

BP0RodriguezJimenezFracisco.pdf

**CAZZ****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**

Arquitectura de Computadores (AC)

Cuaderno de prácticas. Bloque Práctico 0. Entorno de programación

Estudiante (nombre y apellidos): Francisco Rodríguez Jiménez
Grupo de prácticas y profesor de prácticas: C1 Juan José Escobar
Fecha de entrega:
Fecha evaluación en clase:

Antes de comenzar a realizar el trabajo de este cuaderno consultar el fichero con los normas de prácticas que se encuentra en SWAD

Parte I. Ejercicios basados en los ejemplos del seminario práctico

Crear el directorio con nombre bp0 en atcgrid y en el PC local.

1. Ejecutar lscpu en el PC y en un nodo de cómputo de atcgrid.

(a) Mostrar con capturas de pantalla el resultado de estas ejecuciones.

RESPUESTA:

```
lscpu en PC
```

```
lscpu en atcgrid
```

```
lscpu en atcgrid
```

lscpu en atcgrid

(b) ¿Cuántos cores físicos y cuántos cores lógicos tienen los nodos de cómputo de atcgrid y del PC? Razonar las respuestas

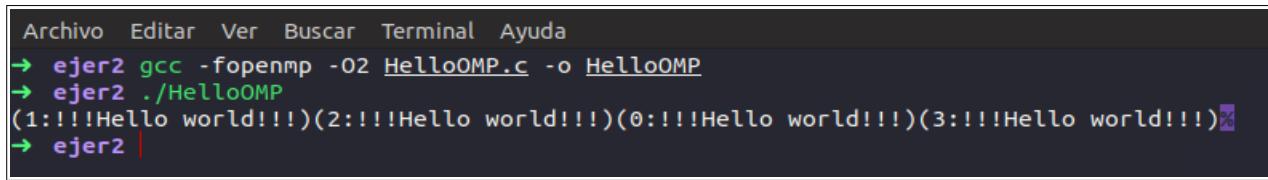
RESPUESTA:

- PC: Tiene **1 procesador** con **4 núcleos físicos** y **1 línea de procesamiento (Thread)** por núcleo. En total son **4 CPU(s)**.

- **Atcgrid:** Tiene **2 sockets** en la placa base, con lo que tiene **2 procesadores**, cada uno de ellos tiene, **6 núcleos físicos** y por cada núcleo **2 líneas de procesamiento (Thread)**, en total cada socket tiene 12 CPU(s), en total **24 CPU(s)** por cada nodo de computo atcgrid.
2. Compilar y ejecutar en el PC el código HelloOMP.c del seminario (recordar que se debe usar un directorio independiente para cada ejercicio dentro de bp0 que contenga todo lo utilizado, implementado o generado durante el desarrollo del mismo, para el presente ejercicio el directorio sería ejer2, como se indica en las normas de prácticas).

(a) Adjuntar capturas de pantalla que muestren la compilación y ejecución en el PC.

RESPUESTA:



```
Archivo Editar Ver Buscar Terminal Ayuda
→ ejer2 gcc -fopenmp -O2 HelloOMP.c -o HelloOMP
→ ejer2 ./HelloOMP
(1:!!!Hello world!!!)(2:!!!Hello world!!!)(0:!!!Hello world!!!)(3:!!!Hello world!!!)
→ ejer2 |
```

HelloOMP en PC

(b) Justificar el número de “Hello world” que se imprimen en pantalla en ambos casos teniendo en cuenta la salida que devuelve lscpu.

RESPUESTA:

Como hemos visto en la información aportada por el comando ‘lscpu’ que el número de **CPU(s) del PC es 4**, al ejecutar HelloOMP, el número de “Hello world” impresos paralelamente son 4 ya que cada hebra ejecuta la parte del código que esta debajo de **#pragma omp parallel** en el código HelloOMP.c

3. Copiar el ejecutable de HelloOMP.c que ha generado anteriormente y que se encuentra en el directorio ejer2 del PC al directorio ejer2 de su home en el *front-end* de atcgrid. Ejecutar (desde el directorio de este ejercicio, ejer3) este código en un nodo de cómputo de atcgrid usando la cola ac del gestor de colas (no use ningún script).

(a) Adjuntar capturas de pantalla que muestren la copia del fichero, el envío a la cola de la ejecución y el resultado de esta ejecución tal y como la devuelve el gestor de colas.

RESPUESTA:



```
D1estudiante20@atcgrid:~/bp0/ejer3
Archivo Editar Ver Buscar Terminal Ayuda
[FranciscoRodriguezJlmenez] D1estudiante20 @ ~/bp0/ejer3
01:44:52 $ echo "Obvio la copia del fichero, se hace con scp en ssh, o mediante los comandos tipicos de una conexion sftp"
Obvio la copia del fichero, se hace con scp en ssh, o mediante los comandos tipicos de una conexion sftp
[FranciscoRodriguezJlmenez] D1estudiante20 @ ~/bp0/ejer3
01:45:18 $ echo './HelloOMP' | qsub -q ac
4644.atcgrid
[FranciscoRodriguezJlmenez] D1estudiante20 @ ~/bp0/ejer3
01:45:36 $ cat STDIN.04644
(2:!!!Hello world!!!)(7:!!!Hello world!!!)(9:!!!Hello world!!!)(1:!!!Hello world!!!)(5:!!!Hello world!!!)(17:!!!Hello world!!!)(20:!!!Hello world!!!)(12:!!!Hello world!!!)(21:!!!Hello world!!!)(3:!!!Hello world!!!)(6:!!!Hello world!!!)(15:!!!Hello world!!!)(0:!!!Hello world!!!)(13:!!!Hello world!!!)(4:!!!Hello world!!!)(18:!!!Hello world!!!)(1:!!!Hello world!!!)(10:!!!Hello world!!!)(16:!!!Hello world!!!)(8:!!!Hello world!!!)(14:!!!Hello world!!!)(23:!!!Hello world!!!)
(19:!!!Hello world!!!)
```

HelloOMP en atcgrid

(b) Justificar el número de “Hello world” que se observan en el resultado teniendo en cuenta la salida que devuelve lscpu.

