

# Arquitectura de Computadores (AC)

## Cuaderno de prácticas.

### Bloque Práctico 0. Entorno de programación

Estudiante (nombre y apellidos):

Grupo de prácticas y profesor de prácticas:

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:

```
[FranciscoJavierBolívarExpósito Diestudiante3@atcgrid:~/bp0/ejer1] 2019-03-05 martes
$echo 'lscpu' | qsub -q ac
7667.atcgrid
[FranciscoJavierBolívarExpósito Diestudiante3@atcgrid:~/bp0/ejer1] 2019-03-05 martes
$cat STDIN.o7667
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                24
On-line CPU(s) list:   0-23
Thread(s) per core:    2
Core(s) per socket:    6
Socket(s):             2
NUMA node(s):          2
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 44
Model name:            Intel(R) Xeon(R) CPU           E5645  @ 2.40GHz
Stepping:               2
CPU MHz:               1600.000
CPU max MHz:           2401,0000
CPU min MHz:           1600,0000
BogoMIPS:              4799.77
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              12288K
NUMA node0 CPU(s):     0-5,12-17
NUMA node1 CPU(s):     6-11,18-23
Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx f
xsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl xtopology non
stop_tsc aperfmperf eagerfpu pni dtes64 monitor ds_cpl vmx smx est tm2 sse3 cx16 xtpr pdcm pcid dca sse4_1 sse4_2 po
pnt lah_f_lm epb ssbd ibrs ibpb stibp tpr_shadow vnmi flexpriority ept vpid dtherm ida arat spec_ctrl intel_stibp flu
sh_lld
[FranciscoJavierBolívarExpósito jakerxd@jakerxd-pc:~] 2019-03-05 martes
$lscpu
Architecture:          x86_64
modo(s) de operación de las CPUs: 32-bit, 64-bit
Orden de los bytes:    Little Endian
Tamaños de las direcciones: 39 bits physical, 48 bits virtual
CPU(s):               8
Lista de la(s) CPU(s) en línea: 0-7
Hilo(s) de procesamiento por núcleo: 2
Núcleo(s) por «socket»: 4
«Socket(s)»:          1
Modo(s) NUMA:          1
ID de fabricante:     GenuineIntel
Familia de CPU:        60
Modelo:               60
Nombre del modelo:     Intel(R) Core(TM) i7-4720HQ CPU @ 2.60GHz
Revisión:              3
CPU MHz:              798.402
CPU MHz máx.:         3600,0000
CPU MHz mín.:         800,0000
BogoMIPS:             5190.63
Virtualización:        VT-x
Caché L1d:            32K
Caché L1i:            32K
Caché L2:             256K
Caché L3:             6144K
CPU(s) del nodo NUMA 0: 0-7
Indicadores:          fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush
dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl
xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl vmx est tm2 sse3 sdbg fma cx16 xtpr pdcm
pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm cpuid_fault epb invp
cid_single_pti ssbd ibrs ibpb stibp tpr_shadow vnmi flexpriority ept vpid ept_ad fsgsbase tsc_adjust bmi1 avx2 smep b
mi2 erms invpcid xsaveopt dtherm ida arat pln pts flush_lld
```

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

**RESPUESTA:** Podemos ver los cores físicos multiplicando el valor de “Núcleo(s) por <<socket>>:” por el número de sockets indicado en <<Socket(s)>>. Los cores lógicos los podemos obtener multiplicando los cores físicos por los “Hilo(s) de procesamiento por núcleo:” o mirando las “CPU(s):”. El PC personal dispone de 4 cores físicos y 8 lógicos. Los nodos de cómputo de atcgrid disponen de

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:**

```
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer2] 2019-03-05 martes
$gcc -fopenmp -O2 HelloOMP.c -o HelloOMP
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer2] 2019-03-05 martes
$./HelloOMP
(0:!!!Hello world!!!)(7:!!!Hello world!!!)(2:!!!Hello world!!!)(5:!!!Hello world!!!)(1:!!!Hello world!!!)(6:!!!Hello
world!!!)(4:!!!Hello world!!!)(3:!!!Hello world!!!)[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Unive
rsidad/AC/Prácticas/bp0/ejer2] 2019-03-05 martes
```

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

**RESPUESTA:** El número de “Hello world” se corresponde con el número de cores lógicos que tiene el pc (8) y que hemos visto en `lscpu`. El programa se ejecuta una vez en cada core lógico.

