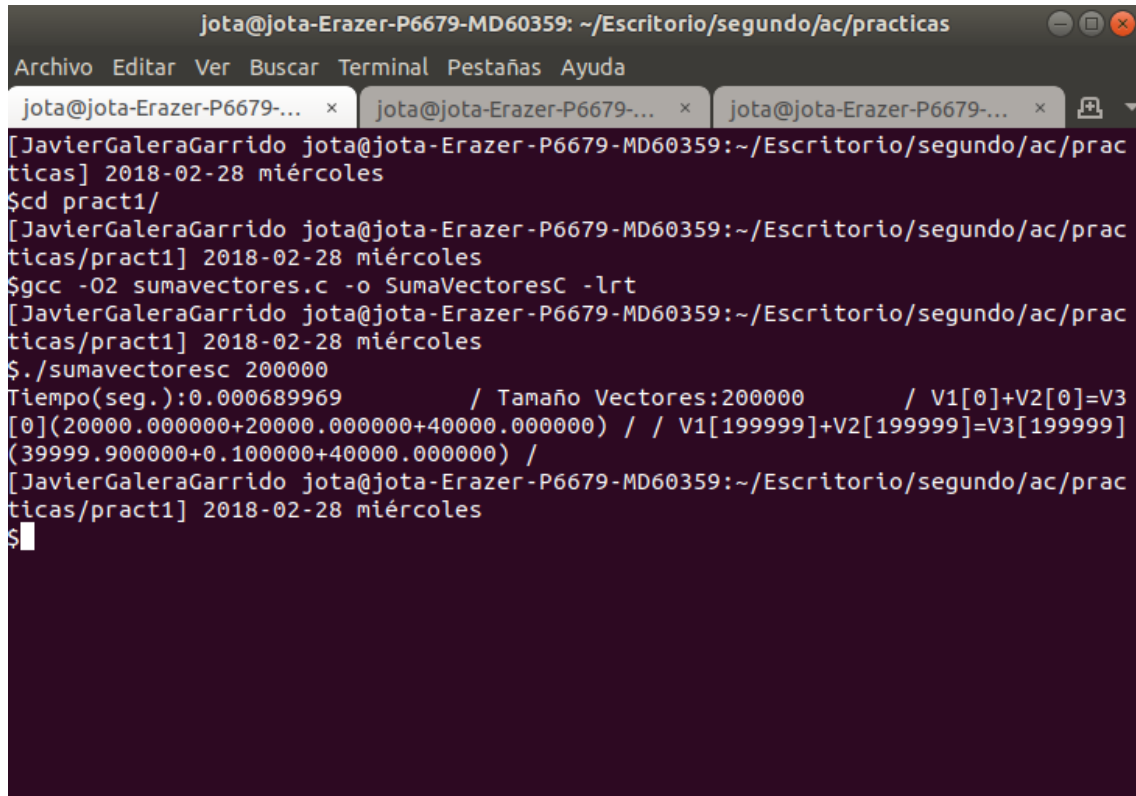
```
jota@jota-Erazer-P6679-MD60359: ~/Escritorio/segundo/ac/practicas
Archivo Editar Ver Buscar Terminal Ayuda
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/prac
ticas/pract1] 2018-02-27 martes
$gcc -O2 sumavectores.c -o sumavectoresc -lrt
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/prac
ticas/pract1] 2018-02-27 martes
$./sumavectoresc 12
Tiempo(seg.):0.000000637 / Tamaño Vectores:12 / V1[0]+V2[0]=V3[0](1.20
0000+1.200000+2.400000) / / V1[11]+V2[11]=V3[11](2.300000+0.100000+2.400000) /
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/prac
ticas/pract1] 2018-02-27 martes
$
```

Resto de ejercicios

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:

1º. En primer lugar compilamos y demostramos que la compilación ha tenido éxito:



```
jota@jota-Erazer-P6679-MD60359: ~/Escritorio/segundo/ac/practicas
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
jota@jota-Erazer-P6679-... x jota@jota-Erazer-P6679-... x jota@jota-Erazer-P6679-... x
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/prac
ticas] 2018-02-28 miércoles
$cd pract1/
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/prac
ticas/pract1] 2018-02-28 miércoles
$gcc -O2 sumavectores.c -o SumaVectoresC -lrt
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/prac
ticas/pract1] 2018-02-28 miércoles
$./sumavectoresc 200000
Tiempo(seg.):0.000689969 / Tamaño Vectores:200000 / V1[0]+V2[0]=V3
[0](20000.000000+20000.000000+40000.000000) / / V1[199999]+V2[199999]=V3[199999]
(39999.900000+0.100000+40000.000000) /
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/prac
ticas/pract1] 2018-02-28 miércoles
$
```

