

2º curso / 2º cuatr.
Grado Ing. Inform.
Doble Grado Ing.
Inform. y Mat.

Arquitectura de Computadores (AC)

Cuaderno de prácticas.

Bloque Práctico 0. Entorno de programación

Estudiante (nombre y apellidos): José Javier Alonso Ramos

Grupo de prácticas: A3

Fecha de entrega:

Fecha evaluación en clase:

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

CAPTURAS:

```
Actividades Terminal vie 10:24
jjavier98@jjavier98-Lenovo-ideapad-310-15IKB: ~/GII/2º Semestre/AC/Seminario_0/ejercicio
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
jjavier98@jjavier98-Lenovo-ideapad-310-15IKB: ~/GII/2º Semestre/AC/Seminario_0/ejercicio
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º Semestre/AC/Seminario_0/ejercicio] 2018-03-02 viernes
$ lscpu
Arquitectura: x86_64
modo(s) de operación de las CPUs: 32-bit, 64-bit
Orden de bytes: Little Endian
CPU(s): 4
On-line CPU(s) list: 0-3
Hilo(s) de procesamiento por núcleo: 2
Núcleo(s) por «socket»: 2
Socket(s): 1
Modo(s) NUMA: 1
ID de fabricante: GenuineIntel
Familia de CPU: 6
Modelo: 142
Model name: Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz
Revisión: 9
CPU MHz: 2700.000
CPU max MHz: 2500.0000
CPU min MHz: 400.0000
BogoMIPS: 5424.00
Virtualización: VT-x
Caché L1d: 32K
Caché L1i: 32K
Caché L2: 256K
Caché L3: 3072K
NUMA node0 CPU(s): 0-3
Flags: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 s
s ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf tsc_known_fr
eq pni pclmulqdq dtes64 monitor ds_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xs
ave avx f16c rdrand lahf_lm abm 3dnowprefetch cpuid_fault invpcid_single pti retpoline intel_pt rsb_ctxsw tpr_shadow vnmi flexpriority ept vpid f
sgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid mpx rdseed adx smap clflushopt xsaveopt xsavec xgetbv1 xsaves dtherm arat pln pts hwp hwp_not
ify hwp_act_window hwp_epp
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º Semestre/AC/Seminario_0/ejercicio] 2018-03-02 viernes
$
```

```

Actividades Terminal vie 10:26
A3estudiante1@atcgrid:~

Archivo Editar Ver Buscar Terminal Pestañas Ayuda
jjavier98@jjavier98-Lenovo-ideapad-310-15IKB: ~/GII/2... x A3estudiante1@atcgrid:~ jjavier98@jjavier98-Lenovo-ideapad-310-15IKB: ~/GII/2... x

[A3estudiante1@atcgrid ~]$ lscpu
Arquitectura: x86_64
modo(s) de operación de las CPUs: 32-bit, 64-bit
Orden de los bytes: Little Endian
CPU(s): 1
Lista de la(s) CPU(s) en línea: 0
Hilo(s) de procesamiento por núcleo: 1
Núcleo(s) por «socket»: 1
«Socket(s)»: 1
Modo(s) NUMA: 1
ID de fabricante: AuthenticAMD
Familia de CPU: 15
Modelo: 47
Nombre del modelo: AMD Athlon(tm) 64 Processor 3000+
Revisión: 2
CPU MHz: 1802.252
BogoMIPS: 3604.50
Caché L1d: 64K
Caché L1i: 64K
Caché L2: 512K
CPU(s) del nodo NUMA 0: 0
Indicadores: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 syscall nx
mmxext fxsr_opt lm 3dnowext 3dnow rep_good nopl cpuid extd_apicid pni lahf_lm 3dnowprefetch retpoline rsb_ctxsw vmxcall
[A3estudiante1@atcgrid ~]$

```

```

Actividades Terminal vie 10:28
jjavier98@jjavier98-Lenovo-ideapad-310-15IKB: ~/GII/2º/2º Semestre/AC/Seminario_0/ejercicio

Archivo Editar Ver Buscar Terminal Pestañas Ayuda
jjavier98@jjavier98-Lenovo-ideapad-310-15IKB: ~/GII/2... x A3estudiante1@atcgrid:~ jjavier98@jjavier98-Lenovo-ideapad-310-15IKB: ~/GII/2... x

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB: ~/GII/2º/2º Semestre/AC/Seminario_0/ejercicio] 2018-03-02 viernes
$cat STDIN.063059
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: 1599.149
CPU max MHz: 2401.0000
CPU min MHz: 1600.0000
BogoMIPS: 4800.14
Virtualization: VT-x
L1d cache: 32K
L1i cache: 32K
L2 cache: 256K
L3 cache: 12288K
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 fxsr sse sse2 ss ht tm pbe sysc
all nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfperf pni dtes64 monitor ds_cpl vmx sm
x est tm2 sse3 cx16 xtpr pdcm pcid dca sse4_1 sse4_2 popcnt lahf_lm epb pti retpoline tpr_shadow vmx flexpriority ept vpid dtherm ida arat
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB: ~/GII/2º/2º Semestre/AC/Seminario_0/ejercicio] 2018-03-02 viernes
$

```

Conteste a las siguientes preguntas:

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

RESPUESTA: PC: 2 núcleos físicos, 4 lógicos ; Front-end: 1 núcleo físico, 1 lógico

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

RESPUESTA: 2 procesadores, 6 núcleos físicos y 12 lógicos cada uno. Luego en total: 12 físicos y 24 lógicos.