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:**

```
sftp> put ./HelloOMP
Uploading ./HelloOMP to /home/D1estudiante3/bp0/ejer2/HelloOMP
./HelloOMP 100% 16KB 418.3KB/s 00:00

[FranciscoJavierBolivarExpósito D1estudiante3@atcgrid:~/bp0/ejer3] 2019-03-06 miércoles
$echo '../ejer2/HelloOMP' | qsub -q ac
8344.atcgrid
```

```
[FranciscoJavierBolivarExpósito D1estudiante3@atcgrid:~/bp0/ejer3] 2019-03-06 miércoles
$echo '../bp0/ejer2/HelloOMP' | qsub -q ac
8351.atcgrid
[FranciscoJavierBolivarExpósito D1estudiante3@atcgrid:~/bp0/ejer3] 2019-03-06 miércoles
$cat STDIN.o8351
(9:!!!Hello world!!!)(18:!!!Hello world!!!)(7:!!!Hello world!!!)(10:!!!Hello world!!!)(17:!!!Hello world!!!)(13:!!!He
llo world!!!)(5:!!!Hello world!!!)(19:!!!Hello world!!!)(1:!!!Hello world!!!)(12:!!!Hello world!!!)(22:!!!Hello world
!!!)(16:!!!Hello world!!!)(11:!!!Hello world!!!)(23:!!!Hello world!!!)(15:!!!Hello world!!!)(14:!!!Hello world!!!)(8:
!!!Hello world!!!)(0:!!!Hello world!!!)(2:!!!Hello world!!!)(3:!!!Hello world!!!)(20:!!!Hello world!!!)(21:!!!Hello w
orld!!!)(4:!!!Hello world!!!)(6:!!!Hello world!!!)[FranciscoJavierBolivarExpósito D1estudiante3@atcgrid:~/bp0/ejer3]
2019-03-06 miércoles
```

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

**RESPUESTA:** El número de “Hello world” se corresponde con el número de cores lógicos que tiene el nodo de cómputo de `atcgrid` (24) y que hemos visto en `lscpu`. El programa se ejecuta una vez en cada core lógico.

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      #pragma omp parallel
6      {
7          printf("(%d:!!!Hello)", omp_get_thread_num());
8          printf("(%d: world!!!)", omp_get_thread_num());
9      }
10     return(0);
11 }

```

```

[FranciscoJavierBolívarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer4] 2019-03-06 miércoles
$gcc -fopenmp -O2 HelloOMP2.c -o HelloOMP2

```

```

sftp> put HelloOMP2
Uploading HelloOMP2 to /home/D1estudiante3/bp0/ejer4/HelloOMP2
HelloOMP2                                     100% 16KB 125.4KB/s 00:00

```

```

[FranciscoJavierBolívarExpósito D1estudiante3@atcgrid:~/bp0/ejer4] 2019-03-07 jueves
$qsub script_helloomp.sh
9200.atcgrid
[FranciscoJavierBolívarExpósito D1estudiante3@atcgrid:~/bp0/ejer4] 2019-03-07 jueves
$ls
HelloOMP2 helloomp.e9200 helloomp.o9200 script_helloomp.sh
[FranciscoJavierBolívarExpósito D1estudiante3@atcgrid:~/bp0/ejer4] 2019-03-07 jueves
$cat helloomp.o9200
Id. usuario del trabajo: D1estudiante3
Id. del trabajo: 9200.atcgrid
Nombre del trabajo especificado por usuario: helloomp
Directorio en el que se ha ejecutado qsub: /home/D1estudiante3/bp0/ejer4
Directorio de trabajo: /home/D1estudiante3
Cola: ac
Nodo que ejecuta qsub: atcgrid
Nodos asignados al trabajo:
atcgrid1
Nº de threads inicial: 12
Directorio de trabajo: /home/D1estudiante3/bp0/ejer4