2º. En segundo lugar comprobamos que en local(es decir, en mi PC) la ejecución va perfecta, a excepción de que, como comenta en el propio código, introduzcas un valor igual o superior a 4294967295 en cuyo caso generará un error sobre la reserva de espacio para los vectores tal que:

```
jota@jota-Erazer-P6679-MD60359: ~/Escritorio/segundo/ac/practicas
Archivo Editar Ver Buscar Terminal Ayuda
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/prac
ticas/pract1] 2018-02-28 miércoles
$./SumaVectoresC 429496729342
Error en la reserva de espacio para los vectores
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/prac
ticas/pract1] 2018-02-28 miércoles
$
```

3º. Subimos el ejecutable y el script al servidor mediante sftp:

```
sftp> put SumaVectoresC
Uploading SumaVectoresC to /home/B3estudiante21/practica1/ejercicio4/SumaVectore
sC
SumaVectoresC                               100% 12KB 242.1KB/s 00:00
sftp> ls
SumaVectoresC
sftp> put Su
SumaVectores.sh                               SumaVectoresC                               SumaVectoresCpp.cpp

sftp> put SumaVectores.sh
Uploading SumaVectores.sh to /home/B3estudiante21/practica1/ejercicio4/SumaVecto
res.sh
SumaVectores.sh                               100% 751 22.6KB/s 00:00
sftp> ls
SumaVectores.sh                               SumaVectoresC
sftp>
```

4º. Ejecutamos mediante ssh en la cola “ac” el script:

```
[B3estudiante21@atcgrid ejercicio4]$ qsub -q ac SumaVectores.sh
62714.atcgrid
[B3estudiante21@atcgrid ejercicio4]$
```


5º. Recogemos de nuevo en nuestro PC el resultado de la ejecución mediante **get** y observamos el resultado:

```
sftp> get SumaVectoresC_vlocales.o62714
Fetching /home/B3estudiante21/practica1/ejercicio4/SumaVectoresC_vlocales.o62714 to SumaVectoresC_vlocales.o62714
/home/B3estudiante21/practica1/ejercicio4/SumaVectoresC_vloc 100% 2715 41.9KB/s 00:00
sftp>
```

```
SumaVectoresC_vlocales.o62714
~/Escritorio/segundo/ac/practicas/resultadosP1 Guardar
```

```
1 Id. usuario del trabajo: B3estudiante21
2 Id. del trabajo: 62714.atcgrid
3 Nombre del trabajo especificado por usuario: SumaVectoresC_vlocales
4 Nodo que ejecuta qsub: atcgrid
5 Directorio en el que se ha ejecutado qsub: /home/B3estudiante21/practica1/ejercicio4
6 Cola: ac
7 Nodos asignados al trabajo:
8 atcgrid1
9 atcgrid1
10 atcgrid1
11 atcgrid1
12 atcgrid1
13 atcgrid1
14 atcgrid1
15 atcgrid1
16 atcgrid1
17 atcgrid1
18 atcgrid1
19 atcgrid1
20 atcgrid1
21 atcgrid1
22 atcgrid1
23 atcgrid1
24 atcgrid1
25 atcgrid1
26 atcgrid1
27 atcgrid1
28 atcgrid1
29 atcgrid1
30 atcgrid1
31 atcgrid1
32 Tiempo(seg.):0.000424049 / Tamaño Vectores:65536 /
33 V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / /
34 V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
35 Tiempo(seg.):0.000839867 / Tamaño Vectores:131072 /
36 V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / /
37 V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
38 Tiempo(seg.):0.001713140 / Tamaño Vectores:262144 /
39 V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / /
40 V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
41 Tiempo(seg.):0.002413193 / Tamaño Vectores:524288 /
42 V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / /
43 V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
44 Tiempo(seg.):0.005977964 / Tamaño Vectores:1048576 /
45 V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / /
46 V1[1048575]+V2[1048575]=V3[1048575](209715.100000+0.100000=209715.200000) /
47 Tiempo(seg.):0.012023733 / Tamaño Vectores:2097152 /
48 V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / /
49 V1[2097151]+V2[2097151]=V3[2097151](419430.300000+0.100000=419430.400000) /
50 Tiempo(seg.):0.023847535 / Tamaño Vectores:4194304 /
51 V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / /
52 V1[4194303]+V2[4194303]=V3[4194303](838860.700000+0.100000=838860.800000) /
53 Tiempo(seg.):0.046890835 / Tamaño Vectores:8388608 /
54 V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / /
55 V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000) /
```

