

Arquitectura de Computadores (AC)

Cuaderno de prácticas.

Bloque Práctico 0. Entorno de programación

Estudiante (nombre y apellidos): Antonio David Villegas Yeguas

Grupo de prácticas y profesor de prácticas: B2 – Christian Morillas

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: Maquina local:

atcgrid:

```
antonio@antonio:~$ lscpu
Architectura: x86_64
modo(s) de operación de las CPUs: 32-bit, 64-bit
Orden de los bytes: Little Endian
Tamaño 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: 6
Modelo: 142
Nombre del modelo: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz
Revisión: 10
CPU MHz: 780.020
CPU MHz máx.: 3400.0000
CPU MHz mín.: 400.0000
BogoMIPS: 3601.00
Virtualización: VT-x
Cache L1d: 32K
Cache L1i: 32K
Cache L2: 256K
Cache 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_l1i cache: 32K
rfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf tsc_known_freq pni pclmulqdq dtes64_l2 cache: 256K
monitor ds_cpl vmx est tm2 sse3 sdbg fma cx16 xtpr pdcm pcid sse4.1 sse4.2 x2apic movbe popcnt tsc_deadl3 cache: 12288K
ne_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch cpuid_fault epb invpcid_single pti lbrs tpr_shadow vnmi flexpriority ept vpid ept_ad fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid mpx rdseed adx snap clflushopt intel_pt xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp hwp_tiff hwp_act_window hwp_epp
[AntonioDavidVillegasYeguas antonio@antonio:~] 2019-02-21 jueves

B1estudiante25@atcgrid:~/BP0/ejer1$ lscpu
Architectura: 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: 1733.000
CPU max MHz: 2401.0000
CPU min MHz: 1600.0000
BogoMIPS: 4800.24
Virtualización: 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 dt
s 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 aperfmperf eagerfpu pni dtes64 monitor ds_cpl vmx smx est tm2 sse3 cx16
xtpr pdcm pcid dca sse4.1 sse4.2 popcnt lahf_lm epb ssbd lbrs lbpv stibp tpr_shadow vnmi flexpriority ep
t vpid dtherm ida arat spec_ctrl intel_stibp flush_l1d
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer1] 2019-02-21 jueves
```

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

RESPUESTA:

Mi PC personal tiene 4 CPUs físicas (núcleos por socket) y 8 CPUs lógicas (CPUs totales)

Los nodos de cómputo de atcgrid tienen 6 CPUs físicas por procesador, como tiene 2 procesadores en un nodo, cada nodo tiene 12 CPUs físicas, y como cada CPU física es capaz de ejecutar dos thread, en total, cada nodo tiene 24 CPUs lógicas

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:



```
[AntonioDavidVillegasYeguas antonio@antonio:~/Documentos/Universidad/2do/2do_cuatri/AC/Practicas/BP0/ejer2] 2019-02-21 jueves
$gcc -fopenmp -o HelloOMP HelloOMP.c
[AntonioDavidVillegasYeguas antonio@antonio:~/Documentos/Universidad/2do/2do_cuatri/AC/Practicas/BP0/ejer2] 2019-02-21 jueves
$./HelloOMP
(0:!!!Hello world!!!)
(7:!!!Hello world!!!)
(5:!!!Hello world!!!)
(6:!!!Hello world!!!)
(1:!!!Hello world!!!)
(2:!!!Hello world!!!)
(4:!!!Hello world!!!)
(3:!!!Hello world!!!)
[AntonioDavidVillegasYeguas antonio@antonio:~/Documentos/Universidad/2do/2do_cuatri/AC/Practicas/BP0/ejer2] 2019-02-21 jueves
$
```

(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 programa imprime tantos Hello world como procesadores lógicos tenga la CPU

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:

```
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer3] 2019-02-21 jueves
$echo 'BP0/ejer3/HelloOMP' | qsub -q ac
2204.atcgrid
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer3] 2019-02-21 jueves
$ls -l
total 24
-rwxr-xr-x 1 B1estudiante25 B1estudiante25 16728 feb 21 13:10 HelloOMP
-rw----- 1 B1estudiante25 B1estudiante25 0 feb 21 13:10 STDIN.e2204
-rw----- 1 B1estudiante25 B1estudiante25 566 feb 21 13:10 STDIN.o2204
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer3] 2019-02-21 jueves
$cat STDIN.o2204
(7:!!!Hello world!!!)
(14:!!!Hello world!!!)
(21:!!!Hello world!!!)
(12:!!!Hello world!!!)
(23:!!!Hello world!!!)
(8:!!!Hello world!!!)
(22:!!!Hello world!!!)
(16:!!!Hello world!!!)
(19:!!!Hello world!!!)
(3:!!!Hello world!!!)
(1:!!!Hello world!!!)
(17:!!!Hello world!!!)
(5:!!!Hello world!!!)
(11:!!!Hello world!!!)
(2:!!!Hello world!!!)
(0:!!!Hello world!!!)
(18:!!!Hello world!!!)
(20:!!!Hello world!!!)
(4:!!!Hello world!!!)
(9:!!!Hello world!!!)
(15:!!!Hello world!!!)
(10:!!!Hello world!!!)
(6:!!!Hello world!!!)
(13:!!!Hello world!!!)
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer3] 2019-02-21 jueves
$
```