1.Para 12 threads:
(2:!!!Hello)(2: world!!!)(5:!!!Hello)(5: world!!!)(0:!!!Hello)(8:!!!Hello)(8: world!!!)(0: world!!!)(10:!!!Hello)(10:
world!!!)(11:!!!Hello)(11: world!!!)(1:!!!Hello)(1: world!!!)(3:!!!Hello)(3: world!!!)(4:!!!Hello)(4: world!!!)(7:!!!
Hello)(7: world!!!)(6:!!!Hello)(6: world!!!)(9:!!!Hello)(9: world!!!)
1.Para 6 threads:
(0:!!!Hello)(0: world!!!)(3:!!!Hello)(3: world!!!)(5:!!!Hello)(5: world!!!)(2:!!!Hello)(2: world!!!)(1:!!!Hello)(1: w
orld!!!)(4:!!!Hello)(4: world!!!)
1.Para 3 threads:
(1:!!!Hello)(1: world!!!)(0:!!!Hello)(0: world!!!)(2:!!!Hello)(2: world!!!)
1.Para 1 threads:
(0:!!!Hello)(0: world!!!)[FranciscoJavierBolívarExpósito D1estudiante3@atcgrid:~/bp0/ejer4] 2019-03-07 jueves

```

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

**RESPUESTA:** Se ejecuta en el nodo de cómputo atcgrid1. Se ha obtenido esta información al imprimir el contenido del archivo que se especifica en la variable de entorno de PBS \$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:** Si se ejecuta el script usando ./HelloOMP2 y el ejecutable sigue estando en la carpeta del ejercicio 4 no se ejecuta HelloOMP2, al ver la salida de error observamos que no encuentra el ejecutable.

```

[FranciscoJavierBolívarExpósito D1estudiante3@atcgrid:~/bp0/ejer4] 2019-03-07 jueves
$cat helloomp.e9203
/var/spool/pbs/mom_priv/jobs/9203.atcgrid.SC: línea 30: ./HelloOMP2: No existe el fichero o el directorio
/var/spool/pbs/mom_priv/jobs/9203.atcgrid.SC: línea 30: ./HelloOMP2: No existe el fichero o el directorio
/var/spool/pbs/mom_priv/jobs/9203.atcgrid.SC: línea 30: ./HelloOMP2: No existe el fichero o el directorio

```

Esto es porque al mandar un trabajo a la cola el directorio “.” se corresponde con el directorio “~” del estudiante. Al poner el ejecutable en este directorio el programa se ejecuta correctamente.

```
[FranciscoJavierBolivarExpósito D1estudiante3@atcgrid:~/bp0/ejer4] 2019-03-07 jueves
$cat helloomp.o9203
Id. usuario del trabajo: D1estudiante3
Id. del trabajo: 9203.atcgrid
Nombre del trabajo especificado por usuario: helloomp
Directorio en el que se ha ejecutado qsub: /home/D1estudiante3/bp0/ejer4
Directorio de trabajo: /home/D1estudiante3
Cola: ac
Nodo que ejecuta qsub: atcgrid
Nodos asignados al trabajo:
atcgrid1
Nº de threads inicial: 12
Directorio de trabajo: /home/D1estudiante3/bp0/ejer4

1.Para 12 threads:

1.Para 6 threads:

1.Para 3 threads:

1.Para 1 threads:
```

Figura 1: Salida de la ejecución sin HelloOMP2 en home

```
[FranciscoJavierBolivarExpósito D1estudiante3@atcgrid:~] 2019-03-07 jueves
$cp ./bp0/ejer4/HelloOMP2 ./
[FranciscoJavierBolivarExpósito D1estudiante3@atcgrid:~] 2019-03-07 jueves
$ls
bp0 HelloOMP2
[FranciscoJavierBolivarExpósito D1estudiante3@atcgrid:~] 2019-03-07 jueves
$cd ./bp0/ejer4
[FranciscoJavierBolivarExpósito D1estudiante3@atcgrid:~/bp0/ejer4] 2019-03-07 jueves
$ls
HelloOMP2 helloomp.e9200 helloomp.e9203 helloomp.o9200 helloomp.o9203 script_helloomp.sh
[FranciscoJavierBolivarExpósito D1estudiante3@atcgrid:~/bp0/ejer4] 2019-03-07 jueves
$qsub script_helloomp.sh
9261.atcgrid
[FranciscoJavierBolivarExpósito D1estudiante3@atcgrid:~/bp0/ejer4] 2019-03-07 jueves
$cat
HelloOMP2          helloomp.e9203      helloomp.o9200      helloomp.o9261
helloomp.e9200      helloomp.e9261      helloomp.o9203      script_helloomp.sh
[FranciscoJavierBolivarExpósito D1estudiante3@atcgrid:~/bp0/ejer4] 2019-03-07 jueves
$cat helloomp.o9261
Id. usuario del trabajo: D1estudiante3
Id. del trabajo: 9261.atcgrid
Nombre del trabajo especificado por usuario: helloomp
Directorio en el que se ha ejecutado qsub: /home/D1estudiante3/bp0/ejer4
Directorio de trabajo: /home/D1estudiante3
Cola: ac
Nodo que ejecuta qsub: atcgrid
Nodos asignados al trabajo:
atcgrid1
Nº de threads inicial: 12
Directorio de trabajo: /home/D1estudiante3/bp0/ejer4

1.Para 12 threads:
(0:!!!Hello)(0: world!!!)(1:!!!Hello)(1: world!!!)(6:!!!Hello)(6: world!!!)(8:!!!Hello)(8: world!!!)(2:!!!Hello)(2: w
orld!!!)(5:!!!Hello)(5: world!!!)(7:!!!Hello)(7: world!!!)(11:!!!Hello)(11: world!!!)(3:!!!Hello)(3: world!!!)(10:!!!
Hello)(10: world!!!)(9:!!!Hello)(9: world!!!)(4:!!!Hello)(4: world!!!)
1.Para 6 threads:
(0:!!!Hello)(0: world!!!)(1:!!!Hello)(1: world!!!)(2:!!!Hello)(2: world!!!)(5:!!!Hello)(5: world!!!)(3:!!!Hello)(3: w
orld!!!)(4:!!!Hello)(4: world!!!)
1.Para 3 threads:
(1:!!!Hello)(1: world!!!)(0:!!!Hello)(0: world!!!)(2:!!!Hello)(2: world!!!)
1.Para 1 threads:
(0:!!!Hello)(0: world!!!)[FranciscoJavierBolivarExpósito D1estudiante3@atcgrid:~/bp0/ejer4] 2019-03-07 jueves
```

Figura 2: Salida de la ejecución con HelloOMP2 en home

## Parte II. Resto de ejercicios

5. 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:

```
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer5] 2019-03-07 jueves
$gcc -O2 SumaVectoresC.c -o SumaVectores -lrt
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer5] 2019-03-07 jueves
$./SumaVectores 6
Tamaño Vectores:6 (4 B)
Tiempo:0.000000446 / Tamaño Vectores:6
/ V1[0]+V2[0]=V3[0](0.600000+0.600000=1.200000) /
/ V1[1]+V2[1]=V3[1](0.700000+0.500000=1.200000) /
/ V1[2]+V2[2]=V3[2](0.800000+0.400000=1.200000) /
/ V1[3]+V2[3]=V3[3](0.900000+0.300000=1.200000) /
/ V1[4]+V2[4]=V3[4](1.000000+0.200000=1.200000) /
/ V1[5]+V2[5]=V3[5](1.100000+0.100000=1.200000) /
```



6. 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:** El tiempo que ha tardado en ejecutarse la suma de vectores en segundos.

(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:** Devuelve la información del tiempo en una estructura “timespec” compuesta por dos variables, “`tv_sec`” de tipo `time_t` que guarda los segundos y “`tv_nsec`” de tipo `long` que guarda los nanosegundos.

(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:** Devuelve el tiempo indicado por un reloj disponible en el sistema. Para el `CLOCK_REALTIME` que es el usado en la suma de vectores es los segundos y nanosegundos que han pasado desde el “Epoch”(01/01/1970 a las 00:00:00).

7. 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:** Se obtiene un error para el tamaño 524288 en el PC de violación de segmento debido a que se supera el tamaño de la pila. En el nodo de cómputo se obtiene para el tamaño 1048576.