2. En el Listado 1 se puede ver un código fuente C que calcula la suma de dos vectores y en el Listado 2 una versión con C++:

```
v3 = v1 + v2; v3(i) = v1(i) + v2(i), i=0,...N-1
```

Los códigos utilizan directivas del compilador para fijar el tipo de variable de los vectores (v1, v2 y v3). En los comentarios que hay al principio de los códigos se indica cómo hay que compilarlos. Los vectores pueden ser:

- Variables locales: descomentando en el código `#define VECTOR_LOCAL` y comentando `#define VECTOR_GLOBAL` y `#define VECTOR_DYNAMIC`
- Variables globales: descomentando `#define VECTOR_GLOBAL` y comentando `#define VECTOR_LOCAL` y `#define VECTOR_DYNAMIC`
- Variables dinámicas: descomentando `#define VECTOR_DYNAMIC` y comentando `#define VECTOR_LOCAL` y `#define VECTOR_GLOBAL`. Si se usan los códigos tal y como están en Listado 1 y Listado 2, sin hacer ningún cambio, los vectores (v1, v2 y v3) serán variables dinámicas.

Por tanto, se debe definir sólo una de las siguientes constantes: `VECTOR_LOCAL`, `VECTOR_GLOBAL` o `VECTOR_DYNAMIC`.

- a. En los dos códigos (Listado 1 y Listado 2) se utiliza la función `clock_gettime()` para obtener el tiempo de ejecución del trozo de código que calcula la suma de vectores. En el código se imprime la variable `ncgt`, ¿qué contiene esta variable? ¿qué información devuelve exactamente la función `clock_gettime()`? ¿en qué estructura de datos devuelve `clock_gettime()` la información (indicar el tipo de estructura de datos y describir la estructura de datos)?

RESPUESTA:

La variable contiene la diferencia de tiempo entre el momento final de la suma de vectores y el momento inicial obteniendo la duración de la suma.

Para calcular los instantes de tiempo utiliza `clock_gettime()` que devuelve un instante de tiempo relativo al reloj pasado como primer argumento (en el programa: el reloj de sistema de tiempo real → `CLOCK_REALTIME`) este instante lo devuelve en una estructura pasada como segundo argumento, en este caso 'timespec' que consta de dos datos miembros: 'tv_sec' de tipo `time_t` y 'tv_nsec' de tipo `double`.

```
time_t tv_sec    (valores válidos >= 0)
long tv_nsec    (valores válidos [0, 999999999])
```

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

RESPUESTA:

Descripción diferencia	En C	En C++
Asignación de memoria dinámica	Se necesita el numero de Bytes para asignarla y utiliza malloc()	Solo se necesita especificar el espacio necesario. Utiliza new()
Cuando falta la asignación de memoria	Devuelve null	Devuelve una excepción
Forma de liberar memoria	Utiliza free()	Utiliza delete []
Mostrar por pantalla	Usa la funcion printf()	Usa la función cout
Declaración de 'i' y 'ncgt'	Al inicio del 'main()'	A la vez que su asignación

3. Generar el ejecutable del código fuente C del Listado 1 para vectores locales (para ello antes de compilar debe descomentar la definición de VECTOR_LOCAL y comentar las definiciones de VECTOR_GLOBAL y VECTOR_DYNAMIC). Incorporar volcados de pantalla que demuestren la ejecución correcta en atcgrid o en su PC.

RESPUESTA:

Como era de esperar los resultados obtenidos en el PC local y en el nodo de atcgrid son los mismos pero obteniendo un tiempo menor en atcgrid.

```

lun 19:29
jjavier98@jjavier98-Lenovo-ideapad-310-15IKB: ~/GII/2º/2º Semestre/AC/Practica_0/Listados
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
jjavier98@jjavier98-Lenovo-ideapad-310-15IKB: ~/GII/2... x  jjavier98@jjavier98-Lenovo-ideapad-310-15IKB: ~ x  A3estudiante1@atcgrid: ~ x
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados] 2018-03-05 lunes
$ll
total 16
drwxr-xr-x 2 jjavier98 jjavier98 4096 mar  5 19:28 ./
drwxr-xr-x 3 jjavier98 jjavier98 4096 mar  5 19:13 ../
-rw-r--r-- 1 jjavier98 jjavier98 3344 mar  5 19:14 Listado1.c
-rw-r--r-- 1 jjavier98 jjavier98 3487 mar  5 19:14 Listado2.cpp
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados] 2018-03-05 lunes
$gcc -O2 Listado1.c -o SumaVectores -lrt
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados] 2018-03-05 lunes
$ll
total 28
drwxr-xr-x 2 jjavier98 jjavier98 4096 mar  5 19:28 ./
drwxr-xr-x 3 jjavier98 jjavier98 4096 mar  5 19:13 ../
-rw-r--r-- 1 jjavier98 jjavier98 3344 mar  5 19:14 Listado1.c
-rw-r--r-- 1 jjavier98 jjavier98 3487 mar  5 19:14 Listado2.cpp
-rwxr-xr-x 1 jjavier98 jjavier98 8536 mar  5 19:28 SumaVectores*
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados] 2018-03-05 lunes
$./SumaVectores 20000
Tiempo(seg.):0.000327826      / Tamaño Vectores:20000      / V1[0]+V2[0]=V3[0](2000.000000+2000.000000=4000.000000) / / V1[19999]+V2[19999]=
V3[19999](3999.900000+0.100000=4000.000000) /
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados] 2018-03-05 lunes
$./SumaVectores 200
Tiempo(seg.):0.000001672      / Tamaño Vectores:200 / V1[0]+V2[0]=V3[0](20.000000+20.000000=40.000000) / / V1[199]+V2[199]=V3[199](39.900000+
0.100000=40.000000) /
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados] 2018-03-05 lunes
$