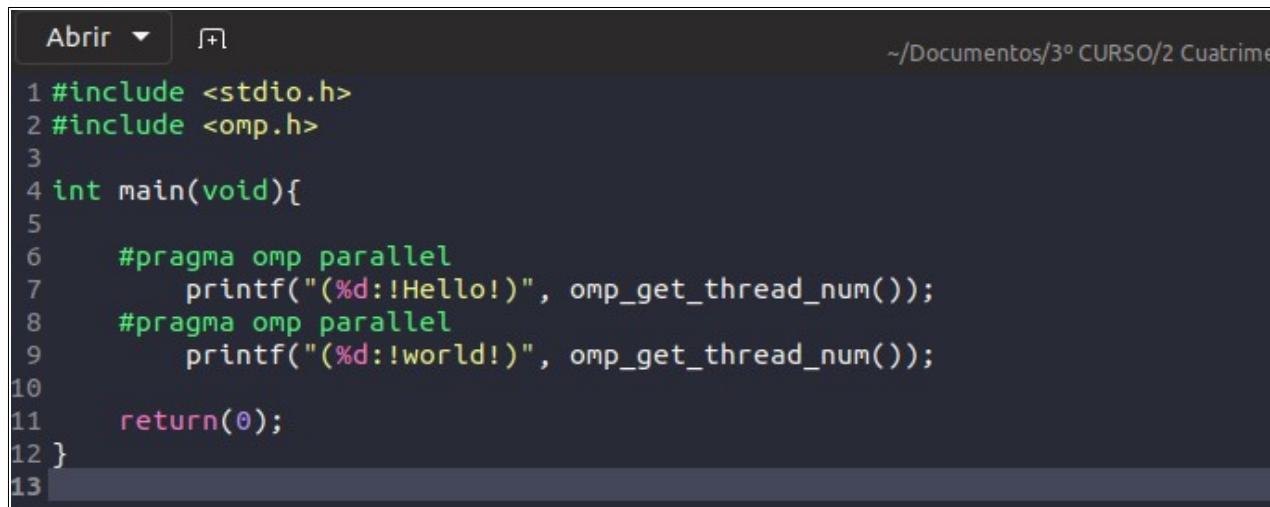
RESPUESTA:

Como hemos visto en la información aportada por el comando '**lscpu**' que el número de **CPU(s) de un nodo de computo atcgrid es 24**, al ejecutar HelloOMP, el número de "**Hello world**" impresos paralelamente son 24 ya que cada hebra ejecuta la parte del código que esta debajo de **#pragma omp parallel** en el código HelloOMP.c

4. Modificar en su PC HelloOMP.c para que se imprima "world" en un printf distinto al usado para "Hello", en ambos printf se debe imprimir el identificador del thread que escribe en pantalla. Nombrar al código resultante HelloOMP2.c. Compilar este nuevo código en el PC y ejecutarlo. Copiar el fichero ejecutable resultante en el front-end de atcgrid (directorio ejer4). Ejecutar el código en un nodo de cómputo de atcgrid usando el script script_helloomp.sh del seminario (el nombre del ejecutable en el script debe ser HelloOMP2).

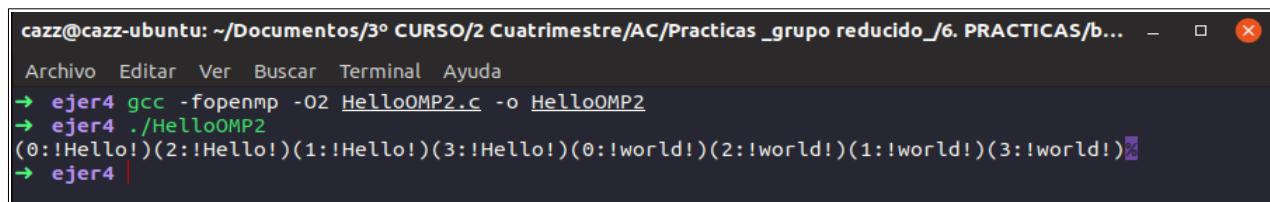
(a) Adjuntar capturas de pantalla que muestren el nuevo código, la compilación, la copia a atcgrid, el envío a la cola de la ejecución y el resultado de esta ejecución tal y como la devuelve el gestor de colas.

RESPUESTA:



```
1 #include <stdio.h>
2 #include <omp.h>
3
4 int main(void){
5
6     #pragma omp parallel
7         printf("(%d:!Hello!)", omp_get_thread_num());
8     #pragma omp parallel
9         printf("(%d:!world!)", omp_get_thread_num());
10
11    return(0);
12 }
13
```

código de HelloOMP2



```
cazz@cazz-ubuntu: ~/Documentos/3º CURSO/2 Cuatrimestre/AC/Practicas _grupo reducido/_6. PRACTICAS/b... - □ ✎
Archivo Editar Ver Buscar Terminal Ayuda
→ ejer4 gcc -fopenmp -O2 HelloOMP2.c -o HelloOMP2
→ ejer4 ./HelloOMP2
(0:!Hello!)(2:!Hello!)(1:!Hello!)(3:!Hello!)(0:!world!)(2:!world!)(1:!world!)(3:!world!)
→ ejer4 |
```

ejecución de HelloOMP2 en PC

ejecución de HelloOMP2 en atcgrid

(b) ¿Qué nodo de cómputo de atcgrid ha ejecutado el script? Explicar cómo ha obtenido esta información.

