

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:

lscpu--->PC

```
[AgustinMeridaGutierrez usuario@agus-merida-gutierrez:~/BP0/ejer1] 2019-03-08 viernes
$ lscpu
Architecture:          x86_64
Mode(s) de operación de las CPUs: 32-bit, 64-bit
Orden de los bytes:    Little Endian
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
Mode(s) NUMA:          1
ID de fabricante:      GenuineIntel
Familia de CPU:        6
Modelo:                158
Nombre del modelo:     Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz
Revisión:              9
CPU MHz:               1499.266
CPU MHz máx.:          3800.0000
CPU MHz mín.:          800.0000
BogoMIPS:              5616.00
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 nca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpeigb rdtscp lm constant_tsc art a
rch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfperf tsc_known_freq pni pclmulqdq dtes64 monitor ds_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe popen
t tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch cpuid_fault epb invpcid_single pti ssbd ibrs ibpb stibp tpr_shadow vnml flexpriority ept vpid fsgsbase tsc_adjust bmi1 avx2 smep bn
i2 erms invpcid mpx rdseed adx snap clflushopt intel_pt xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp hwp_notify hwp_act_window hwp_epp flush_lid
[AgustinMeridaGutierrez usuario@agus-merida-gutierrez:~/BP0/ejer1] 2019-03-08 viernes
```

lscpu-->atcgrid

```
[AgustinMeridaGutierrez Aiestudiante14@atcgrid:~/BP0/ejer1] 2019-03-08 viernes
$ cat STDIN.09077
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:             66
Model:                 44
Model name:             Intel(R) Xeon(R) CPU           E5645  @ 2.40GHz
Stepping:               2
CPU MHz:               1000.000
CPU max MHz:           2400.0000
CPU min MHz:           1000.0000
BogoMIPS:              4800.17
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 nca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpeigb rdtscp lm constant_tsc arch_perfmon pebs b
ts rep_good nopl xtopology nonstop_tsc aperfperf eagerfpu pni dtes64 monitor ds_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm pcid dca sse4_1 sse4_2 popcnt lahf_lm epb ssbd ibrs ibpb stibp tpr_shadow vnml fl
expriority ept vpid dtherm ida arat spec_ctrl intel_ptibp flush_lid
[AgustinMeridaGutierrez Aiestudiante14@atcgrid:~/BP0/ejer1] 2019-03-08 viernes
```

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

RESPUESTA: En mi PC, podemos ver que tenemos 4 cores físicos (1 socket por 4 cores) y 8 lógicos. En atcgrid podemos ver que tenemos 12 cores físicos (2 socket por 6 cores) y 24 lógicos.

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:

```
[AgustinMeridaGutierrez usuario@agus-merida-gutierrez:~/BP0/ejer2] 2019-03-08 viernes
$gcc -fopenmp -o HelloOMP HelloOMP.c
[AgustinMeridaGutierrez usuario@agus-merida-gutierrez:~/BP0/ejer2] 2019-03-08 viernes
$ls
HelloOMP HelloOMP.c
[AgustinMeridaGutierrez usuario@agus-merida-gutierrez:~/BP0/ejer2] 2019-03-08 viernes
$./HelloOMP
(0:!!Hello world!!)(7:!!Hello world!!)(6:!!Hello world!!)(5:!!Hello world!!)(2:!!Hello world!!)(4:!!Hello world!!)(1:!!Hello world!!)(3:!!Hello world!!)[AgustinMeridaGutierrez usuario@agus-merida-gutierrez:~/BP0/ejer2] 2019-03-08 viernes
```

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

RESPUESTA: Se debe a que mi PC tiene 8 núcleos lógicos, por lo que el programa imprime por pantalla Hello World ese numero de veces.