Texto plano ▾ Anchura del tabulador: 2 ▾ Ln 1, Col 1 ▾ INS

Nota: Completa el bucle for entero hasta el tamaño máximo alcanzado en él sin generar ningún error, esta aclaración es debida a que en el pantallazo no es posible recoger los últimos datos.

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:

El error en la version de ubuntu 17.x no muestra por pantalla el warning al compilar el ejecutable de `SumaVectoresGlobales`, pero aún así podemos comprobarlo en el resultado saltándose la última ejecución del bucle y repitiendo la penúltima. Esto es así debido a que utilizando las variables globales, la memoria se asigna en tiempo de compilación.

1º He creado el ejecutable de **GLOBALES** en mi pc y lo he ejecutado tal que:

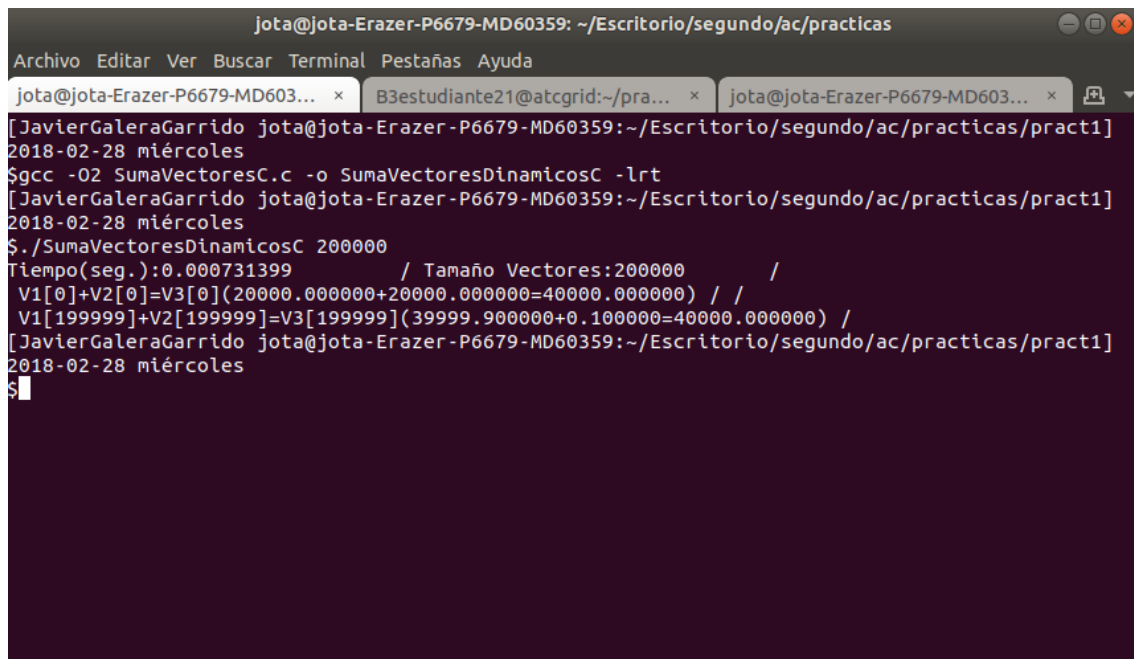
```
jota@jota-Erazer-P6679-MD60359: ~/Escritorio/segundo/ac/practicas
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
jota@jota-Erazer-P6679-MD60359: ~/Escritorio/segundo/ac/practicas/pract1
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/practicas/pract1]
2018-02-28 miércoles
$gcc -O2 SumaVectoresC.c -o SumaVectoresGlobalesC -lrt
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/practicas/pract1]
2018-02-28 miércoles
$./SumaVectoresGlobalC 200000
bash: ./SumaVectoresGlobalC: No existe el archivo o el directorio
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/practicas/pract1]
2018-02-28 miércoles
$./SumaVectoresGlobalesC 200000
Tiempo(seg.):0.000732756 / Tamaño Vectores:200000 /
V1[0]+V2[0]=V3[0](20000.000000+20000.000000=40000.000000) / /
V1[199999]+V2[199999]=V3[199999](39999.900000+0.100000=40000.000000) /
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/practicas/pract1]
2018-02-28 miércoles
$
```