RESPUESTA:

El nodo de computo **atcgrid2**, mostrado en el script, contenido en la variable **\$PBS_NODEFILE**

(c) ¿Qué ocurre si se ejecuta el script usando `./HelloOMP2` en lugar de `$PBS_O_WORKDIR>HelloOMP2`? Razonar respuesta y adjuntar capturas de pantalla que muestren lo que ocurre.

RESPUESTA:

No se ejecuta ya que se necesita poner en el script, la ruta absoluta donde se encuentra el programar a ejecutar, y eso es lo que hace la variable **\$PBS_O_WORKDIR**, esta variable contiene la ruta absoluta del directorio donde se manda la orden para enviar a ejecutar a atcgrid. Si se ejecuta usando ./HelloOMP2 no encontrara el fichero.

```
[FranciscoRodriguezJimenez] Diestudiante20 @ ~/bp0/ejer4
17:43:23 $ qsub $(pwd)/script.sh -q ac
9374.atcgrid

[FranciscoRodriguezJimenez] Diestudiante20 @ ~/bp0/ejer4
17:43:26 $ ls
codigo.png ejecucion_atcgrid.png ejecucion_pc.png HelloOMP2 HelloOMP2.c helloomp.e9374 helloomp.o9374 script.sh

[FranciscoRodriguezJimenez] Diestudiante20 @ ~/bp0/ejer4
17:43:27 $ cat helloomp.o9374
Id. usuario del trabajo: Diestudiante20
Id. del trabajo: 9374.atcgrid
Nombre del trabajo especificado por usuario: helloomp
Directorio en el que se ha ejecutado qsub: /home/Diestudiante20/bp0/ejer4
Directorio de trabajo: /home/Diestudiante20
Cola: ac
Nodo que ejecuta qsub: atcgrid
Nodos asignados al trabajo:
atcgrid3
Nº de threads inicial: 12
Directorio de trabajo: /home/Diestudiante20/bp0/ejer4

1.Para 12 threads:

1.Para 6 threads:

1.Para 3 threads:

1.Para 1 threads:

[FranciscoRodriguezJimenez] Diestudiante20 @ ~/bp0/ejer4
17:43:30 $ cat helloomp.e9374
/var/spool/pbs/mom_priv/jobs/9374.atcgrid.SC: linea 30: ./HelloOMP2: No existe el fichero o el directorio
/var/spool/pbs/mom_priv/jobs/9374.atcgrid.SC: linea 30: ./HelloOMP2: No existe el fichero o el directorio
/var/spool/pbs/mom_priv/jobs/9374.atcgrid.SC: linea 30: ./HelloOMP2: No existe el fichero o el directorio
/var/spool/pbs/mom_priv/jobs/9374.atcgrid.SC: linea 30: ./HelloOMP2: No existe el fichero o el directorio
```



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

Continúa de

Parte II. Resto de ejercicios

RESPUESTA:

- Generar en el PC 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). El comentario inicial del código muestra la orden para compilar (siempre hay que usar -O2 al compilar como se indica en las normas de prácticas). Incorporar volcados de pantalla que demuestren la compilación y la ejecución correcta del código en el PC (leer lo indicado al respecto en las normas de prácticas).

RESPUESTA:

```
cazz@cazz-ubuntu:~/Documentos/3º CURSO/2 Cuatrimestre/AC/Prácticas_grupo reducido/_6. PRACTICAS/bp0/ejer5
Archivo Editar Ver Buscar Terminal Ayuda
$ jers5 gcc -O2 SumaVectoresC -o SumaVectoresC
$ ejers5 ./SumaVectoresC
Tamaño Vectores:5 (4 B)
Tamaño Vectores:5 / Tamaño Vectores:5
V1[0]+V2[0]=V3[0](0.000000+500000i-0.000000) /
V1[1]+V2[1]=V3[1](0.000000+0.400000i-1.000000) /
V1[2]+V2[2]=V3[2](0.700000+0.300000i-1.000000) /
V1[3]+V2[3]=V3[3](0.000000+0.000000i-1.000000) /
V1[4]+V2[4]=V3[4](0.900000+0.000000i-1.000000) /
$ ejers5 ./sumaVectoresC
Tamaño Vectores:1000000 (0 B)
Tamaño Vectores:1000000 / Tamaño Vectores:1000000 / V1[0]+V2[0]=V3[0](10000.000000+10000.000000+20000.000000) / V1[99999]+V2[99999]=V3[99999](19999.900000+0.100000=20000.000000) /
$ ejers5
```

SumaVectoresC en PC

- En el código del Listado 1 se utiliza la función `clock_gettime()` para obtener el tiempo de ejecución del trozo de código que calcula la suma de vectores. El código se imprime la variable `ncgt`,

(a) ¿qué contiene esta variable?

RESPUESTA:

La variable de tipo **double** contiene, el tiempo en (segundos+nanosegundos), para ser mas preciso se toman dos medidas, y son guardadas en esta variable `ncgt`.

(b) ¿en qué estructura de datos devuelve `clock_gettime()` la información de tiempo (indicar el tipo de estructura de datos, describir la estructura de datos, e indicar los tipos de datos que usa)?