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:

```
[AgustinMeridaGutierrez Aiestudiante14@atcgrid:~/BP0/ejer3] 2019-03-08 viernes
$echo 'BP0/ejer3/HelloOMP' | qsub -q ac
9737.atcgrid
[AgustinMeridaGutierrez Aiestudiante14@atcgrid:~/BP0/ejer3] 2019-03-08 viernes
$ls
HelloOMP HelloOMP.c STDIN.e9737 STDIN.o9737
[AgustinMeridaGutierrez Aiestudiante14@atcgrid:~/BP0/ejer3] 2019-03-08 viernes
$cat STDIN.o9737
(18:!!Hello world!!)(8:!!Hello world!!)(17:!!Hello world!!)(16:!!Hello world!!)(22:!!Hello world!!)(9:!!Hello world!!)(8:!!Hello world!!)(15:!!Hello world!!)(12:!!Hello world!!)(19:!!Hello world!!)(5:!!Hello world!!)(10:!!Hello world!!)(3:!!Hello world!!)(4:!!Hello world!!)(14:!!Hello world!!)(7:!!Hello world!!)(21:!!Hello world!!)(1:!!Hello world!!)(11:!!Hello world!!)(23:!!Hello world!!)(20:!!Hello world!!)(2:!!Hello world!!)(6:!!Hello world!!)(13:!!Hello world!!)[AgustinMeridaGutierrez Aiestudiante14@atcgrid:~/BP0/ejer3] 2019-03-08 viernes
$
```

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

RESPUESTA: Como en el servidor atcgrid tenemos 24 cores lógicos, el programa nos imprime por pantalla ese numero de Hello World.

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:

Nuevo Código:

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

int main(void){
#pragma omp parallel
    printf("(%d!!!Hello ",
            omp_get_thread_num());
    printf("(%d: World!!!",
            omp_get_thread_num());
return(0);
}
```

```
sftp> put BP0/ejer4/HelloOMP2
Uploading BP0/ejer4/HelloOMP2 to /home/A1estudiante14/BP0/ejer4/HelloOMP2
BP0/ejer4/HelloOMP2 100% 8688 168.4KB/s 00:00
sftp> █
```

```
[AgustinMeridaGutierrez A1estudiante14@atcgrid:~/BP0/ejer4] 2019-03-13 miércoles
$ls
HelloOMP2 script_helloomp.sh
[AgustinMeridaGutierrez A1estudiante14@atcgrid:~/BP0/ejer4] 2019-03-13 miércoles
$qsub script_helloomp.sh -q ac
11456.atcgrid
[AgustinMeridaGutierrez A1estudiante14@atcgrid:~/BP0/ejer4] 2019-03-13 miércoles
$cat helloomp.o11456
Id. usuario del trabajo: A1estudiante14
Id. del trabajo: 11456.atcgrid
Nombre del trabajo especificado por usuario: helloomp
Directorio en el que se ha ejecutado qsub: /home/A1estudiante14/BP0/ejer4
Directorio de trabajo: /home/A1estudiante14
Cola: ac
Nodo que ejecuta qsub: atcgrid
Nodos asignados al trabajo:
atcgrid3
Nº de threads inicial: 12
Directorio de trabajo: /home/A1estudiante14/BP0/ejer4

1.Para 12 threads:
(6!!!Hello (2!!!Hello (0!!!Hello (11!!!Hello (5!!!Hello (9!!!Hello (4!!!Hello (8!!!Hello (7!!!Hello (10!!!Hello (1!!!Hello (3!!!Hello (0: World!!!
1.Para 6 threads:
(0!!!Hello (1!!!Hello (5!!!Hello (3!!!Hello (4!!!Hello (2!!!Hello (0: World!!!
1.Para 3 threads:
(1!!!Hello (2!!!Hello (0!!!Hello (0: World!!!
1.Para 1 threads:
(0!!!Hello (0: World!!![AgustinMeridaGutierrez A1estudiante14@atcgrid:~/BP0/ejer4] 2019-03-13 miércoles
$█
```

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

RESPUESTA: El script ha sido ejecutado por el nodo atcgrid . Esta información la sabemos por que en el script hemos puesto en el apartado *Nodo que ejecuta qsub*: “PBS_O_WORKDIR” que nos indica el nodo en el que se ejecuta el programa HelloOMP2.

(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

```
[AgustinMeridaGutierrez A1estudiante14@atcgrid:~/BP0/ejer4] 2019-03-13 miércoles
$cat helloomp.o11459
Id. usuario del trabajo: A1estudiante14
Id. del trabajo: 11459.atcgrid
Nombre del trabajo especificado por usuario: helloomp
Directorio en el que se ha ejecutado qsub: /home/A1estudiante14/BP0/ejer4
Directorio de trabajo: /home/A1estudiante14
Cola: ac
Nodo que ejecuta qsub: atcgrid
Nodos asignados al trabajo:
atcgrid3
Nº de threads inicial: 12
Directorio de trabajo: /home/A1estudiante14/BP0/ejer4