(b)

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

RESPUESTA: De nuevo, imprime tantos Hello world como CPUs lógicas tiene el nodo

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:

```
#include <stdio.h>
#include <omp.h>

int main (void) {

#pragma omp parallel
    printf("%d:!!!Hello world!!!)\n", omp_get_thread_num());
    printf("%d: world!!!)\n", omp_get_thread_num());

    return (0);
}
```

```
sftp> put HelloOMP2.c
Uploading HelloOMP2.c to /home/B1estudiante25/BP0/ejer4/HelloOMP2.c
HelloOMP2.c          100% 209    53.0KB/s   00:00
sftp> █
```

```
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer4] 2019-03-05 martes
$gcc -fopenmp -o HelloOMP2 HelloOMP2.c
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer4] 2019-03-05 martes
$█
```

```
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer4] 2019-03-05 martes
$qsub script_helloomp.sh -q ac
7199.atcgrid
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer4] 2019-03-05 martes
$ls
HelloOMP2 HelloOMP2.c helloomp.e7199 helloomp.o7199 script_helloomp.sh
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer4] 2019-03-05 martes
$ls -l
total 24
-rwxrwxr-x 1 B1estudiante25 B1estudiante25 8352 mar  5 16:46 HelloOMP2
-rw-r--r-- 1 B1estudiante25 B1estudiante25  209 mar  5 16:43 HelloOMP2.c
-rw----- 1 B1estudiante25 B1estudiante25    0 mar  5 16:48 helloomp.e7199
-rw----- 1 B1estudiante25 B1estudiante25 1049 mar  5 16:48 helloomp.o7199
-rw-r--r-- 1 B1estudiante25 B1estudiante25 1214 mar  5 16:49 script_helloomp.sh
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer4] 2019-03-05 martes
$█
```

```
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer4] 2019-03-05 martes
$cat helloomp.o7199
Id. usuario del trabajo: B1estudiante25
Id. del trabajo: 7199.atcgrid
Nombre del trabajo especificado por usuario: helloomp
Directorio en el que se ha ejecutado qsub: /home/B1estudiante25/BP0/ejer4
Directorio de trabajo: /home/B1estudiante25
Cola: ac
Nodo que ejecuta qsub: atcgrid
Nodos asignados al trabajo:
atcgrid3
Nº de threads inicial: 12
Directorio de trabajo: /home/B1estudiante25/BP0/ejer4

1. Para 12 threads:
(0:!!!Hello world!!!)
(3:!!!Hello world!!!)
(1:!!!Hello world!!!)
(5:!!!Hello world!!!)
(7:!!!Hello world!!!)
(8:!!!Hello world!!!)
(6:!!!Hello world!!!)
(9:!!!Hello world!!!)
(10:!!!Hello world!!!)
(4:!!!Hello world!!!)
(2:!!!Hello world!!!)
(11:!!!Hello world!!!)
(0: world!!!)

1. Para 6 threads:
(4:!!!Hello world!!!)
(3:!!!Hello world!!!)
(0:!!!Hello world!!!)
(2:!!!Hello world!!!)
(5:!!!Hello world!!!)
(1:!!!Hello world!!!)
(0: world!!!)

1. Para 3 threads:
(0:!!!Hello world!!!)
(2:!!!Hello world!!!)
(1:!!!Hello world!!!)
(0: world!!!)

1. Para 1 threads:
(0:!!!Hello world!!!)
(0: world!!!)
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer4] 2019-03-05 martes
$
```