```

```

Actividades Terminal
lun 19:41
A3estudiante1@atcgrid:~

Archivo Editar Ver Buscar Terminal Pestañas Ayuda
jjavier98@jjavier98-Lenovo-ideapad-310-15IKB: ~/GII/2... x jjavier98@jjavier98-Lenovo-ideapad-310-15IKB: ~/GII/2... x A3estudiante1@atcgrid:~ x

[A3estudiante1@atcgrid ~]$ ll
total 12
-rwxr-xr-x 1 A3estudiante1 A3estudiante1 8536 mar  5 19:38 SumaVectores
[A3estudiante1@atcgrid ~]$ echo './SumaVectores 20000' | qsub -q ac
64991.atcgrid
[A3estudiante1@atcgrid ~]$ ll
total 16
-rw----- 1 A3estudiante1 A3estudiante1    0 mar  5 19:40 STDIN.e64991
-rw----- 1 A3estudiante1 A3estudiante1  178 mar  5 19:40 STDIN.o64991
-rwxr-xr-x 1 A3estudiante1 A3estudiante1 8536 mar  5 19:38 SumaVectores
[A3estudiante1@atcgrid ~]$ cat STDIN.o64991
Tiempo(seg.):0.000144574 / Tamaño Vectores:20000 / V1[0]+V2[0]=V3[0](2000.000000+2000.000000=4000.000000) / / V1[19999]+V2[19999]=
V3[19999](3999.900000+0.100000=4000.000000) /
[A3estudiante1@atcgrid ~]$ echo './SumaVectores 200' | qsub -q ac
64992.atcgrid
[A3estudiante1@atcgrid ~]$ ll
total 20
-rw----- 1 A3estudiante1 A3estudiante1    0 mar  5 19:40 STDIN.e64991
-rw----- 1 A3estudiante1 A3estudiante1    0 mar  5 19:41 STDIN.e64992
-rw----- 1 A3estudiante1 A3estudiante1  178 mar  5 19:40 STDIN.o64991
-rw----- 1 A3estudiante1 A3estudiante1  160 mar  5 19:41 STDIN.o64992
-rwxr-xr-x 1 A3estudiante1 A3estudiante1 8536 mar  5 19:38 SumaVectores
[A3estudiante1@atcgrid ~]$ cat STDIN.64992
cat: STDIN.64992: No such file or directory
[A3estudiante1@atcgrid ~]$ cat STDIN.o64992
Tiempo(seg.):0.000000545 / Tamaño Vectores:200 / V1[0]+V2[0]=V3[0](20.000000+20.000000=40.000000) / / V1[199]+V2[199]=V3[199](39.900000+
0.100000=40.000000) /
[A3estudiante1@atcgrid ~]$

```

4. Ejecutar en atcgrid el código generado en el apartado anterior usando el script del Listado 3. Generar el ejecutable usando la opción de optimización `-O2` tal y como se indica en el comentario que hay al principio del programa. Ejecutar el código también en su PC para los mismos tamaños. ¿Se obtiene error para alguno de los tamaños? En caso afirmativo, ¿a qué se debe este error? (Incorporar volcados de pantalla)

RESPUESTA: SCRIPT EN ATCGRID

The screenshot displays a Linux terminal window titled "A3estudiante1@atcgrid::~". The terminal shows the following sequence of commands and output:

```
[A3estudiante1@atcgrid ~]$ ll
total 16
-rwxr-xr-x 1 A3estudiante1 A3estudiante1 8536 mar  6 12:07 SumaVectoresC
-rwxrwxr-x 1 A3estudiante1 A3estudiante1  745 mar  6 12:15 SumaVectores.sh
[A3estudiante1@atcgrid ~]$ qsub SumaVectores.sh
65233.atcgrid
[A3estudiante1@atcgrid ~]$ ll
total 24
-rwxr-xr-x 1 A3estudiante1 A3estudiante1 8536 mar  6 12:07 SumaVectoresC
-rw----- 1 A3estudiante1 A3estudiante1 1064 mar  6 12:18 SumaVectoresC_vlocales.e65233
-rw----- 1 A3estudiante1 A3estudiante1 1042 mar  6 12:18 SumaVectoresC_vlocales.o65233
-rwxrwxr-x 1 A3estudiante1 A3estudiante1  745 mar  6 12:15 SumaVectores.sh
[A3estudiante1@atcgrid ~]$ cat SumaVectoresC_vlocales.*
/var/lib/torque/mom_priv/jobs/65233.atcgrid.SC: line 20: 32300 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
/var/lib/torque/mom_priv/jobs/65233.atcgrid.SC: line 20: 32303 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
/var/lib/torque/mom_priv/jobs/65233.atcgrid.SC: line 20: 32306 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
/var/lib/torque/mom_priv/jobs/65233.atcgrid.SC: line 20: 32311 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
/var/lib/torque/mom_priv/jobs/65233.atcgrid.SC: line 20: 32317 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
/var/lib/torque/mom_priv/jobs/65233.atcgrid.SC: line 20: 32320 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
/var/lib/torque/mom_priv/jobs/65233.atcgrid.SC: line 20: 32323 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
/var/lib/torque/mom_priv/jobs/65233.atcgrid.SC: line 20: 32326 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
Id. usuario del trabajo: A3estudiante1
Id. del trabajo: 65233.atcgrid
Nombre del trabajo especificado por usuario: SumaVectoresC_vlocales
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/A3estudiante1
Cola: ac
Nodos asignados al trabajo:
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
```

The second part of the image shows the continuation of the job output, displaying timing information and vector sizes for each node:

```
Tiempo(seg.):0.000270215          / Tamaño Vectores:65536          / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / / V1[65535]+V2[65535]
=V3[65535](13107.100000+0.100000=13107.200000) /
Tiempo(seg.):0.000859796          / Tamaño Vectores:131072        / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / / V1[131071]+V2[131
071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tiempo(seg.):0.001265571          / Tamaño Vectores:262144        / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]+V2[262
143]=V3[262143](52428.700000+0.100000=52428.800000) /
[A3estudiante1@atcgrid ~]$
```


Se obtienen errores de ejecución porque al ser los vectores variables locales se almacenan en la pila. Al aumentar mucho el tamaño de éstos sobrepasamos la capacidad de la pila y el programa aborta.