```
[FranciscoJavierBolívarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer5] 2019-03-08 viernes
$./SumaVectores 65536
Tamaño Vectores:65536 (4 B)
Tiempo:0.000761444 / 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) /
[FranciscoJavierBolívarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer5] 2019-03-08 viernes
$./SumaVectores 131072
Tamaño Vectores:131072 (4 B)
Tiempo:0.001549448 / 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) /
[FranciscoJavierBolívarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer5] 2019-03-08 viernes
$./SumaVectores 262144
Tamaño Vectores:262144 (4 B)
Tiempo:0.003302471 / 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) /
[FranciscoJavierBolívarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer5] 2019-03-08 viernes
$./SumaVectores 524288
Tamaño Vectores:524288 (4 B)
Violación de segmento ('core' generado)
```

```
[FranciscoJavierBolívarExpósito D1estudiante3@atcgrid:~/bp0/ejer7] 2019-03-08 viernes
$cat sumavectores.o9796
Id. usuario del trabajo: D1estudiante3
Id. del trabajo: 9796.atcgrid
Nombre del trabajo especificado por usuario: sumavectores
Directorio en el que se ha ejecutado qsub: /home/D1estudiante3/bp0/ejer7
Directorio de trabajo: /home/D1estudiante3
Cola: ac
Nodo que ejecuta qsub: atcgrid
Nodos asignados al trabajo:
atcgrid1
Tamaño Vectores:65536 (4 B)
Tiempo:0.000468649 / 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)
Tiempo:0.000949366 / 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)
Tiempo:0.001903477 / 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)
Tiempo:0.002880275 / 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) /
```

```
[FranciscoJavierBolivarExpósito D1estudiante3@atcgrid:~/bp0/ejer7] 2019-03-08 viernes
$cat sumavectores.e9796
/var/spool/pbs/mom_priv/jobs/9796.atcgrid.SC: línea 23: 20925 Violación de segmento ('core' generado) ./bp0/ejer7/Su
maVectores $N
/var/spool/pbs/mom_priv/jobs/9796.atcgrid.SC: línea 23: 20927 Violación de segmento ('core' generado) ./bp0/ejer7/Su
maVectores $N
/var/spool/pbs/mom_priv/jobs/9796.atcgrid.SC: línea 23: 20930 Violación de segmento ('core' generado) ./bp0/ejer7/Su
maVectores $N
/var/spool/pbs/mom_priv/jobs/9796.atcgrid.SC: línea 23: 20933 Violación de segmento ('core' generado) ./bp0/ejer7/Su
maVectores $N
/var/spool/pbs/mom_priv/jobs/9796.atcgrid.SC: línea 23: 20936 Violación de segmento ('core' generado) ./bp0/ejer7/Su
maVectores $N
/var/spool/pbs/mom_priv/jobs/9796.atcgrid.SC: línea 23: 20938 Violación de segmento ('core' generado) ./bp0/ejer7/Su
maVectores $N
/var/spool/pbs/mom_priv/jobs/9796.atcgrid.SC: línea 23: 20940 Violación de segmento ('core' generado) ./bp0/ejer7/Su
maVectores $N
```

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:** Las variables globales se guardan en la sección de datos del programa y no dependen del tamaño de la pila por lo que no hay error al reservar la memoria necesaria. Para las variables dinámicas se va reservando más memoria conforme es necesario y hay que liberarla cuando dejen de usarse.