1.Para 12 threads:

1.Para 6 threads:

1.Para 3 threads:

1.Para 1 threads:
[AgustinMeridaGutierrez A1estudiante14@atcgrid:~/BP0/ejer4] 2019-03-13 miércoles
$
```

En este caso, al cambiar la orden, no se ejecuta el programa por que no se le indica en que nodo ha de realizarlo, por lo que no nos muestra el resultado correspondiente a la ejecución de HelloOMP2 en el nodo atcgrid

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:

```
[AgustinMeridaGutierrez usuario@agus-merida-gutierrez:~/BP0/ejer5] 2019-03-14 jueves
$gcc -O2 SumaVectores.c -o SumaVectores -lrt
[AgustinMeridaGutierrez usuario@agus-merida-gutierrez:~/BP0/ejer5] 2019-03-14 jueves
$./SumaVectores 5
Tiempo(seg.):0.000000247 / Tamaño Vectores:5 / V1[0]+V2[0]=V3[0](0.500000+0.500000=1.000000) / / V1[4]+V2[4]=V3[4](0.900000+0.100000=1.000000) /
[AgustinMeridaGutierrez usuario@agus-merida-gutierrez:~/BP0/ejer5] 2019-03-14 jueves
$
```

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: Es un dato de tipo double que almacena el tiempo que tardan los vectores en sumarse.

(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: La estructura de datos de esta función consiste en un struct. Devuelve un dato correspondiente a los segundos, y otro correspondiente a los nanosegundos. Además, todas estas estructuras de datos se encuentran almacenadas en la librería “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: Por lo general, devuelve el tiempo actual en la estructura time descrita en el apartado anterior, un tipo time_h para los segundos y un long para los nanosegundos.

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:

```
[AgustinMeridaGutierrez usuario@agus-merida-gutierrez:~/BP0/ejer7] 2019-03-14 jueves
$ ./SumaVectores.sh
Tiempo(seg.):0.000535202 / 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.):0.000953386 / 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) /
Tiempo(seg.):0.001018888 / 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) /
./SumaVectores.sh: línea 21: 4819 Violación de segmento (core generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 21: 4821 Violación de segmento (core generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 21: 4823 Violación de segmento (core generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 21: 4825 Violación de segmento (core generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 21: 4827 Violación de segmento (core generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 21: 4829 Violación de segmento (core generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 21: 4831 Violación de segmento (core generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 21: 4833 Violación de segmento (core generado) ./SumaVectoresC $N
[AgustinMeridaGutierrez usuario@agus-merida-gutierrez:~/BP0/ejer7] 2019-03-14 jueves

sftp> put BP0/ejer7/SumaVectores.sh
Uploading BP0/ejer7/SumaVectores.sh to /home/Aiestudiante14/BP0/ejer7/SumaVectores.sh
BP0/ejer7/SumaVectores.sh 100% 798 16.3KB/s 00:00
sftp> put BP0/ejer7/SumaVectores.c
Uploading BP0/ejer7/SumaVectores.c to /home/Aiestudiante14/BP0/ejer7/SumaVectores.c
BP0/ejer7/SumaVectores.c 100% 3333 66.2KB/s 00:00
sftp>