EJECUCIÓN DEL PROGRAMA CON LOS MISMOS TAMAÑOS EN PC LOCAL

```

jjavier98@jjavier98-Lenovo-ideapad-310-15IKB: ~/GII/2º/2º Semestre/AC/Practica_0/Listados
$ls
SumaVectorC.c SumaVectorC++.cpp SumaVectoresC SumaVectores.sh
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados] 2018-03-06 martes
$./SumaVectoresC 65536
Tiempo(seg.):0.000300079 / 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) /
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados] 2018-03-06 martes
$./SumaVectoresC 131072
Tiempo(seg.):0.001117609 / 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) /
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados] 2018-03-06 martes
$./SumaVectoresC 262144
Tiempo(seg.):0.006041759 / 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) /
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados] 2018-03-06 martes
$
  
```

Se obtienen los mismos errores (desbordamiento de pila) a partir del mismo tamaño que en atcgrid.

5. Generar los ejecutables del código fuente C para vectores globales y para dinámicos. Genere el ejecutable usando -O2. Ejecutar los dos códigos en atcgrid usando un script como el del Listado 3 (hay que poner en el script el nombre de los ficheros ejecutables generados en este ejercicio) para el mismo rango de tamaños utilizado en el ejercicio anterior. Ejecutar también los códigos en su PC. ¿Se obtiene error usando vectores globales o dinámicos? ¿A qué cree que es debido? (Incorporar volcados de pantalla)

Depto. Arquitectura y Tecnología de Computadores

VECTORES GLOBALES PC LOCAL

```

Actividades Terminal mar 23:28
jjavier98@jjavier98-Lenovo-ideapad-310-15IKB: ~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin

Archivo Editar Ver Buscar Terminal Ayuda

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_globales 65536
Tiempo(seg.):0.001107937 / 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) /

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_globales 131072
Tiempo(seg.):0.002277472 / 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) /

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_globales 262144
Tiempo(seg.):0.006847666 / 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) /

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_globales 524288
Tiempo(seg.):0.002111146 / 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) /

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_globales 1048576
Tiempo(seg.):0.007289663 / 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) /

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_globales 2097152
Tiempo(seg.):0.008444181 / 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) /

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_globales 4194304
Tiempo(seg.):0.016487723 / 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) /

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_globales 8388608
Tiempo(seg.):0.032369435 / 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) /

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_globales 16777216
Tiempo(seg.):0.063104418 / 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) /

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_globales 33554432

Actividades Terminal mar 23:28
jjavier98@jjavier98-Lenovo-ideapad-310-15IKB: ~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin

Archivo Editar Ver Buscar Terminal Ayuda

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_globales 262144
Tiempo(seg.):0.006847666 / 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) /

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_globales 524288
Tiempo(seg.):0.002111146 / 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) /

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_globales 1048576
Tiempo(seg.):0.007289663 / 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) /

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_globales 2097152
Tiempo(seg.):0.008444181 / 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) /

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_globales 4194304
Tiempo(seg.):0.016487723 / 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) /

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_globales 8388608
Tiempo(seg.):0.032369435 / 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) /

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_globales 16777216
Tiempo(seg.):0.063104418 / 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) /

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_globales 33554432
Tiempo(seg.):0.127344041 / 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) /

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_globales 67108864
Tiempo(seg.):0.128565255 / 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) /

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$

```

Vectores de tamaño 67108864 no entran en el segmento de datos de memoria pero en vez de dar error se adecua el tamaño a 33554432.

VECTORES DINÁMICOS EN ATCGRID

[illegible]

```

javier98@javier98-Lenovo-Ideapad-310-15IKB: ~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin
Archivo Editar Ver Buscar Terminal Ayuda
[JoseJavierAlonsoRamos j javier98@javier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_dinamicos 65536
Tiempo(seg.):0.001129377 / 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) /
[JoseJavierAlonsoRamos j javier98@javier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_dinamicos 131072
Tiempo(seg.):0.002238768 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / / V1[131071]+V2[131
071]=V3[131071](26214.300000+0.100000=26214.400000) /
[JoseJavierAlonsoRamos j javier98@javier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_dinamicos 262144
Tiempo(seg.):0.004979504 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]+V2[262
143]=V3[262143](52428.700000+0.100000=52428.800000) /
[JoseJavierAlonsoRamos j javier98@javier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_dinamicos 524288
Tiempo(seg.):0.008720256 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / / V1[524287]+V2[52
4287]=V3[524287](104857.500000+0.100000=104857.600000) /
[JoseJavierAlonsoRamos j javier98@javier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_dinamicos 1048576
Tiempo(seg.):0.006330589 / 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) /
[JoseJavierAlonsoRamos j javier98@javier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_dinamicos 2097152
Tiempo(seg.):0.009186522 / 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) /
[JoseJavierAlonsoRamos j javier98@javier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_dinamicos 4194304
Tiempo(seg.):0.016386140 / 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) /
[JoseJavierAlonsoRamos j javier98@javier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_dinamicos 8388608
Tiempo(seg.):0.031955328 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / / V1[8388607]+V
[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000) /
[JoseJavierAlonsoRamos j javier98@javier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_dinamicos 16777216
Tiempo(seg.):0.063132502 / 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) /
[JoseJavierAlonsoRamos j javier98@javier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$./SumaVectoresC_dinamicos 33554432