Pantallazo del error:

```
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/prac
ticas/pract1] 2018-03-02 viernes
$./SumaVectoresGlobalesC 67108864
Tiempo(seg.):0.119486451 / 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)
/
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/prac
ticas/pract1] 2018-03-02 viernes
$
```

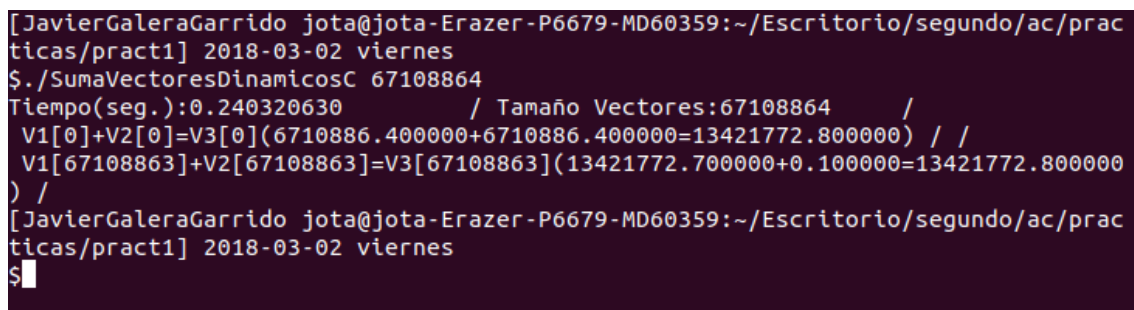
Podemos comprobar como aun exigiendo la ejecución con un valor para “N” de **67108864** podemos comprobar que se queda en la anterior ejecución, la del valor 33554432.

2º Mismo caso para los **DINAMICOS**:



```
jota@jota-Erazer-P6679-MD60359: ~/Escritorio/segundo/ac/practicas
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
jota@jota-Erazer-P6679-MD603... x B3estudiante21@atcgrid:~/pra... x jota@jota-Erazer-P6679-MD603... x
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/practicas/pract1]
2018-02-28 miércoles
$gcc -O2 SumaVectoresC.c -o SumaVectoresDinamicosC -lrt
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/practicas/pract1]
2018-02-28 miércoles
$./SumaVectoresDinamicosC 200000
Tiempo(seg.):0.000731399 / Tamaño Vectores:200000 /
V1[0]+V2[0]=V3[0](20000.000000+20000.000000=40000.000000) / /
V1[199999]+V2[199999]=V3[199999](39999.900000+0.100000=40000.000000) /
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/practicas/pract1]
2018-02-28 miércoles
$
```

Comprobación del buen funcionamiento en vectores dinámicos:



```
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/prac
ticas/pract1] 2018-03-02 viernes
$./SumaVectoresDinamicosC 67108864
Tiempo(seg.):0.240320630 / Tamaño Vectores:67108864 /
V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) / /
V1[67108863]+V2[67108863]=V3[67108863](13421772.700000+0.100000=13421772.800000
) /
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/prac
ticas/pract1] 2018-03-02 viernes
$
```