[AgustinMeridaGutierrez Aiestudiante14@atcgrid:~/BP0/ejer7] 2019-03-14 jueves
$ qsub SumaVectores.sh -q ac
11490.atcgrid
[AgustinMeridaGutierrez Aiestudiante14@atcgrid:~/BP0/ejer7] 2019-03-14 jueves
$ cat SumaVectoresC_vlocales.o11490
Id. usuario del trabajo: Aiestudiante14
Id. del trabajo: 11490.atcgrid
Nombre del trabajo especificado por usuario: SumaVectoresC_vlocales
Modo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/Aiestudiante14/BP0/ejer7
Cola: ac
Nodos asignados al trabajo:
atcgrid1
Tiempo(seg.):0.000461571 / 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.):0.000945077 / 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) /
Tiempo(seg.):0.001713358 / 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) /
Tiempo(seg.):0.002858935 / 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) /
[AgustinMeridaGutierrez Aiestudiante14@atcgrid:~/BP0/ejer7] 2019-03-14 jueves
$ cat SumaVectoresC_vlocales.e11490
/var/spool/pbs/mom_priv/jobs/11490.atcgrid.SC: línea 21: 23439 Violación de segmento (core generado) $PBS_O_WORKDIR/SumaVectoresC $N
/var/spool/pbs/mom_priv/jobs/11490.atcgrid.SC: línea 21: 23441 Violación de segmento (core generado) $PBS_O_WORKDIR/SumaVectoresC $N
/var/spool/pbs/mom_priv/jobs/11490.atcgrid.SC: línea 21: 23444 Violación de segmento (core generado) $PBS_O_WORKDIR/SumaVectoresC $N
/var/spool/pbs/mom_priv/jobs/11490.atcgrid.SC: línea 21: 23447 Violación de segmento (core generado) $PBS_O_WORKDIR/SumaVectoresC $N
/var/spool/pbs/mom_priv/jobs/11490.atcgrid.SC: línea 21: 23450 Violación de segmento (core generado) $PBS_O_WORKDIR/SumaVectoresC $N
/var/spool/pbs/mom_priv/jobs/11490.atcgrid.SC: línea 21: 23452 Violación de segmento (core generado) $PBS_O_WORKDIR/SumaVectoresC $N
/var/spool/pbs/mom_priv/jobs/11490.atcgrid.SC: línea 21: 23454 Violación de segmento (core generado) $PBS_O_WORKDIR/SumaVectoresC $N
[AgustinMeridaGutierrez Aiestudiante14@atcgrid:~/BP0/ejer7] 2019-03-14 jueves
$
```

El error se debe a que el tamaño máximo ha sido superado por el vector a partir de la quinta suma de vectores. Por eso, muestra solamente 4 en el archivo .o.

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: Los archivos que hacen uso de los vectores globales se han subido sin cambiarles el nombre, y posteriormente se les ha cambiado este en atcgrid.

```
sftp> put BP0/ejer8/SumaVectores
Uploading BP0/ejer8/SumaVectores to /home/A1estudiante14/BP0/ejer8/SumaVectores
BP0/ejer8/SumaVectores          100% 8624    53.2KB/s   00:00
sftp> put BP0/ejer8/SumaVectores.sh
Uploading BP0/ejer8/SumaVectores.sh to /home/A1estudiante14/BP0/ejer8/SumaVectores.sh
BP0/ejer8/SumaVectores.sh      100% 785    17.5KB/s   00:00
sftp> put BP0/ejer8/SumaVectores.c
Uploading BP0/ejer8/SumaVectores.c to /home/A1estudiante14/BP0/ejer8/SumaVectores.c
BP0/ejer8/SumaVectores.c      100% 3337    71.6KB/s   00:00
sftp> put BP0/e
 ejer1/   ejer10/  ejer2/   ejer3/   ejer4/   ejer5/   ejer6/   ejer7/
 ejer8/   ejer9/
sftp> put BP0/ejer8/Su
SumaVectores      SumaVectores.c      SumaVectores.sh      SumaVectoresD
SumaVectoresD.c   SumaVectoresD.sh
sftp> put BP0/ejer8/SumaVectoresD
Uploading BP0/ejer8/SumaVectoresD to /home/A1estudiante14/BP0/ejer8/SumaVectoresD
BP0/ejer8/SumaVectoresD      100% 12KB   69.3KB/s   00:00
sftp> put BP0/ejer8/SumaVectoresD.sh
Uploading BP0/ejer8/SumaVectoresD.sh to /home/A1estudiante14/BP0/ejer8/SumaVectoresD.sh
BP0/ejer8/SumaVectoresD.sh  100% 785    17.1KB/s   00:00
sftp> put BP0/ejer8/SumaVectoresD.c
Uploading BP0/ejer8/SumaVectoresD.c to /home/A1estudiante14/BP0/ejer8/SumaVectoresD.c
BP0/ejer8/SumaVectoresD.c   100% 3333    70.9KB/s   00:00
sftp>
```

Vectores Globales en PC y en atcgrid