```
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$gcc -O2 SumaVectoresG.c -o SumaVectoresGlobales -lrt
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$gcc -O2 SumaVectoresD.c -o SumaVectoresDinamicos -lrt
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$./SumaVectoresGlobales 65536
Tamaño Vectores:65536 (4 B)
Tiempo:0.000688762 / 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) /
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$./SumaVectoresGlobales 131072
Tamaño Vectores:131072 (4 B)
Tiempo:0.001532213 / 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) /
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$./SumaVectoresGlobales 262144
Tamaño Vectores:262144 (4 B)
Tiempo:0.002180451 / 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) /
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$./SumaVectoresGlobales 524288
Tamaño Vectores:524288 (4 B)
Tiempo:0.002350207 / 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) /
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$./SumaVectoresGlobales 1048576
Tamaño Vectores:1048576 (4 B)
Tiempo:0.005199455 / 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) /
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$./SumaVectoresGlobales 2097152
Tamaño Vectores:2097152 (4 B)
Tiempo:0.010448770 / 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) /
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$./SumaVectoresGlobales 4194304
Tamaño Vectores:4194304 (4 B)
Tiempo:0.013944915 / 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) /
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$./SumaVectoresGlobales 8388608
Tamaño Vectores:8388608 (4 B)
Tiempo:0.027899651 / 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) /
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$./SumaVectoresGlobales 16777216
Tamaño Vectores:16777216 (4 B)
Tiempo:0.055561490 / 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) /
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$./SumaVectoresGlobales 33554432
Tamaño Vectores:33554432 (4 B)
Tiempo:0.129728112 / 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) /
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$./SumaVectoresGlobales 67108864
Tamaño Vectores:67108864 (4 B)
Tiempo:0.109372210 / 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) /
```

```
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$ ./SumaVectoresDinamicos 65536
Tamaño Vectores:65536 (4 B)
Tiempo:0.000776315 / 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) /
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$ ./SumaVectoresDinamicos 131072
Tamaño Vectores:131072 (4 B)
Tiempo:0.001609560 / 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) /
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$ ./SumaVectoresDinamicos 262144
Tamaño Vectores:262144 (4 B)
Tiempo:0.002978173 / 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) /
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$ ./SumaVectoresDinamicos 524288
Tamaño Vectores:524288 (4 B)
Tiempo:0.004486691 / 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) /
```

```
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$ ./SumaVectoresDinamicos 1048576
Tamaño Vectores:1048576 (4 B)
Tiempo:0.003611488 / 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) /
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$ ./SumaVectoresDinamicos 2097152
Tamaño Vectores:2097152 (4 B)
Tiempo:0.009567235 / 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) /
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$ ./SumaVectoresDinamicos 4194304
Tamaño Vectores:4194304 (4 B)
Tiempo:0.014165786 / 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) /
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$ ./SumaVectoresDinamicos 8388608
Tamaño Vectores:8388608 (4 B)
Tiempo:0.027554964 / 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) /
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$ ./SumaVectoresDinamicos 16777216
Tamaño Vectores:16777216 (4 B)
Tiempo:0.054802510 / 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) /
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$ ./SumaVectoresDinamicos 33554432
Tamaño Vectores:33554432 (4 B)
Tiempo:0.110555113 / 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) /
[FranciscoJavierBolivarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer8] 2019-03-08 viernes
$ ./SumaVectoresDinamicos 67108864
Tamaño Vectores:67108864 (4 B)
Tiempo:0.218270622 / 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) /
```



```

[FranciscoJavierBolívarExpósito D1estudiante3@atcgrid:~/bp0/ejer8] 2019-03-08 viernes
$cat sumavectoresglobales.e9899
[FranciscoJavierBolívarExpósito D1estudiante3@atcgrid:~/bp0/ejer8] 2019-03-08 viernes
$cat sumavectoresdinamicos.o9898
Id. usuario del trabajo: D1estudiante3
Id. del trabajo: 9898.atcgrid
Nombre del trabajo especificado por usuario: sumavectoresdinamicos
Directorio en el que se ha ejecutado qsub: /home/D1estudiante3/bp0/ejer8
Directorio de trabajo: /home/D1estudiante3
Cola: ac
Nodo que ejecuta qsub: atcgrid
Nodos asignados al trabajo:
atcgrid1
Tamaño Vectores:65536 (4 B)
Tiempo:0.000481928 / 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)
Tiempo:0.000944861 / 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)
Tiempo:0.001888342 / 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)
Tiempo:0.002892058 / 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)
Tiempo:0.005312990 / 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)
Tiempo:0.009617622 / 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)
Tiempo:0.017726926 / 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)
Tiempo:0.033614455 / 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)
Tiempo:0.066524124 / 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)
Tiempo:0.132181935 / 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) /
Tamaño Vectores:67108864 (4 B)
Tiempo:0.260502143 / 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) /
[FranciscoJavierBolívarExpósito D1estudiante3@atcgrid:~/bp0/ejer8] 2019-03-08 viernes
$cat sumavectoresdinamicos.e9898
[FranciscoJavierBolívarExpósito D1estudiante3@atcgrid:~/bp0/ejer8] 2019-03-08 viernes

```