3º He subido los archivos al servidor mediante **SFTP**:

```
jota@jota-Erazer-P6679-MD60359: ~/Escritorio/segundo/ac/practicas/pract1
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
jota@jota-Erazer-P6679-... x B3estudiante21@atcgrid:... x jota@jota-Erazer-P6679-... x
sftp> put Sum
SumaVectores.sh                               SumaVectoresC
SumaVectoresC.c                               SumaVectoresCpp.cpp
SumaVectoresDinamicosC                       SumaVectoresGlobalYDinamic.sh
SumaVectoresGlobalesC                       SumaVectoresLocalesC

sftp> put SumaVectoresGl
SumaVectoresGlobalYDinamic.sh                 SumaVectoresGlobalesC

sftp> put SumaVectoresGlobalYDinamic.sh
Uploading SumaVectoresGlobalYDinamic.sh to /home/B3estudiante21/practica1/ejerci
cio5/SumaVectoresGlobalYDinamic.sh
SumaVectoresGlobalYDinamic.sh                 100% 924    27.7KB/s   00:00
sftp> ls
SumaVectoresDinamicosC                       SumaVectoresGlobalYDinamic.sh
SumaVectoresGlobalesC
sftp> ls -l
-rwxr-xr-x    1 B3estudiante21 B3estudiante21    12720 Mar  2 21:23 SumaVectores
DinamicosC
-rwxrwxr-x    1 B3estudiante21 B3estudiante21      924 Mar  2 21:24 SumaVectores
GlobalYDinamic.sh
-rwxr-xr-x    1 B3estudiante21 B3estudiante21    8624 Mar  2 21:23 SumaVectores
GlobalesC
sftp>
```

4º Ejecución en el servidor mediante **SSH**:

```
[B3estudiante21@atcgrid ejercicio5]$ ls
SumaVectoresDinamicosC SumaVectoresGlobalesC SumaVectoresGlobalYDinamic.sh
[B3estudiante21@atcgrid ejercicio5]$ qsub -q ac SumaVectoresGlobalYDinamic.sh
63738.atcgrid
[B3estudiante21@atcgrid ejercicio5]$ ls -l
total 40
-rw----- 1 B3estudiante21 B3estudiante21      0 mar  2 21:26 SumaVectoresC_glob
al_dynamic.e63738
-rw----- 1 B3estudiante21 B3estudiante21    4988 mar  2 21:26 SumaVectoresC_glob
al_dynamic.o63738
-rwxr-xr-x 1 B3estudiante21 B3estudiante21   12720 mar  2 21:23 SumaVectoresDinami
cosC
-rwxr-xr-x 1 B3estudiante21 B3estudiante21    8624 mar  2 21:23 SumaVectoresGlobal
esC
-rwxrwxr-x 1 B3estudiante21 B3estudiante21      924 mar  2 21:24 SumaVectoresGlobal
YDinamic.sh
[B3estudiante21@atcgrid ejercicio5]$
```

5º Obtención de datos mediante GET:

```
sftp> get SumaVectoresC
SumaVectoresC_global_dynamic.e63738      SumaVectoresC_global_dynamic.o63738

sftp> get SumaVectoresC_global_dynamic.o63738
Fetching /home/B3estudiante21/practica1/ejercicio5/SumaVectoresC_global_dynamic.
o63738 to SumaVectoresC_global_dynamic.o63738
/home/B3estudiante21/practica1/ejercicio5/Sum 100% 4988    75.9KB/s    00:00
sftp> █
```

6º Visualición de datos:

GLOBALES

```

Abrir ▾  SumaVectoresC_global_dynamic.o63738  Guardar  ≡  ⌵  ⌵  ⌵  ⌵
~/Escritorio/segundo/ac/practicas/pract1/ejercicio5

1 ID. usuario del trabajo: B3estudiante21
2 Id. del trabajo: 63738.atcgrid
3 Nombre del trabajo especificado por usuario: SumaVectoresC_global_dynamic
4 Nodo que ejecuta qsub: atcgrid
5 Directorio en el que se ha ejecutado qsub: /home/B3estudiante21/practica1/ejercicio5
6 Cola: ac
7 Nodos asignados al trabajo:
8 atcgrid1
9 atcgrid1
10 atcgrid1
11 atcgrid1
12 atcgrid1
13 atcgrid1
14 atcgrid1
15 atcgrid1
16 atcgrid1
17 atcgrid1
18 atcgrid1
19 atcgrid1
20 atcgrid1
21 atcgrid1
22 atcgrid1
23 atcgrid1
24 atcgrid1
25 atcgrid1
26 atcgrid1
27 atcgrid1
28 atcgrid1
29 atcgrid1
30 atcgrid1
31 atcgrid1
32 Ejecucion de SumaVectoresGlobalC
33 Tiempo(seg.):0.000429271 / Tamaño Vectores:65536 /
34 V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / /
35 V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
36 Tiempo(seg.):0.000554144 / Tamaño Vectores:131072 /
37 V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / /
38 V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
39 Tiempo(seg.):0.001490986 / Tamaño Vectores:262144 /
40 V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / /
41 V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
42 Tiempo(seg.):0.002603872 / Tamaño Vectores:524288 /
43 V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / /
44 V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
45 Tiempo(seg.):0.005686723 / Tamaño Vectores:1048576 /
46 V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / /
47 V1[1048575]+V2[1048575]=V3[1048575](209715.100000+0.100000=209715.200000) /
48 Tiempo(seg.):0.011786636 / Tamaño Vectores:2097152 /
49 V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / /
50 V1[2097151]+V2[2097151]=V3[2097151](419430.300000+0.100000=419430.400000) /
51 Tiempo(seg.):0.023626420 / Tamaño Vectores:4194304 /
52 V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / /
53 V1[4194303]+V2[4194303]=V3[4194303](838860.700000+0.100000=838860.800000) /
54 Tiempo(seg.):0.046980400 / Tamaño Vectores:8388608 /
55 V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / /

Texto plano ▾  Anchura del tabulador: 2 ▾  Ln 1, Col 1 ▾  INS