RESPUESTA:

En esta estructura “**struct timespec**”, esta estructura de datos tiene dos componentes:

.time `tv_sec` : tiempo transcurrido en segundos.

.time `tv_nsec` : tiempo transcurrido en nanosegundos.

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

(c) ¿qué información devuelve exactamente la función `clock_gettime()` en la estructura de datos descrita en el apartado (b)? ¿qué representan los valores numéricos que devuelve?

RESPUESTA:

`int clock_gettime(clockid_t clk_id, struct timespec *tp);`

`clock_gettime(CLOCK_REALTIME,&cgt1);` → guarda en la estructura `cgt1`, el instante de tiempo.

Devuelve 0, si se ha ejecutado la función correctamente, y 1 si no.

- Ejecutar en atcgrid el código generado en el apartado anterior usando el script del Listado 2. Ejecutar el código también en el PC para los mismos tamaños de los vectores. ¿Se obtiene error para alguno de los tamaños? En caso afirmativo, ¿a qué se debe este error? (Incorporar volcados de pantalla como se indica en las normas de prácticas)

RESPUESTA:

```
|→ ejer7 ./sumaVectores_pc.sh
Tamanio Vectores:65536 (4 B)
Tiempo:0.000195543      / Tamanio 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) /
Tamanio Vectores:131072 (4 B)
Tiempo:0.000380115      / Tamanio 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) /
Tamanio Vectores:262144 (4 B)
Tiempo:0.000868713      / Tamanio 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) /
Tamanio Vectores:524288 (4 B)
./sumaVectores_pc.sh: línea 5: 11162 Violación de segmento ('core' generado) ./SumaVectoresC $N
Tamanio Vectores:1048576 (4 B)
./sumaVectores_pc.sh: línea 5: 11164 Violación de segmento ('core' generado) ./SumaVectoresC $N
Tamanio Vectores:2097152 (4 B)
./sumaVectores_pc.sh: línea 5: 11166 Violación de segmento ('core' generado) ./SumaVectoresC $N
Tamanio Vectores:4194304 (4 B)
./sumaVectores_pc.sh: línea 5: 11168 Violación de segmento ('core' generado) ./SumaVectoresC $N
Tamanio Vectores:8388608 (4 B)
./sumaVectores_pc.sh: línea 5: 11170 Violación de segmento ('core' generado) ./SumaVectoresC $N
Tamanio Vectores:16777216 (4 B)
./sumaVectores_pc.sh: línea 5: 11172 Violación de segmento ('core' generado) ./SumaVectoresC $N
Tamanio Vectores:33554432 (4 B)
./sumaVectores_pc.sh: línea 5: 11174 Violación de segmento ('core' generado) ./SumaVectoresC $N
Tamanio Vectores:67108864 (4 B)
./sumaVectores_pc.sh: línea 5: 11176 Violación de segmento ('core' generado) ./SumaVectoresC $N
→ ejer7 |
```

resultados en PC

```
D1estudiante20@atcgrid:~/bp0/ejer7
Archivo Editar Ver Buscar Terminal Ayuda
[FranciscoRodriguezJLmenez] D1estudiante20 @ ~/bp0/ejer7
15:48:34 $ qsub ./sumaVectores.sh -q ac
7122.atcgrid

[FranciscoRodriguezJLmenez] D1estudiante20 @ ~/bp0/ejer7
16:04:20 $ qsub ./sumaVectores_vlocales.07122
Identificador del trabajo: D1estudiante20
Id. del trabajo: 7122.atcgrid
Nombre del trabajo especificado por usuario: SumaVectoresC_vlocales
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/D1estudiante20/bp0/ejer7
Cola: ac
Nodos asignados al trabajo:
atcgrid
Tamanio Vectores:65536 (4 B)
Tiempo:0.000472580      / Tamanio 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) /
Tamanio Vectores:131072 (4 B)
Tiempo:0.000942333      / Tamanio 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) /
Tamanio Vectores:262144 (4 B)
Tiempo:0.001876233      / Tamanio 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) /
Tamanio Vectores:524288 (4 B)
Tiempo:0.002709972      / Tamanio Vectores:524288      / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / / V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
[FranciscoRodriguezJLmenez] D1estudiante20 @ ~/bp0/ejer7
16:04:32 $ cat SumaVectoresC_vlocales.7122
/var/spool/pbs/non_priv/jobs/7122.atcgrid.SC: línea 23: 9919 Violación de segmento ('core' generado) /home/D1estudiante20/bp0/ejer7/SumaVectoresC $N
/var/spool/pbs/non_priv/jobs/7122.atcgrid.SC: línea 23: 9921 Violación de segmento ('core' generado) /home/D1estudiante20/bp0/ejer7/SumaVectoresC $N
/var/spool/pbs/non_priv/jobs/7122.atcgrid.SC: línea 23: 9924 Violación de segmento ('core' generado) /home/D1estudiante20/bp0/ejer7/SumaVectoresC $N
/var/spool/pbs/non_priv/jobs/7122.atcgrid.SC: línea 23: 9927 Violación de segmento ('core' generado) /home/D1estudiante20/bp0/ejer7/SumaVectoresC $N
/var/spool/pbs/non_priv/jobs/7122.atcgrid.SC: línea 23: 9930 Violación de segmento ('core' generado) /home/D1estudiante20/bp0/ejer7/SumaVectoresC $N
/var/spool/pbs/non_priv/jobs/7122.atcgrid.SC: línea 23: 9932 Violación de segmento ('core' generado) /home/D1estudiante20/bp0/ejer7/SumaVectoresC $N
/var/spool/pbs/non_priv/jobs/7122.atcgrid.SC: línea 23: 9934 Violación de segmento ('core' generado) /home/D1estudiante20/bp0/ejer7/SumaVectoresC $N
```

resultados en atcgrid

El error obtenido **Violación del segmento('core' generado)** se debe al intentar acceder a una posición de memoria el cual no esta permitido el acceso, esto se debe al **superar el tamaño de la pila**, en el caso del PC es de 2MB, ya que con el máximo tamaño que ejecuta es **262144 elementos * 8B** (ya que el vector es de tipo double) = **2MB**.