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

RESPUESTA: atcgrid3 – lo he sabido por la variable \$PBS_QUEUE

(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: Ocurre un error, ya que el directorio de trabajo no es el actual, es decir, el binario a ejecutar en el script esta en otro lugar

```
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer4] 2019-03-05 martes
$qsub script_helloomp.sh -q ac
7227.atcgrid
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer4] 2019-03-05 martes
$ls -l
total 28
-rwxrwxr-x 1 B1estudiante25 B1estudiante25 8352 mar  5 16:46 HelloOMP2
-rw-r--r-- 1 B1estudiante25 B1estudiante25 209 mar  5 16:43 HelloOMP2.c
-rw----- 1 B1estudiante25 B1estudiante25 424 mar  5 16:59 helloomp.e7227
-rw----- 1 B1estudiante25 B1estudiante25 481 mar  5 16:59 helloomp.o7227
-rw-r--r-- 1 B1estudiante25 B1estudiante25 1200 mar  5 17:02 script_helloomp.sh
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer4] 2019-03-05 martes
$cat helloomp.e7227
/var/spool/pbs/mom_priv/jobs/7227.atcgrid.SC: línea 30: ./HelloOMP: No existe el fichero o el directorio
/var/spool/pbs/mom_priv/jobs/7227.atcgrid.SC: línea 30: ./HelloOMP: No existe el fichero o el directorio
/var/spool/pbs/mom_priv/jobs/7227.atcgrid.SC: línea 30: ./HelloOMP: No existe el fichero o el directorio
/var/spool/pbs/mom_priv/jobs/7227.atcgrid.SC: línea 30: ./HelloOMP: No existe el fichero o el directorio
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer4] 2019-03-05 martes
$cat helloomp.o7227
Id. usuario del trabajo: B1estudiante25
Id. del trabajo: 7227.atcgrid
Nombre del trabajo especificado por usuario: helloomp
Directorio en el que se ha ejecutado qsub: /home/B1estudiante25/BP0/ejer4
Directorio de trabajo: /home/B1estudiante25
Cola: ac
Nodo que ejecuta qsub: atcgrid
Nodos asignados al trabajo:
atcgrid2
Nº de threads inicial: 12
Directorio de trabajo: /home/B1estudiante25/BP0/ejer4

1.Para 12 threads:

1.Para 6 threads:

1.Para 3 threads:

1.Para 1 threads:
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer4] 2019-03-05 martes
$
```

Parte II. Resto de ejercicios

RESPUESTA:

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:

```
[AntonioDavidVillegasYeguas antonio@antonio:~/Documentos/Universidad/2do/2do_cuatri/AC/Practicas/BP0/ejer5] 2019-03-05 martes
$gcc -O2 SumaVectoresC.c -o SumaVectores -lrt
[AntonioDavidVillegasYeguas antonio@antonio:~/Documentos/Universidad/2do/2do_cuatri/AC/Practicas/BP0/ejer5] 2019-03-05 martes
$./SumaVectores 10
Tama Vector:10 (4 B)
Tiempo:0.000000458 / Tama Vector:10 / V1[0]+V2[0]=V3[0](1.000000+1.000000=2.000000) / / V1[9]+V2[9]=V3[9](1.900000+0.100000=2.000000) /
[AntonioDavidVillegasYeguas antonio@antonio:~/Documentos/Universidad/2do/2do_cuatri/AC/Practicas/BP0/ejer5] 2019-03-05 martes
$./SumaVectores 9
Tama Vector:9 (4 B)
Tiempo:0.000000453 / Tama Vector:9
/ V1[0]+V2[0]=V3[0](0.900000+0.900000=1.800000) /
/ V1[1]+V2[1]=V3[1](1.000000+0.800000=1.800000) /
/ V1[2]+V2[2]=V3[2](1.100000+0.700000=1.800000) /
/ V1[3]+V2[3]=V3[3](1.200000+0.600000=1.800000) /
/ V1[4]+V2[4]=V3[4](1.300000+0.500000=1.800000) /
/ V1[5]+V2[5]=V3[5](1.400000+0.400000=1.800000) /
/ V1[6]+V2[6]=V3[6](1.500000+0.300000=1.800000) /
/ V1[7]+V2[7]=V3[7](1.600000+0.200000=1.800000) /
/ V1[8]+V2[8]=V3[8](1.700000+0.100000=1.800000) /
[AntonioDavidVillegasYeguas antonio@antonio:~/Documentos/Universidad/2do/2do_cuatri/AC/Practicas/BP0/ejer5] 2019-03-05 martes
$
```

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: Como vemos, contiene la diferencia del tiempo entre la finalización de la suma de los vectores y el inicio de la suma:

```
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));
```

- (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: Lo devuelve en la estructura `timespec`, definida en `time.h`