```

DINÁMICOS

```

55 V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / /
56 V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000) /
57 Tiempo(seg.):0.089131576 / Tamaño Vectores:16777216 /
58 V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / /
59 V1[16777215]+V2[16777215]=V3[16777215](3355443.100000+0.100000=3355443.200000) /
60 Tiempo(seg.):0.177790443 / Tamaño Vectores:33554432 /
61 V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / /
62 V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
63 Tiempo(seg.):0.184170684 / Tamaño Vectores:33554432 /
64 V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / /
65 V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
66 Ejecucion de SumaVectoresDynamicC
67 Tiempo(seg.):0.000415299 / Tamaño Vectores:65536 /
68 V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / /
69 V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
70 Tiempo(seg.):0.000831925 / Tamaño Vectores:131072 /
71 V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / /
72 V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
73 Tiempo(seg.):0.001682979 / Tamaño Vectores:262144 /
74 V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / /
75 V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
76 Tiempo(seg.):0.003491202 / Tamaño Vectores:524288 /
77 V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / /
78 V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
79 Tiempo(seg.):0.005831514 / Tamaño Vectores:1048576 /
80 V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / /
81 V1[1048575]+V2[1048575]=V3[1048575](209715.100000+0.100000=209715.200000) /
82 Tiempo(seg.):0.011953506 / Tamaño Vectores:2097152 /
83 V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / /
84 V1[2097151]+V2[2097151]=V3[2097151](419430.300000+0.100000=419430.400000) /
85 Tiempo(seg.):0.023764604 / Tamaño Vectores:4194304 /
86 V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / /
87 V1[4194303]+V2[4194303]=V3[4194303](838860.700000+0.100000=838860.800000) /
88 Tiempo(seg.):0.047384664 / Tamaño Vectores:8388608 /
89 V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / /
90 V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000) /
91 Tiempo(seg.):0.090985307 / Tamaño Vectores:16777216 /
92 V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / /
93 V1[16777215]+V2[16777215]=V3[16777215](3355443.100000+0.100000=3355443.200000) /
94 Tiempo(seg.):0.182241159 / Tamaño Vectores:33554432 /
95 V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / /
96 V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
97 Tiempo(seg.):0.368169582 / Tamaño Vectores:67108864 /
98 V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) / /
99 V1[67108863]+V2[67108863]=V3[67108863](13421772.700000+0.100000=13421772.800000) /

```

Texto plano ▾ Anchura del tabulador: 2 ▾ Ln 1, Col 1

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:

Sí, en mi caso, se puede comprobar que mi PC obtiene mejores resultados que el servidor atcgrid. Mirar carpeta: **ejercicio 6**