En cambio en atcgrid el tamaño de pila es de **2MB**, ya que con el máximo tamaño que ejecuta es **524288 elementos * 8B = 4MB**.

8. Generar los ejecutables del código fuente C para vectores globales y para dinámicos. Ejecutar los dos códigos en un nodo de cómputo de atcgrid usando un script como el del Listado 2 para el mismo rango de tamaños utilizado en el ejercicio anterior. Hay que poner en el script el nombre de los ficheros ejecutables generados en este ejercicio. Ejecutar también los códigos en el PC. ¿Se obtiene error usando vectores globales o dinámicos? ¿A qué cree que es debido? (Incorporar volcados de pantalla como se indica en las normas de prácticas)

RESPUESTA:

```
cazz@cazz-ubuntu: ~/Documentos/3º CURSO/2 Cuatrimestre/AC/Prácticas _grupo reducido/_6. PRACTICAS/bp0/ejer8
Archivo Editar Ver Buscar Terminal Ayuda
→ ejer8 ./sumaVectores_pc.sh SumaVectoresC_global
Tamaño Vectores:65536 (4 B) / 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) /
Tamaño Vectores:131072 (4 B) / 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) /
Tamaño Vectores:262144 (4 B) / 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) /
Tamaño Vectores:524288 (4 B) / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
Tamaño Vectores:1048576 (4 B) / 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) /
Tamaño Vectores:2097152 (4 B) / 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) /
Tamaño Vectores:4194304 (4 B) / 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) /
Tamaño Vectores:8388608 (4 B) / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000) /
Tamaño Vectores:16777216 (4 B) / 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) /
Tamaño Vectores:33554432 (4 B) / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](33554432.200000+33554432.200000=6710886.400000) / V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
Tamaño Vectores:67108864 (4 B) / Tamaño Vectores:67108864 / V1[0]+V2[0]=V3[0](33554432.200000+33554432.200000=6710886.400000) / V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
→ ejer8 |
```

SumaVectoresC_global en PC

```
cazz@cazz-ubuntu: ~/Documentos/3º CURSO/2 Cuatrimestre/AC/Prácticas _grupo reducido/_6. PRACTICAS/bp0/ejer8
Archivo Editar Ver Buscar Terminal Ayuda
→ ejer8 ./sumaVectores_pc.sh SumaVectoresC_dinamica
Tamaño Vectores:65536 (4 B) / 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) /
Tamaño Vectores:131072 (4 B) / 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) /
Tamaño Vectores:262144 (4 B) / 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) /
Tamaño Vectores:524288 (4 B) / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
Tamaño Vectores:1048576 (4 B) / 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) /
Tamaño Vectores:2097152 (4 B) / 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) /
Tamaño Vectores:4194304 (4 B) / 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) /
Tamaño Vectores:8388608 (4 B) / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000) /
Tamaño Vectores:16777216 (4 B) / 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) /
Tamaño Vectores:33554432 (4 B) / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](33554432.200000+33554432.200000=6710886.400000) / V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
Tamaño Vectores:67108864 (4 B) / Tamaño Vectores:67108864 / V1[0]+V2[0]=V3[0](33554432.200000+33554432.200000=6710886.400000) / V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
→ ejer8 |
```

SumaVectoresC_dinamica en PC

```
D1estudiante20@atgrid:~/bp0/ejer8
Archivo Editar Ver Buscar Terminal Ayuda
[FranciscoRodríguezJlmenez] D1estudiante20 @ ~/bp0/ejer8
23:52:38 S qsub /home/D1estudiante20/bp0/ejer8/sumaVectores.sh -q ac
7960.atgrid

[FranciscoRodríguezJlmenez] D1estudiante20 @ ~/bp0/ejer8
23:52:47 S cat SumaVectoresC_dinamica
SumaVectoresC_dinamica.c SumaVectoresC_global.c SumaVectoresC_global.c SumaVectoresC_vlocales.e7960 SumaVectoresC_vlocales.e7960

[FranciscoRodríguezJlmenez] D1estudiante20 @ ~/bp0/ejer8
23:52:47 S cat SumaVectoresC_vlocales
D1.usuario del trabajo: D1estudiante20
Id. del trabajo: 7960.atgrid
Nombre del trabajo especificado por usuario: SumaVectoresC_vlocales
Modo que ejecuta el job: atgrid
Directorio en el que se ha ejecutado qsub: /home/D1estudiante20/bp0/ejer8
Cola: ac
Nodos asignados al trabajo:
atgrid2
Tamaño Vectores:65536 (4 B)
Tamaño Vectores:131072 (4 B) / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
Tamaño Vectores:262144 (4 B) / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tamaño Vectores:4194308 (4 B) / Tamaño Vectores:4194308 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
Tamaño Vectores:524288 (4 B) / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
Tamaño Vectores:1048576 (4 B) / 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) /
Tamaño Vectores:2097152 (4 B) / 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) /
Tamaño Vectores:4194304 (4 B) / 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) /
Tamaño Vectores:8388608 (4 B) / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000) /
Tamaño Vectores:16777216 (4 B) / 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) /
Tamaño Vectores:33554432 (4 B) / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](33554432.200000+33554432.200000=6710886.400000) / V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
Tamaño Vectores:67108864 (4 B) / Tamaño Vectores:67108864 / V1[0]+V2[0]=V3[0](33554432.200000+33554432.200000=6710886.400000) / V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) |