(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: En concreto en `struct timespec.h` (localizado en `/usr/include/bits/types/struct_timespec.h` en mi PC) que como vemos almacena los segundos y los nanosegundos

```
/* NB: Include guard matches what <linux/time.h> uses. */
#ifndef _STRUCT_TIMESPEC
#define _STRUCT_TIMESPEC 1

#include <bits/types.h>

/* POSIX.1b structure for a time value. This is like a 'struct timeval' but
   has nanoseconds instead of microseconds. */
struct timespec
{
    __time_t tv_sec; /* Seconds. */
    __syscall_slong_t tv_nsec; /* Nanoseconds. */
};

#endif
```

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:

```
sftp> pwd
Remote working directory: /home/B1estudiante25/BP0/ejer7
sftp> lpwd
Local working directory: /home/antonio/Documentos/Universidad/2do/2do_cuatric/AC/Practicas/BP0/ejer7
sftp> put
SumaVectoresC SumaVectoresC.c script_sumavectorc.sh

sftp> put *
Uploading SumaVectoresC to /home/B1estudiante25/BP0/ejer7/SumaVectoresC
SumaVectoresC 100% 16KB 278.9KB/s 00:00
Uploading SumaVectoresC.c to /home/B1estudiante25/BP0/ejer7/SumaVectoresC.c
SumaVectoresC.c 100% 3285 554.8KB/s 00:00
Uploading script_sumavectorc.sh to /home/B1estudiante25/BP0/ejer7/script_sumavectorc.sh
script_sumavectorc.sh 100% 810 79.0KB/s 00:00
sftp> ls
SumaVectoresC SumaVectoresC.c script_sumavectorc.sh
sftp>
```

```
[AntonioDavidVillegasYeguas antonio@antonio:~/Documentos/Universidad/2do/2do_cuatric/AC/Practicas/BP0/ejer7] 2019-03-05 martes
$for ((N=65536;N<67108865;N=N*2))
do
./SumaVectoresC $N
done
Tamaño Vectores:65536 (4 B)
Tiempo:0.000940722 / 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:0.001852387 / 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:0.00374233 / 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)
Violación de segmento ('core' generado)
Tamaño Vectores:1048576 (4 B)
Violación de segmento ('core' generado)
Tamaño Vectores:2097152 (4 B)
Violación de segmento ('core' generado)
Tamaño Vectores:4194304 (4 B)
Violación de segmento ('core' generado)
Tamaño Vectores:8388608 (4 B)
Violación de segmento ('core' generado)
Tamaño Vectores:16777216 (4 B)
Violación de segmento ('core' generado)
Tamaño Vectores:33554432 (4 B)
Violación de segmento ('core' generado)
Tamaño Vectores:67108864 (4 B)
Violación de segmento ('core' generado)
[AntonioDavidVillegasYeguas antonio@antonio:~/Documentos/Universidad/2do/2do_cuatric/AC/Practicas/BP0/ejer7] 2019-03-05 martes
```



```

[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer7] 2019-03-05 martes
$ qsub script_sumavectorc.sh -q ac
7707.atcgrid
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer7] 2019-03-05 martes
$ ls -l
total 36
-rwxr-xr-x 1 B1estudiante25 B1estudiante25 791 mar 5 18:59 script_sumavectorc.sh
-rwxr-xr-x 1 B1estudiante25 B1estudiante25 16848 mar 5 18:45 SumaVectoresC
-rw-r--r-- 1 B1estudiante25 B1estudiante25 3285 mar 5 18:45 SumaVectoresC.c
-rw----- 1 B1estudiante25 B1estudiante25 959 mar 5 18:55 SumaVectoresC_vlocales.e7707
-rw----- 1 B1estudiante25 B1estudiante25 1139 mar 5 18:55 SumaVectoresC_vlocales.o7707
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer7] 2019-03-05 martes
$ cat SumaVectoresC_vlocales.e7707
/var/spool/pbs/mom_priv/jobs/7707.atcgrid.SC: línea 22: 17514 Violación de segmento ('core' generado) $P
BS_0_WORKDIR/SumaVectoresC $N
/var/spool/pbs/mom_priv/jobs/7707.atcgrid.SC: línea 22: 17516 Violación de segmento ('core' generado) $P
BS_0_WORKDIR/SumaVectoresC $N
/var/spool/pbs/mom_priv/jobs/7707.atcgrid.SC: línea 22: 17519 Violación de segmento ('core' generado) $P
BS_0_WORKDIR/SumaVectoresC $N
/var/spool/pbs/mom_priv/jobs/7707.atcgrid.SC: línea 22: 17522 Violación de segmento ('core' generado) $P
BS_0_WORKDIR/SumaVectoresC $N
/var/spool/pbs/mom_priv/jobs/7707.atcgrid.SC: línea 22: 17525 Violación de segmento ('core' generado) $P
BS_0_WORKDIR/SumaVectoresC $N
/var/spool/pbs/mom_priv/jobs/7707.atcgrid.SC: línea 22: 17527 Violación de segmento ('core' generado) $P
BS_0_WORKDIR/SumaVectoresC $N
/var/spool/pbs/mom_priv/jobs/7707.atcgrid.SC: línea 22: 17529 Violación de segmento ('core' generado) $P
BS_0_WORKDIR/SumaVectoresC $N
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer7] 2019-03-05 martes
$ cat SumaVectoresC_vlocales.o7707
Id. usuario del trabajo: B1estudiante25
Id. del trabajo: 7707.atcgrid
Nombre del trabajo especificado por usuario: SumaVectoresC_vlocales
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/B1estudiante25/BP0/ejer7
Cola: ac
Nodos asignados al trabajo:
atcgrid2
Tamaño Vectores: 65536 (4 B)
Tiempo: 0.000481997 / 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: 0.000943513 / Tamaño Vectores: 131072 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=262
14.400000) / / V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tamaño Vectores: 262144 (4 B)
Tiempo: 0.001893173 / Tamaño Vectores: 262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=524
28.800000) / / V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
Tamaño Vectores: 524288 (4 B)
Tiempo: 0.002815960 / Tamaño Vectores: 524288 / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104
857.600000) / / V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer7] 2019-03-05 martes
$