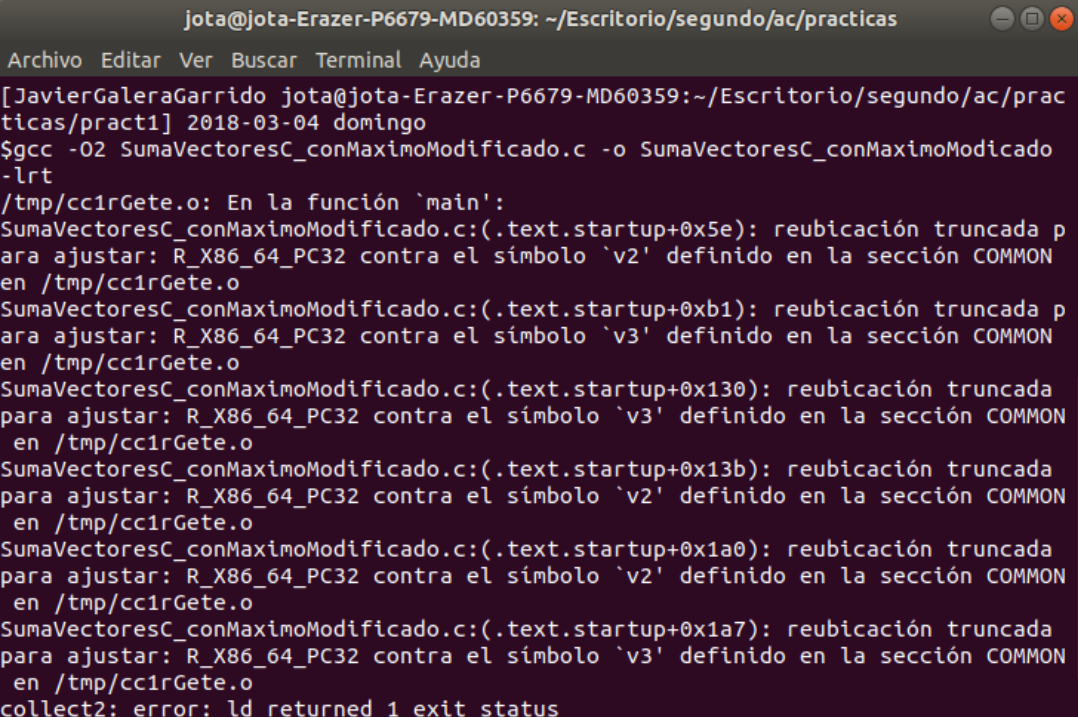
7. Modificar el código fuente C para que el límite de los vectores cuando se declaran como variables globales sea igual al máximo número que se puede almacenar en la variable N ($MAX=2^{32}-1$).

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:

Al intentar compilar el archivo SumaVectoresC_conMaximoModificado.c aparece un error porque se ha excedido el tamaño máximo permitido para el programa y no puede generar el ejecutable ya que no cabe.

El máximo valor que puede almacenar un tipo de dato unsigned int es $2^{32}-1$ porque el tamaño de unsigned int es de 4B, que pasado a bit es 32b ($4*8$) y por tanto, tenemos 2^{32} posibilidades de números para representar (sin contar con números negativos).



```
jota@jota-Erazer-P6679-MD60359: ~/Escritorio/segundo/ac/practicas
Archivo Editar Ver Buscar Terminal Ayuda
[JavierGaleraGarrido jota@jota-Erazer-P6679-MD60359:~/Escritorio/segundo/ac/practicas/pract1] 2018-03-04 domingo
$gcc -O2 SumaVectoresC_conMaximoModificado.c -o SumaVectoresC_conMaximoModificado -lrt
/tmp/cc1rGete.o: En la función `main':
SumaVectoresC_conMaximoModificado.c:(.text.startup+0x5e): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo `v2' definido en la sección COMMON en /tmp/cc1rGete.o
SumaVectoresC_conMaximoModificado.c:(.text.startup+0xb1): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo `v3' definido en la sección COMMON en /tmp/cc1rGete.o
SumaVectoresC_conMaximoModificado.c:(.text.startup+0x130): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo `v3' definido en la sección COMMON en /tmp/cc1rGete.o
SumaVectoresC_conMaximoModificado.c:(.text.startup+0x13b): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo `v2' definido en la sección COMMON en /tmp/cc1rGete.o
SumaVectoresC_conMaximoModificado.c:(.text.startup+0x1a0): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo `v2' definido en la sección COMMON en /tmp/cc1rGete.o
SumaVectoresC_conMaximoModificado.c:(.text.startup+0x1a7): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo `v3' definido en la sección COMMON en /tmp/cc1rGete.o
collect2: error: ld returned 1 exit status
```

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

```