```



```

Actividades Terminal mar 23:57
jjavier98@jjavier98-Lenovo-ideapad-310-15IKB: ~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin

[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$ ./SumaVectoresC_dinamicos 262144
Tiempo(seg.):0.004979504 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]+V2[262
143]=V3[262143](52428.700000+0.100000=52428.800000) /
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$ ./SumaVectoresC_dinamicos 524288
Tiempo(seg.):0.008720256 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / / V1[524287]+V2[52
4287]=V3[524287](104857.500000+0.100000=104857.600000) /
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$ ./SumaVectoresC_dinamicos 1048576
Tiempo(seg.):0.006330589 / 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) /
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$ ./SumaVectoresC_dinamicos 2097152
Tiempo(seg.):0.009186522 / 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) /
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$ ./SumaVectoresC_dinamicos 4194304
Tiempo(seg.):0.016386140 / 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) /
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$ ./SumaVectoresC_dinamicos 8388608
Tiempo(seg.):0.031955328 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / / V1[8388607]+V
2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000) /
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$ ./SumaVectoresC_dinamicos 16777216
Tiempo(seg.):0.063132502 / 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) /
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$ ./SumaVectoresC_dinamicos 33554432
Tiempo(seg.):0.126967413 / 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) /
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$ ./SumaVectoresC_dinamicos 67108864
Tiempo(seg.):0.339207270 / 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) /
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/bin] 2018-03-06 martes
$

```

No se obtienen errores ya que los vectores globales se almacenan en el segmento de datos y los dinámicos en el heap. Estas dos zonas de memoria tienen el tamaño suficiente para albergar los vectores sin que se produzca desbordamiento. Hemos visto que en el caso de vectores globales se redimensiona el vector para que entre en memoria y no salte una excepción mientras que los vectores dinámicos pueden crecer sin problema.

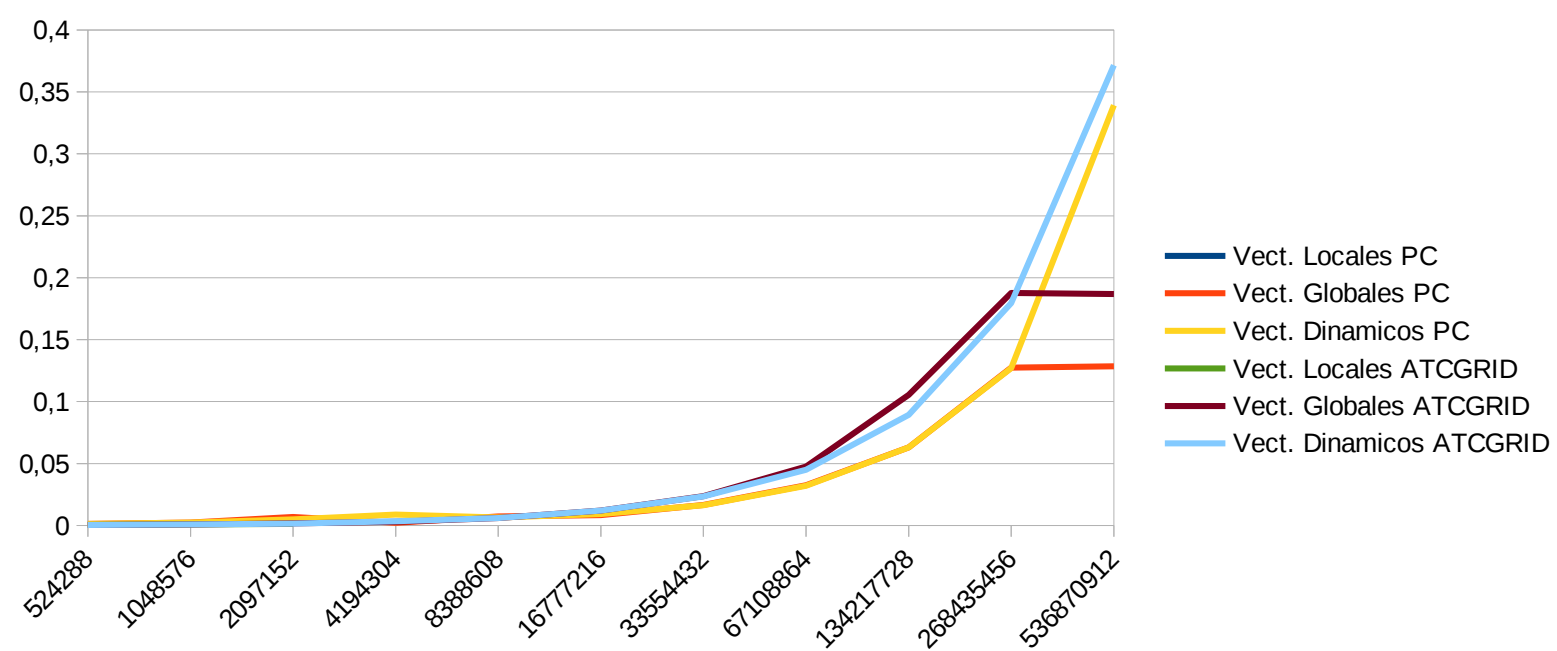
6. Rellenar una tabla como la Tabla 1 para atcgrid y otra para su PC con los tiempos de ejecución obtenidos en los ejercicios anteriores para el trozo de código que realiza la suma de vectores. En la columna “Bytes de un vector” hay que poner el total de bytes reservado para un vector. Ayudándose de una hoja de cálculo represente en una misma gráfica los tiempos de ejecución obtenidos en atcgrid y en su PC para vectores locales, globales y dinámicos (eje y) en función del tamaño en bytes de un vector (los valores de la segunda columna de la tabla, que están en escala logarítmica, deben estar en el eje x). Utilice escala logarítmica en el eje de ordenadas (eje y). ¿Hay diferencias en los tiempos de ejecución?