```

Este error se debe a que el tamaño de N supera el tamaño de la pila

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: Con vectores globales no se obtiene error, sin embargo, llega un punto en el que limitamos el tamaño del vector, para que no se desborde con el máximo, como vemos en este segmento del código sacado del .c:

```
#ifdef VECTOR_GLOBAL
if (N>MAX) N=MAX;
#endif
```

```
B1estudiante25@atcgrid:~/BP0/ejer8
Archivo Editar Ver Buscar Terminal Ayuda
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer8] 2019-03-12 martes
$head -26 SumaVectoresC.c
/* SumaVectoresC.c
Suma de dos vectores: v3 = v1 + v2

Para compilar usar (-lrt: real time library, es posible que no sea necesario usar -lrt):
gcc -O2 SumaVectores.c -o SumaVectores -lrt
gcc -O2 -S SumaVectores.c -lrt

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()

//Solo puede estar definida una de las tres constantes VECTOR_ (solo 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 ...
// // generar 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)

[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer8] 2019-03-12 martes
$gcc -O2 SumaVectoresC.c -o SumaVectoresC -lrt
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer8] 2019-03-12 martes
$qsub script_sumavectorc.sh -q ac
11003.atcgrid
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer8] 2019-03-12 martes
$ls -l
total 24
-rwxr-xr-x 1 B1estudiante25 B1estudiante25 792 mar 12 16:04 script_sumavectorc.sh
-rwxrwxr-x 1 B1estudiante25 B1estudiante25 8488 mar 12 16:04 SumaVectoresC
-rw-r--r-- 1 B1estudiante25 B1estudiante25 3285 mar 12 16:04 SumaVectoresC.c
-rw----- 1 B1estudiante25 B1estudiante25 0 mar 12 16:05 SumaVectoresC_vglobales.e11003
-rw----- 1 B1estudiante25 B1estudiante25 2735 mar 12 16:05 SumaVectoresC_vglobales.o11003
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer8] 2019-03-12 martes
$cat SumaVectoresC_vglobales.o11003
Id. usuario del trabajo: B1estudiante25
Id. del trabajo: 11003.atcgrid
Nombre del trabajo especificado por usuario: SumaVectoresC_vglobales
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/B1estudiante25/BP0/ejer8
Cola: ac
Nodos asignados al trabajo:
atcgrid3
Tamaño Vectores:65536 (4 B)
Tiempo:0.000563340 / 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:0.000511091 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=262
14.400000) / / V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
```

```

B1estudiante25@atcgrid:~/BP0/ejer8
Archivo Editar Ver Buscar Terminal Ayuda
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer8] 2019-03-12 martes
$ qsub script_sumavectorc.sh -q ac
11003.atcgrid
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer8] 2019-03-12 martes
$ ls -l
total 24
-rwxr-xr-x 1 B1estudiante25 B1estudiante25 792 mar 12 16:04 script_sumavectorc.sh
-rwxrwxr-x 1 B1estudiante25 B1estudiante25 8488 mar 12 16:06 SumaVectoresC
-rw-r--r-- 1 B1estudiante25 B1estudiante25 3285 mar 12 16:04 SumaVectoresC.c
-rw----- 1 B1estudiante25 B1estudiante25 0 mar 12 16:05 SumaVectoresC_vglobales.e11003
-rw----- 1 B1estudiante25 B1estudiante25 2735 mar 12 16:05 SumaVectoresC_vglobales.o11003
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer8] 2019-03-12 martes
$ cat SumaVectoresC_vglobales.o11003
Id. usuario del trabajo: B1estudiante25
Id. del trabajo: 11003.atcgrid
Nombre del trabajo especificado por usuario: SumaVectoresC_vglobales
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/B1estudiante25/BP0/ejer8
Cola: ac
Nodos asignados al trabajo:
atcgrid3
Tamaño Vectores:65536 (4 B)
Tiempo:0.000563340 / 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:0.000511091 / 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:0.001456178 / 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:0.002895745 / 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:0.005677700 / 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:0.010370095 / 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:0.019406058 / 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:0.036505557 / 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:0.073091868 / 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:0.144019636 / 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:0.144409945 / 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) /
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer8] 2019-03-12 martes
$