```
[AgustinMeridaGutierrez usuario@agus-merida-gutierrez:~/BP0/ejer8] 2019-03-14 jueves
S./SumaVectoresD.sh
Tiempo(seg.):0.000368396 / 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.):0.000738822 / 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) /
Tiempo(seg.):0.001558672 / 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) /
Tiempo(seg.):0.001757509 / 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) /
Tiempo(seg.):0.003445509 / 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) /
/
Tiempo(seg.):0.006183411 / 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) /
/
Tiempo(seg.):0.012899765 / 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) /
/
Tiempo(seg.):0.024425676 / 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) /
/
Tiempo(seg.):0.047835566 / 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) /
Tiempo(seg.):0.096949497 / 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) /
Tiempo(seg.):0.190217510 / 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) /
[AgustinMeridaGutierrez usuario@agus-merida-gutierrez:~/BP0/ejer8] 2019-03-14 jueves
```

```
[AgustinMeridaGutierrez Aiestudiante14@atcgrid:~/BP0/ejer8] 2019-03-14 jueves
$ qsub sumaVectores.sh -q ac
11593.atcgrid
[AgustinMeridaGutierrez Aiestudiante14@atcgrid:~/BP0/ejer8] 2019-03-14 jueves
$ cat sumaVectoresC_vglobales.o11593
Id. usuario del trabajo: Aiestudiante14
Id. del trabajo: 11593.atcgrid
Nombre del trabajo especificado por usuario: SumaVectoresC_vglobales
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/Aiestudiante14/BP0/ejer8
Cola: ac
Nodos asignados al trabajo:
atcgrid2
Tiempo(seg.):0.000553345 / 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.):0.000693251 / 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) /
Tiempo(seg.):0.001463122 / 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) /
Tiempo(seg.):0.002560182 / 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) /
Tiempo(seg.):0.005224840 / 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) /
Tiempo(seg.):0.009540665 / 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) /
Tiempo(seg.):0.017984291 / 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) /
Tiempo(seg.):0.034431868 / 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) /
Tiempo(seg.):0.067934343 / 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) /
Tiempo(seg.):0.134794651 / 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) /
Tiempo(seg.):0.264092382 / 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) /
[AgustinMeridaGutierrez Aiestudiante14@atcgrid:~/BP0/ejer8] 2019-03-14 jueves
```

Vectores dinámicos en PC y en atcgrid

```
[AgustinMeridaGutierrez usuario@agus-merida-gutierrez:~/BP0/ejer8] 2019-03-14 jueves
$ ./SumaVectoresD.sh
Tiempo(seg.):0.000368555 / 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.):0.000777721 / 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) /
Tiempo(seg.):0.001629668 / 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) /
Tiempo(seg.):0.003927366 / 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) /
Tiempo(seg.):0.006324489 / 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) /
Tiempo(seg.):0.012749721 / 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) /
Tiempo(seg.):0.025895203 / 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) /
Tiempo(seg.):0.051430356 / 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) /
Tiempo(seg.):0.096562768 / 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) /
Tiempo(seg.):0.194622673 / 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) /
Tiempo(seg.):0.264092382 / 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) /
[AgustinMeridaGutierrez usuario@agus-merida-gutierrez:~/BP0/ejer8] 2019-03-14 jueves
```