RESPUESTA:**PC LOCAL**

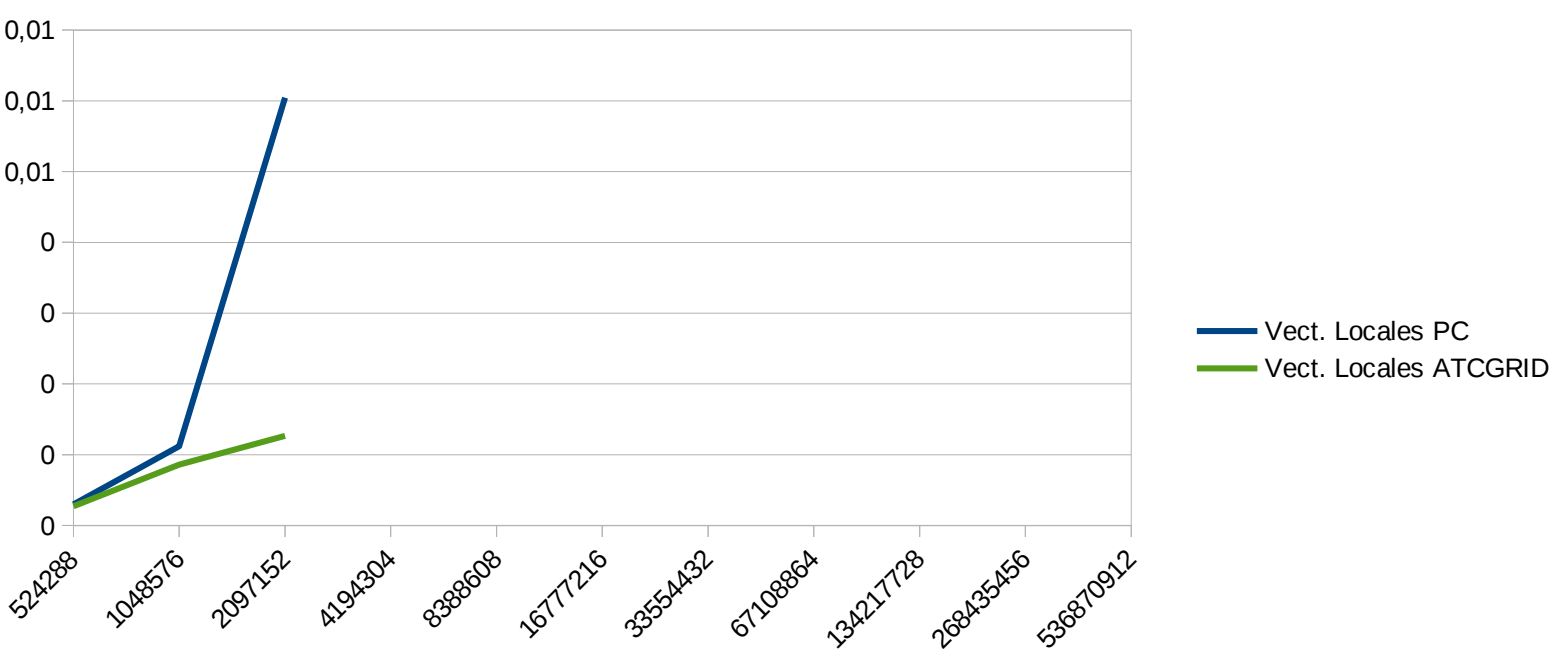
n.º componentes	bytes del vector	tiempo para vect. Locales	tiempo para vect. Globales	tiempo para vect. Dinámicos
65536	524288	0,000300079	0,001107937	0,001129377
131072	1048576	0,001117609	0,002277472	0,002238768
262144	2097152	0,006041759	0,006847666	0,004979504
524288	4194304 excepción		0,002111146	0,008720256
1048576	8388608 excepción		0,007289663	0,006330589
2097152	16777216 excepción		0,008444181	0,009186522
4194304	33554432 excepción		0,016487723	0,01638614
8388608	67108864 excepción		0,032369435	0,031955328
16777216	134217728 excepción		0,063104418	0,063132502
33554432	268435456 excepción		0,127344041	0,126967413
67108864	536870912 excepción		0,128565255	0,33920727

ATCGRID

n.º componentes	bytes del vector	tiempo para vect. Locales	tiempo para vect. Globales	tiempo para vect. Dinámicos
65536	524288	0,000270215	0,000423254	0,000422906
131072	1048576	0,000859796	0,000878351	0,00082747
262144	2097152	0,001265571	0,001712852	0,001411379
524288	4194304 excepción		0,002986076	0,003330158
1048576	8388608 excepción		0,006036517	0,005876513
2097152	16777216 excepción		0,012020302	0,012059453
4194304	33554432 excepción		0,023591602	0,023372877
8388608	67108864 excepción		0,047606865	0,044985447
16777216	134217728 excepción		0,105404895	0,089218313
33554432	268435456 excepción		0,187818538	0,179458974
67108864	536870912 excepción		0,186814452	0,371680261



Para representar mejor los vectores locales miramos la siguiente gráfica:



A partir de vectores de un tamaño grande (aprox la mitad del eje X) el PC local realiza la tarea en menor tiempo.