```

Con vectores dinámicos tampoco tiene errores y lo realiza de forma correcta:

```

B1estudiante25@atcgrid:~/BP0/ejer8
Archivo Editar Ver Buscar Terminal Ayuda
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer8] 2019-03-12 martes
$head -26 SumaVectoresC.c
/* SumaVectoresC.c
Suma de dos vectores: v3 = v1 + v2

Para compilar usar (-lrt: real time library, es posible que no sea necesario usar -lrt):
gcc -O2 SumaVectores.c -o SumaVectores -lrt
gcc -O2 -S SumaVectores.c -lrt

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()

//Solo puede estar definida una de las tres constantes VECTOR_ (solo 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 ...
// generar 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)

[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer8] 2019-03-12 martes
$gcc -O2 SumaVectoresC.c -o SumaVectoresC -lrt
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer8] 2019-03-12 martes
$qsub script_sumavectorc.sh -q ac
11004.atcgrid
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer8] 2019-03-12 martes
$ls -l
total 28
-rwxr-xr-x 1 B1estudiante25 B1estudiante25 793 mar 12 16:09 script_sumavectorc.sh
-rwxrwxr-x 1 B1estudiante25 B1estudiante25 8512 mar 12 16:10 SumaVectoresC
-rw-r--r-- 1 B1estudiante25 B1estudiante25 3285 mar 12 16:09 SumaVectoresC.c
-rw----- 1 B1estudiante25 B1estudiante25 0 mar 12 16:06 SumaVectoresC_vdinamicos.e11004
-rw----- 1 B1estudiante25 B1estudiante25 2739 mar 12 16:06 SumaVectoresC_vdinamicos.o11004
-rw----- 1 B1estudiante25 B1estudiante25 0 mar 12 16:05 SumaVectoresC_vglobales.e11003
-rw----- 1 B1estudiante25 B1estudiante25 2735 mar 12 16:05 SumaVectoresC_vglobales.o11003
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer8] 2019-03-12 martes
$cat SumaVectoresC_vdinamicos.o11004
Id. usuario del trabajo: B1estudiante25
Id. del trabajo: 11004.atcgrid
Nombre del trabajo especificado por usuario: SumaVectoresC_vdinamicos
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/B1estudiante25/BP0/ejer8
Cola: ac
Nodos asignados al trabajo:
atcgrid1
Tamaño Vectores:65536 (4 B)
Tiempo:0.000467013 / 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)