```

[FranciscoJavierBolivarExpósito D1estudiante3@atcgrid:~/bp0/ejer8] 2019-03-08 viernes
$qsub SumaVectoresDinamicos.sh
9898.atcgrid
[FranciscoJavierBolivarExpósito D1estudiante3@atcgrid:~/bp0/ejer8] 2019-03-08 viernes
$qsub SumaVectoresGlobales.sh
9899.atcgrid
[FranciscoJavierBolivarExpósito D1estudiante3@atcgrid:~/bp0/ejer8] 2019-03-08 viernes
$cat sumavectoresglobales.o9899
Id. usuario del trabajo: D1estudiante3
Id. del trabajo: 9899.atcgrid
Nombre del trabajo especificado por usuario: sumavectoresglobales
Directorio en el que se ha ejecutado qsub: /home/D1estudiante3/bp0/ejer8
Directorio de trabajo: /home/D1estudiante3
Cola: ac
Nodo que ejecuta qsub: atcgrid
Nodos asignados al trabajo:
atcgrid2
Tamaño Vectores:65536 (4 B)
Tiempo:0.000953585 / 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)
Tiempo:0.000637335 / 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)
Tiempo:0.001281900 / 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)
Tiempo:0.002746555 / 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)
Tiempo:0.005644924 / 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)
Tiempo:0.010625432 / 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)
Tiempo:0.019990123 / 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)
Tiempo:0.037559602 / 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)
Tiempo:0.075153011 / 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)
Tiempo:0.147553085 / 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) /
Tamaño Vectores:67108864 (4 B)
Tiempo:0.146952828 / 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) /

```

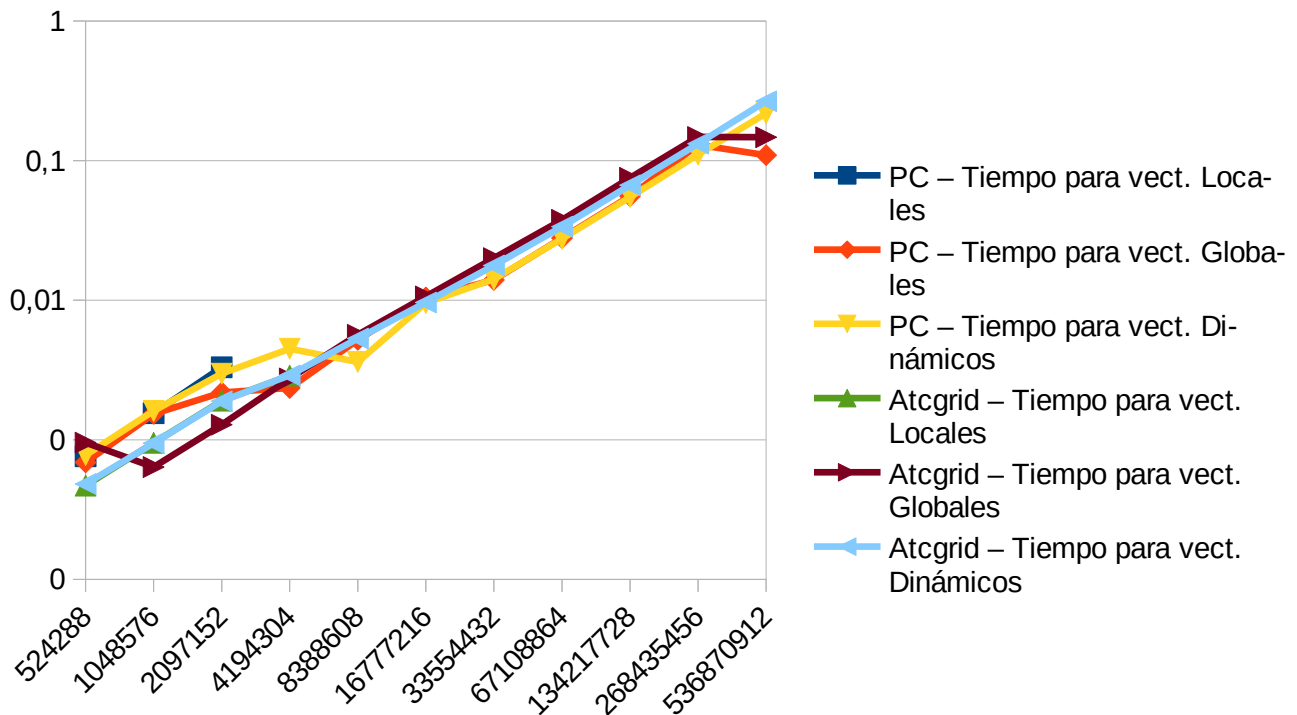
9. Rellenar una tabla como la Tabla 1 **en una hoja de cálculo** con los tiempos de ejecución obtenidos en los ejercicios anteriores para el trozo de código que realiza la suma de vectores. Debe haber una tabla para atcgrid y otra para su PC en la hoja de cálculo. En la columna “Bytes de un vector” hay 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.

**RESPUESTA:**

N.º de Componentes	Bytes de un vector	PC – Tiempo para vect.	LocaPC – Tiempo para vect.	GlobaPC – Tiempo para vect.	Dinám
65536	524288	0,000761444	0,000688762	0,000776315	
131072	1048576	0,001549448	0,001532213	0,00160956	
262144	2097152	0,003302471	0,002180451	0,002978173	
524288	4194304		0,002350207	0,004486691	
1048576	8388608		0,005199455	0,003611488	