7. Modificar el código fuente C para que el límite de los vectores cuando se declaran como variables globales sea igual al máximo número que se puede almacenar en la variable N ($MAX=2^{32}-1$). Generar el ejecutable usando variables globales. ¿Qué ocurre? ¿A qué es debido? Razone además por qué el máximo número que se puede almacenar en N es $2^{32}-1$.

RESPUESTA:

No compila porque sobrepasa el límite de 2GB permitido en el segmento de datos. Además el tamaño máximo es $2^{32}-1$ porque es el mayor número de 32bits.

```

jjavier98@jjavier98-Lenovo-ideapad-310-15IKB: ~/GII/2º/2º Semestre/AC/Practica_0/Listados/src
Archivo Editar Ver Buscar Terminal Ayuda
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/src] 2018-03-07 miércoles
$gcc -O2 SumaVectorC.c -o ../bin/SumaVectores32-1 -lrt
/tmp/ccBTbj7d.o: En la función 'main':
SumaVectorC.c:(.text.startup+0x5e): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo 'v2' definido en la sección COMMON en /tmp/ccBTbj7d.o
SumaVectorC.c:(.text.startup+0xb1): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo 'v3' definido en la sección COMMON en /tmp/ccBTbj7d.o
SumaVectorC.c:(.text.startup+0x130): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo 'v3' definido en la sección COMMON en /tmp/ccBTbj7d.o
SumaVectorC.c:(.text.startup+0x13b): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo 'v2' definido en la sección COMMON en /tmp/ccBTbj7d.o
SumaVectorC.c:(.text.startup+0x1a0): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo 'v2' definido en la sección COMMON en /tmp/ccBTbj7d.o
SumaVectorC.c:(.text.startup+0x1a7): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo 'v3' definido en la sección COMMON en /tmp/ccBTbj7d.o
collect2: error: ld returned 1 exit status
[JoseJavierAlonsoRamos jjavier98@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/AC/Practica_0/Listados/src] 2018-03-07 miércoles
$

```

Tabla 1 .

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536				
131072				
262144				
524288				
1048576				
2097152				
4194304				
8388608				
16777216				
33554432				
67108864				

Listado 1. Código C que suma dos vectores

```

/* SumaVectoresC.c
Suma de dos vectores: v3 = v1 + v2

Para compilar usar (-lrt: real time library):
    gcc -O2 SumaVectores.c -o SumaVectores -lrt
gcc -O2 -S SumaVectores.c -lrt //para generar el código ensamblador

Para ejecutar use: SumaVectoresC longitud
*/

#include <stdlib.h> // biblioteca con funciones atoi(), malloc() y free()
#include <stdio.h> // biblioteca donde se encuentra la función printf()
#include <time.h> // biblioteca donde se encuentra la función clock_gettime()

// #define PRINTF_ALL // comentar para quitar el printf ...
// // que imprime todos los componentes
// // Sólo puede estar definida una de las tres constantes VECTOR_ (sólo uno de los ...
// // tres defines siguientes puede estar descomentado):
// #define VECTOR_LOCAL // descomentar para que los vectores sean variables ...
// // locales (si se supera el tamaño de la pila se ...
// // generará el error "Violación de Segmento")
// #define VECTOR_GLOBAL // descomentar para que los vectores sean variables ...
// // globales (su longitud no estará limitada por el ...
// // tamaño de la pila del programa)
#define VECTOR_DYNAMIC // descomentar para que los vectores sean variables ...
// // dinámicas (memoria reutilizable durante la ejecución)

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

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

    int i;
    struct timespec cgt1, cgt2; double ncgt; // para tiempo de ejecución

    // Leer argumento de entrada (nº de componentes del vector)
    if (argc < 2){
        printf("Faltan nº componentes del vector\n");
        exit(-1);
    }

    unsigned int N = atoi(argv[1]); // Máximo N = 2^32-1 = 4294967295 (sizeof(unsigned int) = 4 B)
    #ifdef VECTOR_LOCAL
        double v1[N], v2[N], v3[N]; // Tamaño variable local en tiempo de ejecución ...
        // disponible en C a partir de actualización C99
    #endif
    #ifdef VECTOR_GLOBAL
        if (N > MAX) N = MAX;
    #endif
    #ifdef VECTOR_DYNAMIC
        double *v1, *v2, *v3;
        v1 = (double*) malloc(N*sizeof(double)); // malloc necesita el tamaño en bytes
        v2 = (double*) malloc(N*sizeof(double)); // si no hay espacio suficiente malloc

```

```

devuelve NULL
v3 = (double*) malloc(N*sizeof(double));
if ( (v1==NULL) || (v2==NULL) || (v3==NULL) ){
    printf("Error en la reserva de espacio para los vectores\n");
    exit(-2);
}
#endif

//Inicializar vectores
for(i=0; i<N; i++){
    v1[i] = N*0.1+i*0.1; v2[i] = N*0.1-i*0.1; //los valores dependen de N
}

clock_gettime(CLOCK_REALTIME,&cgt1);
//Calcular suma de vectores
for(i=0; i<N; i++)
    v3[i] = v1[i] + v2[i];

clock_gettime(CLOCK_REALTIME,&cgt2);
ncgt=(double) (cgt2.tv_sec-cgt1.tv_sec)+
    (double) ((cgt2.tv_nsec-cgt1.tv_nsec)/(1.e+9));

//Imprimir resultado de la suma y el tiempo de ejecución
#ifdef PRINTF_ALL
printf("Tiempo(seg.):%11.9f\t / Tamaño Vectores:%u\n",ncgt,N);
for(i=0; i<N; i++)
    printf("/ v1[%d]+v2[%d]=v3[%d](%8.6f+%8.6f=%8.6f) /\n",
        i,i,i,v1[i],v2[i],v3[i]);

#else
printf("Tiempo(seg.):%11.9f\t / Tamaño Vectores:%u\t/ v1[0]+v2[0]=v3[0](%8.6f+
%8.6f=%8.6f) / /
        v1[%d]+v2[%d]=v3[%d](%8.6f+%8.6f=%8.6f) /\n",
        ncgt,N,v1[0],v2[0],v3[0],N-1,N-1,N-1,v1[N-1],v2[N-1],v3[N-1]);
#endif

#ifdef VECTOR_DYNAMIC
free(v1); // libera el espacio reservado para v1
free(v2); // libera el espacio reservado para v2
free(v3); // libera el espacio reservado para v3
#endif
return 0;
}