```

```

B1estudiante25@atcgrid:~/BP0/ejer8
Archivo Editar Ver Buscar Terminal Ayuda
11004.atcgrid
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer8] 2019-03-12 martes
$ls -l
total 28
-rwxr-xr-x 1 B1estudiante25 B1estudiante25 793 mar 12 16:09 script_sumavectorc.sh
-rwxrwxr-x 1 B1estudiante25 B1estudiante25 8512 mar 12 16:10 SumaVectoresC
-rw-r--r-- 1 B1estudiante25 B1estudiante25 3285 mar 12 16:09 SumaVectoresC.c
-rw----- 1 B1estudiante25 B1estudiante25 0 mar 12 16:06 SumaVectoresC_vdinamicos.e11004
-rw----- 1 B1estudiante25 B1estudiante25 2739 mar 12 16:06 SumaVectoresC_vdinamicos.o11004
-rw----- 1 B1estudiante25 B1estudiante25 0 mar 12 16:05 SumaVectoresC_vglobales.e11003
-rw----- 1 B1estudiante25 B1estudiante25 2735 mar 12 16:05 SumaVectoresC_vglobales.o11003
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer8] 2019-03-12 martes
$cat SumaVectoresC_vdinamicos.o11004
Id. usuario del trabajo: B1estudiante25
Id. del trabajo: 11004.atcgrid
Nombre del trabajo especificado por usuario: SumaVectoresC_vdinamicos
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/B1estudiante25/BP0/ejer8
Cola: ac
Nodos asignados al trabajo:
atcgrid1
Tama Vector: 65536 (4 B)
Tiempo: 0.000467013 / Tama Vector: 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) /
Tama Vector: 131072 (4 B)
Tiempo: 0.000939864 / Tama Vector: 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) /
Tama Vector: 262144 (4 B)
Tiempo: 0.001884046 / Tama Vector: 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) /
Tama Vector: 524288 (4 B)
Tiempo: 0.002978039 / Tama Vector: 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) /
Tama Vector: 1048576 (4 B)
Tiempo: 0.005279473 / Tama Vector: 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) /
Tama Vector: 2097152 (4 B)
Tiempo: 0.009474164 / Tama Vector: 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) /
Tama Vector: 4194304 (4 B)
Tiempo: 0.017322818 / Tama Vector: 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) /
Tama Vector: 8388608 (4 B)
Tiempo: 0.032983131 / Tama Vector: 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) /
Tama Vector: 16777216 (4 B)
Tiempo: 0.065174684 / Tama Vector: 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) /
Tama Vector: 33554432 (4 B)
Tiempo: 0.129763834 / Tama Vector: 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) /
Tama Vector: 67108864 (4 B)
Tiempo: 0.257395414 / Tama Vector: 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) /
[AntonioDavidVillegasYeguas B1estudiante25@atcgrid:~/BP0/ejer8] 2019-03-12 martes
$

```

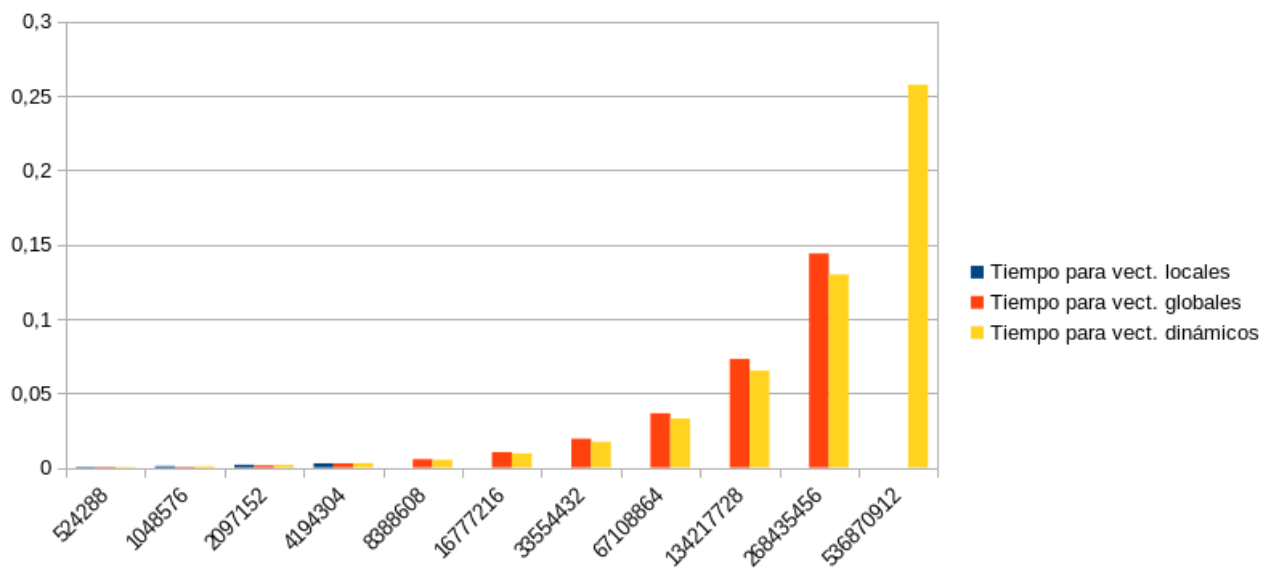
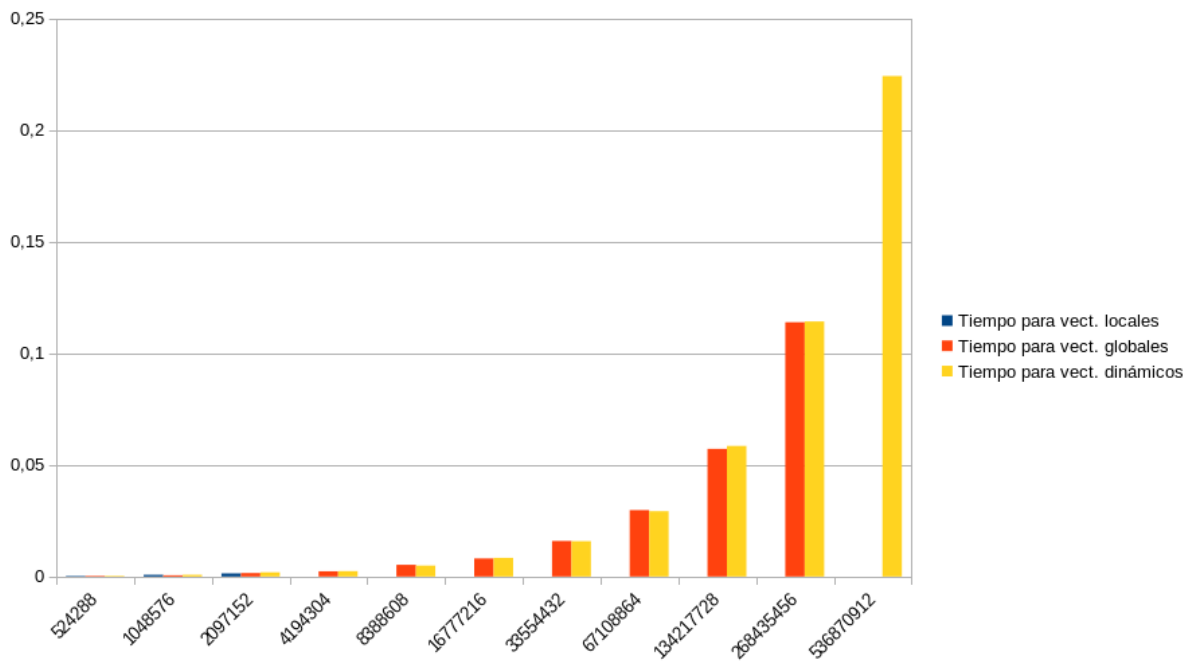