2097152	16777216		0,01044877	0,009567235	
4194304	33554432		0,013944915	0,014165786	
8388608	67108864		0,027899651	0,027554964	
16777216	134217728		0,05556149	0,05480251	
33554432	268435456		0,129728112	0,1105551113	
67108864	536870912		0,10937221	0,218270622	

N.º de Componentes	Bytes de un vector	Atcgrid – Tiempo para vect.	Atcgrid – Tiempo para vect.	GAtcgrid – Tiempo para vect.	Dii
65536	524288	0,000468649	0,000953585	0,000481928	
131072	1048576	0,000949366	0,000637335	0,000944861	
262144	2097152	0,001903477	0,0012819	0,001888342	
524288	4194304	0,002880275	0,002746555	0,002892058	
1048576	8388608		0,005644924	0,00531299	
2097152	16777216		0,010625432	0,009617622	
4194304	33554432		0,019990123	0,017726926	
8388608	67108864		0,037559602	0,033614455	
16777216	134217728		0,075153011	0,066524124	
33554432	268435456		0,147553085	0,132181935	
67108864	536870912		0,146952828	0,2660502143	



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

**RESPUESTA:** Para tamaños más pequeños los vectores globales han tardado menos, pero para los más grandes la diferencia es pequeña.

**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:** Es un unsigned int (4 Bytes) por lo que el valor máximo con estos 4B y sin valores negativos sería  $2^{32}-1 = 4294967295$ .

(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:** El tamaño para direccionar todas las componentes del vector es demasiado grande y no cabe en el puntero.

```
[FranciscoJavierBolívarExpósito jakerxd@jakerxd-pc:~/Apuntes_Universidad/AC/Prácticas/bp0/ejer10] 2019-03-08 viernes
$gcc -O2 SumaVectoresG.c -o SumaVectoresGlobales -lrt
/bin/ld: /tmp/ccn7gGhs.o: en la función 'main':
SumaVectoresG.c:(.text.startup+0x83): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo 'v2' definido
o en la sección COMMON en /tmp/ccn7gGhs.o
/bin/ld: SumaVectoresG.c:(.text.startup+0x83): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo 'v3'
definido en la sección COMMON en /tmp/ccn7gGhs.o
collect2: error: ld devolvió el estado de salida 1
```

## 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):
//#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
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) / /
        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]);

#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 .** 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
```