```
[AgustinMeridaGutierrez Aiestudiante14@atcgrid:~/BP0/ejer8] 2019-03-14 jueves
$ qsub sumaVectores.sh -q ac
11593.atcgrid
[AgustinMeridaGutierrez Aiestudiante14@atcgrid:~/BP0/ejer8] 2019-03-14 jueves
$ cat sumaVectoresC_vglobales.o11593
Id. usuario del trabajo: Aiestudiante14
Id. del trabajo: 11593.atcgrid
Nombre del trabajo especificado por usuario: SumaVectoresC_vglobales
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/Aiestudiante14/BP0/ejer8
Cola: ac
Nodos asignados al trabajo:
atcgrid2
Tiempo(seg.):0.000553345 / 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.):0.000693251 / 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) /
Tiempo(seg.):0.001463122 / 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) /
Tiempo(seg.):0.002560182 / 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) /
Tiempo(seg.):0.005224840 / 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) /
Tiempo(seg.):0.009540665 / 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) /
Tiempo(seg.):0.017984291 / 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) /
Tiempo(seg.):0.034431868 / 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) /
Tiempo(seg.):0.067934343 / 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) /
Tiempo(seg.):0.134794651 / 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) /
Tiempo(seg.):0.264092382 / 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) /
[AgustinMeridaGutierrez Aiestudiante14@atcgrid:~/BP0/ejer8] 2019-03-14 jueves
```

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

Tabla para PC

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0,000535202	0,000368396	0,000368555
131072	1048576	0,00095386	0,000738022	0,000737721
262144	2097152	0,001018888	0,001558672	0,001828805
524288	4194304		0,001757509	0,001629668
1048576	8388608		0,003445509	0,003927366
2097152	16777216		0,006183411	0,006324489
4194304	33554432		0,006183411	0,012749721
8388608	67108864		0,024425676	0,025895203
16777216	134217728		0,017835566	0,051430356
33554432	268435456		0,096949497	0,096562768
67108864	536870912		0,190217510	0,194622673

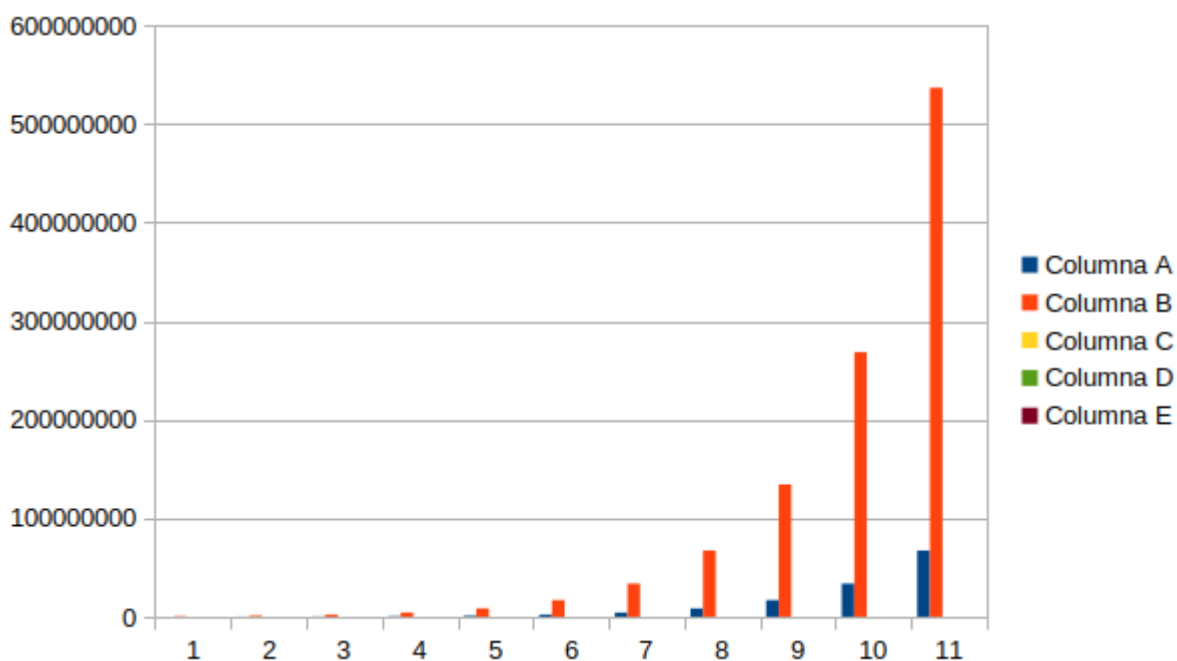
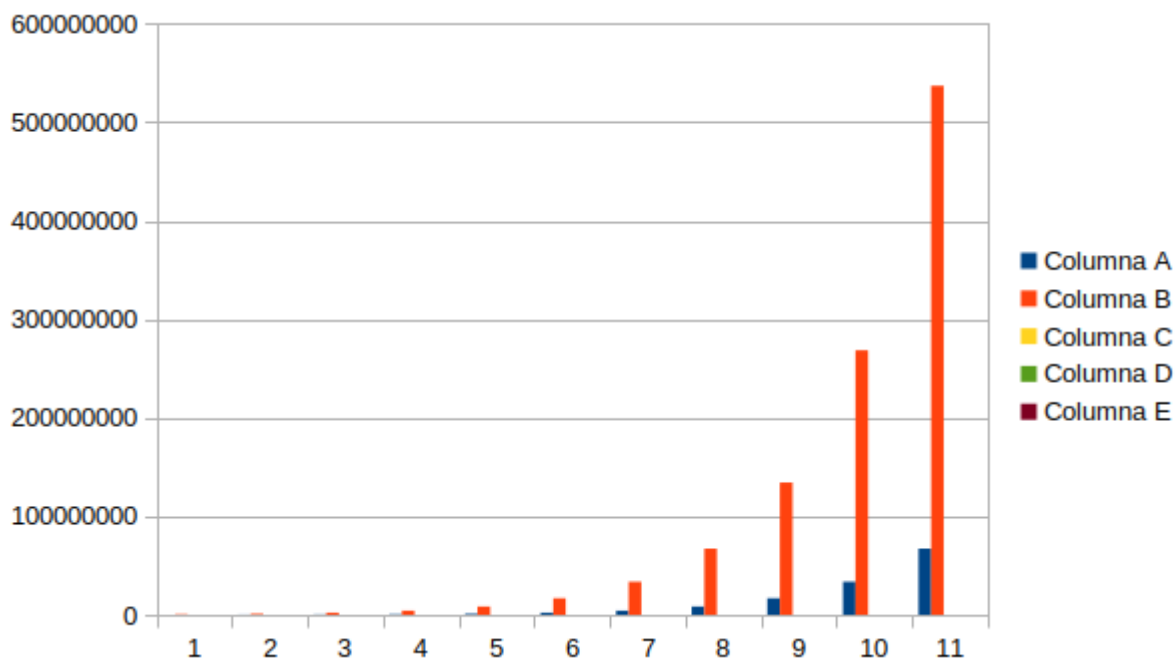


Tabla para atcgrid

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0,000461571	0,000553345	0,000467683
131072	1048576	0,000945077	0,000693251	0,000935368
262144	2097152	0,001713358	0,001463122	0,001875594
524288	4194304	0,002858935	0,002560182	0,002843909
1048576	8388608		0,005224840	0,005561661
2097152	16777216		0,00950665	0,009772751
4194304	33554432		0,017984291	0,017763581
8388608	67108864		0,034431868	0,032932669
16777216	134217728		0,067934343	0,064808640
33554432	268435456		0,134794651	0,128903374
67108864	536870912		0,264092382	0,254592937



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

En el PC las operaciones se realizan con mayor facilidad que en el nodo ategid.

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

RESPUESTA: El máximo valor que puede almacenar la variable N siendo de tipo unsigned int es de $2^{32}-1$. Esto se debe a que es el tamaño de este dato. Si se superase, provocaría desbordamiento.

(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: Al realizar el cambio, no se crea el ejecutable. Esto se debe a que se produce truncamiento,