9. Rellenar una tabla como la Tabla 1 **en una hoja de cálculo** con los tiempos de ejecución obtenidos en los ejercicios anteriores para el trozo de código que realiza la suma de vectores. Debe haber una tabla para atcgrid y otra para su PC en la hoja de cálculo. En la columna “Bytes de un vector” hay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

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0,000477043	0,00056334	0,000467013
131072	1048576	0,000910645	0,000511091	0,000939864
262144	2097152	0,001881268	0,001456178	0,001884046
524288	4194304	0,002893137	0,002895745	0,002978039
1048576	8388608		0,0056777	0,005279473
2097152	16777216		0,010370095	0,009474164
4194304	33554432		0,019406058	0,017322818
8388608	67108864		0,036505557	0,032983131
16777216	134217728		0,073091868	0,065174684
33554432	268435456		0,144019636	0,129763834
67108864	536870912			0,257395414

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0,0002662	0,000275424	0,000276659
131072	1048576	0,00073831	0,000557976	0,000740787
262144	2097152	0,001367572	0,001522643	0,001852146
524288	4194304		0,002336806	0,002381669
1048576	8388608		0,005211347	0,004873994
2097152	16777216		0,008075152	0,00833362
4194304	33554432		0,015921711	0,015788736
8388608	67108864		0,029708806	0,029212335
16777216	134217728		0,05716565	0,058444054
33554432	268435456		0,113889759	0,114217549
67108864	536870912			0,224113958



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

RESPUESTA: Si, vemos como los vectores dinámicos son mucho más rápidos que los globales y (aunque no se aprecie en los gráficos) que los locales.

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

RESPUESTA: El maximo valor de N es 4294967295, ya que un unsigned int tiene 4 Bytes, que son 32 bits, luego el mayor numero seria $2^{32} - 1 = 4294967295$

(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: Es debido a que este nuevo tamaño es muy grande, y no es capaz de ubicar los vectores con este tamaño en el segmento de datos.

```
sgcc -O2 SumaVectoresC.c -o SumaVectoresC -lrt
/usr/bin/ld: /tmp/ccK57pGT.o: en la función `main':
SumaVectoresC.c:(.text.startup+0x83): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo `v2' definido en la sección COMMON en /tmp/ccK57pGT.o
/usr/bin/ld: SumaVectoresC.c:(.text.startup+0x9c): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo `v3' definido en la sección COMMON en /tmp/ccK57pGT.o
collect2: error: ld devolvió el estado de salida 1
[AntonioDavidVillegasYeguas antonio@antonio:~/Documentos/Universidad/2do/2do_cuatri/AC/Practicas/BP0/ejer10] 2019-03-07 jueves
$
```

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
```