```

SumaVectoresC_global en atcgrid

```
Diestudiante20@atcgrid:~/bp0/ejer8

Archivo Editar Ver Buscar Terminal Ayuda

[FranciscoRodriguezJlmenez] Diestudiante20 @ ~/bp0/ejer8
[FranciscoRodriguezJlmenez] Diestudiante20 @ ~/bp0/ejer8
23:54:49 $ qsub /home/Diestudiante20/bp0/ejer8/SumaVectores.sh -q ac
7961.atcgrid

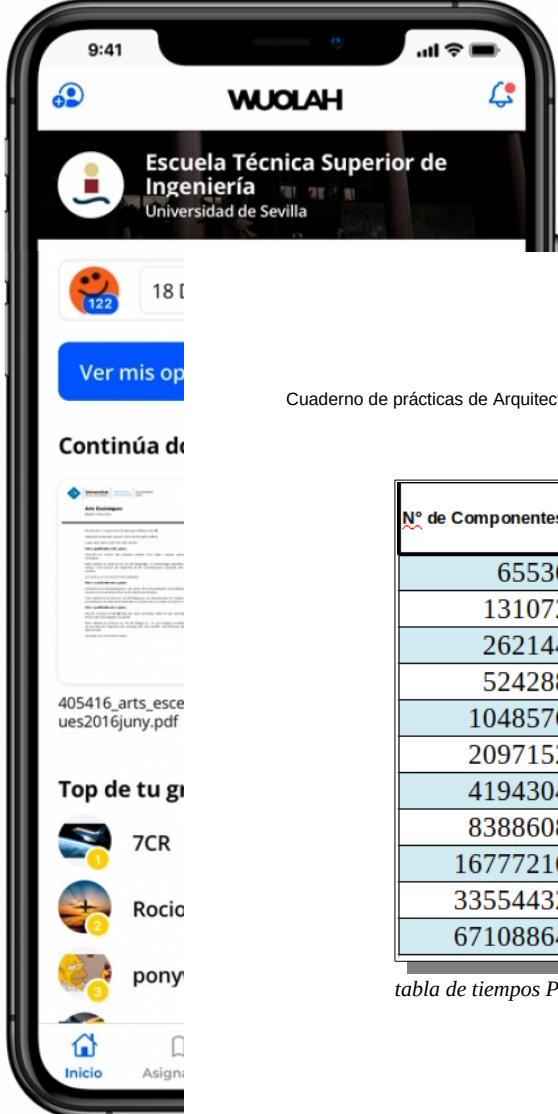
[FranciscoRodriguezJlmenez] Diestudiante20 @ ~/bp0/ejer8
23:54:59 $ cat SsumaVectores_vlocales.07961
Id. usuario del trabajo: Diestudiante20
Id. del trabajo: 7961.atcgrid
Nombre del trabajo especificado por usuario: SumaVectores_vlocales
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/Diestudiante20/bp0/ejer8
Cola en la que se ha ejecutado:
Nodos asignados al trabajo:
atcgrid3
Tamanio Vectores:65536 (4 B)
[Tiempo: 0.008481922] / Tamanio 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) /
Tamanio Vectores:131072 (4 B)
[Tiempo: 0.008944360] / Tamanio 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) /
Tamanio Vectores:262144 (4 B)
[Tiempo: 0.008878977] / Tamanio 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) /
Tamanio Vectores:524288 (4 B)
[Tiempo: 0.002866963] / Tamanio Vectores:524288 / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.000000) / V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.000000) /
Tamanio Vectores:1048576 (4 B)
[Tiempo: 0.005245291] / Tamanio 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) /
Tamanio Vectores:2097152 (4 B)
[Tiempo: 0.005245290] / Tamanio 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) /
Tamanio Vectores:4194304 (4 B)
[Tiempo: 0.01020605] / Tamanio 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) /
Tamanio Vectores:8388608 (4 B)
[Tiempo: 0.033874556] / Tamanio Vectores:8388608 / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000) /
Tamanio Vectores:16777216 (4 B)
[Tiempo: 0.033554431] / Tamanio 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) /
Tamanio Vectores:33554432 (4 B)
[Tiempo: 0.129771518] / Tamanio 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) /
Tamanio Vectores:67108864 (4 B)
[Tiempo: 0.263085457] / Tamanio 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) /
```

SumaVectoresC_dinamica en atcgrid

- En **SumaVectoresC_global**, la longitud del vector no está limitada por el tamaño de la pila del programa, además if ($N > MAX$) $N = MAX$; siendo $MAX = 33554432$, el tamaño máximo.
 - En **SumaVectoresC_dinamica**, se usa memoria dinámica, se reserva el vector con el tamaño N , mediante la función `malloc`, con lo cual no habrá ningún problema por el tamaño de la pila, ademas de que dicha memoria es reutilizable.

9. Rellenar una tabla como la Tabla 1 **en una hoja de cálculo** con los tiempos de ejecución obtenidos en los ejercicios anteriores para el trozo de código que realiza la suma de vectores. Debe haber una tabla para atcgrid y otra para su PC en la hoja de cálculo. En la columna “Bytes de un vector” hay que poner el total de bytes reservado para un vector. Con ayuda de la hoja de cálculo representar **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 (por tanto, los valores de la segunda columna de la tabla, que están en escala logarítmica, deben estar en el eje x). Utilizar escala logarítmica en el eje de ordenadas (eje y). (NOTA: Se recomienda usar en la hoja de cálculo el mismo separador para decimales que usan los códigos. Este separador se puede modificar en la hoja de cálculo.)