```

Listado 2. Código C++ que suma dos vectores

```

/* SumaVectoresCpp.cpp
Suma de dos vectores: v3 = v1 + v2

Para compilar usar (-lrt: real time library):
    g++ -O2 SumaVectoresCpp.cpp -o SumaVectoresCpp -lrt

Para ejecutar use: SumaVectoresCpp longitud

```



```

*/

#include <cstdlib> // biblioteca con atoi()
#include <iostream> // biblioteca donde se encuentra la función cout
using namespace std;
#include <time.h> // biblioteca donde se encuentra la función clock_gettime()

// #define COUT_ALL // comentar para quitar el cout ...
// que imprime todos los componentes
// Sólo puede estar definida una de las tres constantes VECTOR_ (sólo uno de los ...
// tres defines siguientes puede estar descomentado):
// #define VECTOR_LOCAL // descomentar para que los vectores sean variables ...
// locales (si se supera el tamaño de la pila se ...
// generará el error "Violación de Segmento")
// #define VECTOR_GLOBAL // descomentar para que los vectores sean variables ...
// globales (su longitud no estará limitada por el ...
// tamaño de la pila del programa)
#define VECTOR_DYNAMIC // descomentar para que los vectores sean variables ...
// dinámicas (memoria reutilizable durante la ejecución)

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

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

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

    // Leer argumento de entrada (nº de componentes del vector)
    if (argc < 2){
        cout << "Faltan nº componentes del vector\n" << endl;
        exit(-1);
    }

    unsigned int N = atoi(argv[1]);
    #ifdef VECTOR_LOCAL
    double v1[N], v2[N], v3[N];
    #endif
    #ifdef VECTOR_GLOBAL
    if (N > MAX) N = MAX;
    #endif
    #ifdef VECTOR_DYNAMIC
    double *v1, *v2, *v3;
    v1 = new double [N]; // si no hay espacio suficiente new genera una excepción
    v2 = new double [N];
    v3 = new double [N];
    #endif

    // Inicializar vectores
    for(int i=0; i<N; i++){
        v1[i] = N*0.1+i*0.1; v2[i] = N*0.1-i*0.1; // los valores dependen de N
    }
    clock_gettime(CLOCK_REALTIME, &cgt1);
    // Calcular suma de vectores
    for(int i=0; i<N; i++){
        v3[i] = v1[i] + v2[i];
        clock_gettime(CLOCK_REALTIME, &cgt2);
    }
}

```

```

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

//Imprimir resultado de la suma y el tiempo de ejecución
#ifdef COUT_ALL
cout << "Tiempo(seg.):" << ncgt << "\t/ Tamaño Vectores:" << N << endl;
for(int i=0; i<N; i++)
    cout << "/" V1[" << i << "]+V2[" << i << "]=V3" << i << "]" << v1[i] << "+"
<< v2[i] << "="
    << v3[i] << ") /\t" << endl;
cout << "\n" << endl;
#else
    cout << "Tiempo(seg.):" << ncgt << "\t/ Tamaño Vectores:" << N << "\t/
V1[0]+V2[0]=V3[0]("
    << v1[0] << "+" << v2[0] << "=" << v3[0] << ") / / V1[" << N-1 << "]+V2["
<< N-1 << "]=V3["
    << N-1 << "]" << v1[N-1] << "+" << v2[N-1] << "=" << v3[N-1] << ")/\n" <<
endl;
#endif

#ifdef VECTOR_DYNAMIC
delete [] v1; // libera el espacio reservado para v1
delete [] v2; // libera el espacio reservado para v2
delete [] v3; // libera el espacio reservado para v3
#endif
return 0;
}

```

Listado 3. Script para la suma de vectores (SumaVectores.sh). Se supone en el script que el fichero a ejecutar se llama SumaVectorC y que se encuentra en el directorio en el que se ha ejecutado qsub.

```

#!/bin/bash
#Se asigna al trabajo el nombre SumaVectoresC_vlocales
#PBS -N SumaVectoresC_vlocales
#Se asigna al trabajo la cola ac
#PBS -q ac
#Se imprime información del trabajo usando variables de entorno de PBS
echo "Id. usuario del trabajo: $PBS_O_LOGNAME"
echo "Id. del trabajo: $PBS_JOBID"
echo "Nombre del trabajo especificado por usuario: $PBS_JOBNAME"
echo "Nodo que ejecuta qsub: $PBS_O_HOST"
echo "Directorio en el que se ha ejecutado qsub: $PBS_O_WORKDIR"
echo "Cola: $PBS_QUEUE"
echo "Nodos asignados al trabajo:"
cat $PBS_NODEFILE
#Se ejecuta SumaVectorC, que está en el directorio en el que se ha ejecutado qsub,
#para N potencia de 2 desde 2^16 a 2^26
for ((N=65536;N<67108865;N=N*2))
do
    $PBS_O_WORKDIR/SumaVectoresC $N
done

```