```
[AgustinMeridaGutierrez usuario@agus-merida-gutierrez:~/BP0/ejer10] 2019-03-14 jueves
$gcc -O2 SumaVectoresG.c -o SumaVectores -lrt
/tmp/ccIBnaPi.o: En la función 'main':
SumaVectoresG.c:(.text.startup+0x5e): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo 'v2' definido en la sección COMMON en /tmp/ccIBnaPi.o
SumaVectoresG.c:(.text.startup+0xb1): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo 'v3' definido en la sección COMMON en /tmp/ccIBnaPi.o
SumaVectoresG.c:(.text.startup+0x130): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo 'v3' definido en la sección COMMON en /tmp/ccIBnaPi.o
SumaVectoresG.c:(.text.startup+0x13b): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo 'v2' definido en la sección COMMON en /tmp/ccIBnaPi.o
SumaVectoresG.c:(.text.startup+0x1a0): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo 'v2' definido en la sección COMMON en /tmp/ccIBnaPi.o
SumaVectoresG.c:(.text.startup+0x1a7): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo 'v3' definido en la sección COMMON en /tmp/ccIBnaPi.o
collect2: error: ld returned 1 exit status
[AgustinMeridaGutierrez usuario@agus-merida-gutierrez:~/BP0/ejer10] 2019-03-14 jueves
$ls
SumaVectoresG.c
[AgustinMeridaGutierrez usuario@agus-merida-gutierrez:~/BP0/ejer10] 2019-03-14 jueves
$
```

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
```