(a) Copiar las tablas y la gráfica en el cuaderno de prácticas.



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

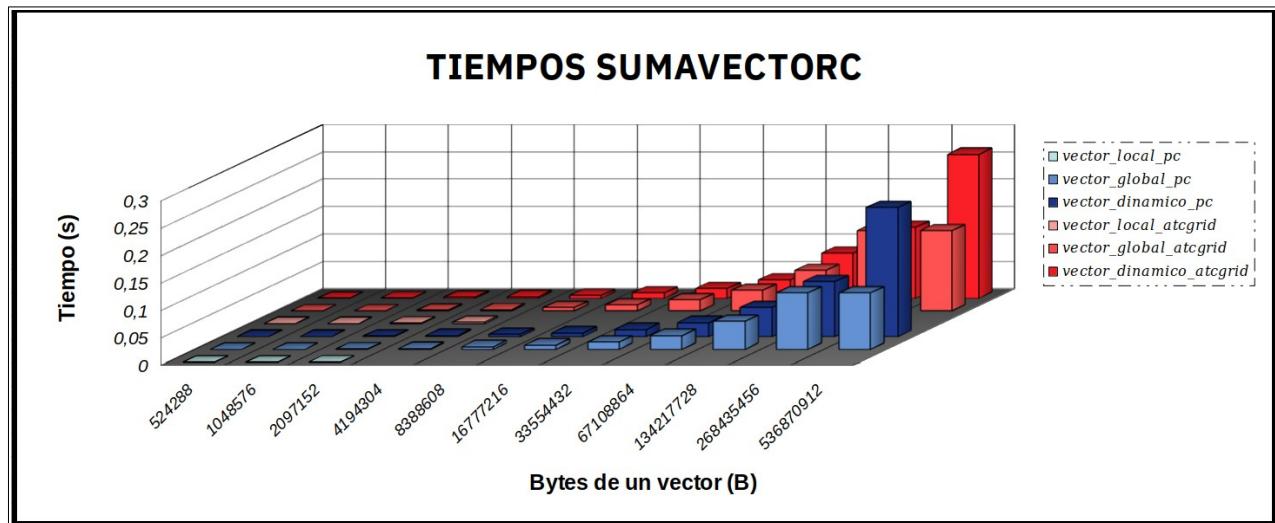
Continúa de

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0,000183837	0,000170789	0,000185145
131072	1048576	0,000359601	0,000331588	0,000358693
262144	2097152	0,00072589	0,000930633	0,001011587
524288	4194304	0	0,001855902	0,002149994
1048576	8388608	0	0,005131903	0,004117546
2097152	16777216	0	0,008050411	0,006858303
4194304	33554432	0	0,013069425	0,012851179
8388608	67108864	0	0,025635885	0,025882674
16777216	134217728	0	0,051069156	0,052940813
33554432	268435456	0	0,102852228	0,10091618
67108864	536870912	0	0,103124847	0,235526693

tabla de tiempos PC

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0,00047258	0,00054342	0,000481922
131072	1048576	0,000942333	0,000495069	0,00094436
262144	2097152	0,001876223	0,001448518	0,001878977
524288	4194304	0,002709972	0,002795975	0,002866963
1048576	8388608	0	0,006111899	0,005452961
2097152	16777216	0	0,010905963	0,009966012
4194304	33554432	0	0,020296107	0,018026005
8388608	67108864	0	0,037895493	0,033874556
16777216	134217728	0	0,074313477	0,082786617
33554432	268435456	0	0,147695139	0,129771518
67108864	536870912	0	0,147136142	0,263008547

tabla de tiempos atcgrid



RESPUESTA:

(b) ¿Hay diferencias en los tiempos de ejecución?

RESPUESTA:

Si el PC es mas rápido que el nodo de computo atcgrid.

Tabla 1 . Copiar la tabla de la hoja de cálculo utilizada

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				

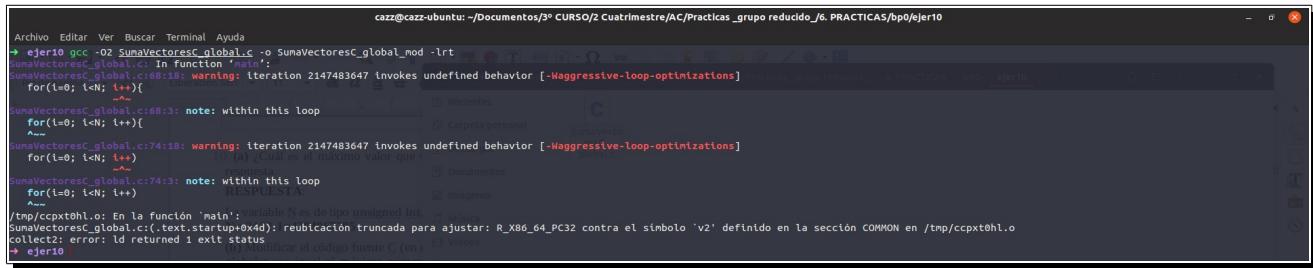
10. (a) ¿Cuál es el máximo valor que se puede almacenar en la variable N teniendo en cuenta su tipo? Razonar respuesta.

RESPUESTA:

La variable **N** es de tipo **unsigned int**, el cual tiene un tamaño máximo de **4B → 4*8 = 32bits**.
 $N = 2^{32} - 1 = 4294967295$.

- (b) Modificar el código fuente C (en el PC) 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 y generar el ejecutable. ¿Qué ocurre? ¿A qué es debido? (Incorporar volcados de pantalla que muestren lo que ocurre)

RESPUESTA:



```
cazz@cazz-ubuntu: ~/Documentos/3º CURSO/2 Cuatrimestre/AC/Prácticas_grupo reducido_/_6. PRACTICAS/bp0/ejer10
Archivo Editar Ver Buscar Terminal Ayuda
ejer10 gcc -O2 SumaVectoresC_global.c -o SumaVectoresC_global_mod -lrt
sumaVectoresC_global.c:168:18: warning: iteration 2147483647 invokes undefined behavior [-Waggressive-loop-optimizations]
  for((i=0 ; i<N ; i++)){
    ~~~~~^~~~~~
sumaVectoresC_global.c:168:3: note: within this loop
  for((i=0 ; i<N ; i++){
  ~~~~~^~~~~~
sumaVectoresC_global.c:174:18: warning: iteration 2147483647 invokes undefined behavior [-Waggressive-loop-optimizations]
  for((i=0 ; i<N ; i++)){
  ~~~~~^~~~~~
sumaVectoresC_global.c:174:3: note: within this loop
  for((i=0 ; i<N ; i++){
  ~~~~~^~~~~~
/tmp/ccpxt0hl.o: En la función `main': variable N es de tipo unsigned int
sumaVectoresC_global.c:(.text.startup+0x4d): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo `v2' definido en la sección COMMON en /tmp/ccpxt0hl.o
collect2: error: ld returned 1 exit status
ejer10
```

Aunque N pueda tener su mayor tamaño **4294967295**, sin que de error, a la hora de crear los vectores,
 $v[4294967295] = 4294967295 * 8B = 31'9...9 GB$ El tamaño máximo de los vectores se desborda, no podemos almacenarlo, falta espacio en memoria.

Entrega del trabajo

Leer lo indicado en las normas de prácticas sobre la entrega del trabajo del bloque práctico en SWAD.

Listado 1 . Código C que suma dos vectores

```
/*
 * SumaVectoresC.c
 * Suma de dos vectores: v3 = v1 + v2
 *
 * Para compilar usar (-lrt: real time library, no todas las versiones de gcc necesitan que se incluya -lrt):
 *     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()

//Sólo puede estar definida una de las tres constantes VECTOR_ (sólo uno de los ...
//tres defines siguientes puede estar descomentado):
#ifndef 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")
#endif#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

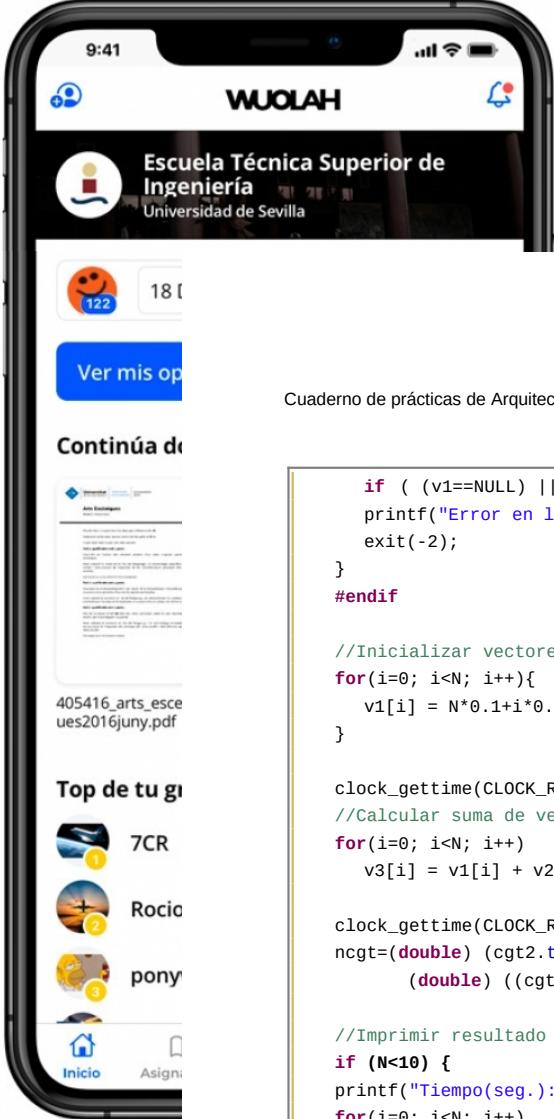
int main(int argc, char** argv){

    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)
#ifndef 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
#ifndef VECTOR_GLOBAL
    if (N>MAX) N=MAX;
#endif
#ifndef 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));

```



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

Continúa d...

```

if ( ( v1==NULL) || (v2==NULL) || (v3==NULL) ){
    printf("Error en la reserva de espacio para los vectores\n");
    exit(-2);
}

#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;
}

//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
if (N<10) {
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) \t /  

        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]);
}

```

Listado 2 . Script para la suma de vectores (SumaVectores.sh). Se supone en el script que el fichero a ejecutar se llama SumaVectorC.

```

#!/bin/bash
#Todos los scripts que se hagan para atcgrid deben incluir lo siguiente:
#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
# FIN del trozo que deben incluir todos los scripts

#para N potencia de 2 desde 2^16 a 2^26
for ((N=65536;N<67108865;N=N*2))
do
    Poner_el_camino_al_ejecutable/SumaVectoresC $N
done
```